Università degli Studi di Napoli "Federico II"

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica

Tesi di Laurea in Sistemi Informativi

"Progetto e Realizzazione di una Data Warehouse per Dati Chimico-Fisici sugli Alimenti"

Relatore:
Ch.mo Prof. Antonio d'Acierno

Candidato:
Fabrizio di Cesare
(matr. 041/2512)

ANNO ACCADEMICO 2002/2003

A Feliciana,
il mio tenero amore,
che riempie di vita
le mie giornate.

INDICE

CAPITOLO UNO

INTRODUZIONE

Questo capitolo introduttivo illustra lo scopo della tesi e descrive a grandi linee il contenuto dei capitoli successivi. Viene infine descritto dettagliatamente il primo passo svolto: la ricerca e selezione delle fonti.
Il materiale presentato è dunque di fondamentale importanza per la comprensione di tutto il seguito del testo.

1.1 Il Problema Affrontato

Oggetto di questa tesi è la descrizione del processo di progettazione e sviluppo di un magazzino di dati riguardanti la composizione chimico-fisica degli alimenti e del relativo software di consultazione.

La tesi tratta principalmente gli aspetti informatici del progetto, mentre vengono fornite solo le informazioni nutrizionali indispensabili alla comprensione dello stesso. Inoltre il linguaggio utilizzato in buona parte del testo è spesso tecnico: si veda [12] per una maggiore comprensione dei termini usati.

Si ricorda, comunque, come sia sempre più sentito l'interesse per i dati di composizione alimentare: l'utente comune, anche grazie alla maggiore sensibilizzazione alla scienza fornita dai media, è spesso interessato ad avere informazioni su particolari aspetti compositivi del cibo che mangia (gli esempi più banali sono le informazioni sul contenuto calorico o su quello vitaminico).

Anche gli esperti del settore, d'altro canto, cercano spesso un metodo rapido per accedere alla grande mole di informazioni oggetto delle loro ricerche, che possono ad esempio essere mirate alla costruzione di diete particolari o alla scoperta di legami tra malattie e nutrizione (non a caso, come sarà ampiamente documentato nel seguito, una delle fonti da cui provengono i dati presenti nel magazzino è stata gentilmente fornita dall'Istituto Europeo di Oncologia, [2]).

Per rendere quindi maggiormente fruibili i dati, si è deciso di realizzare in prima battuta un software di consultazione della datawarehouse accessibile via web.

1.2 Definizioni Fondamentali

Prima di descrivere a fondo le motivazioni e la natura del progetto, si ritiene necessario dare alcune definizioni basilari riguardanti l'ambito applicativo di riferimento. Per maggiori informazioni si faccia riferimento alla letteratura specifica e alla bibliografia presentata.

Si tenga comunque presente che le definizioni presentate non vogliono essere universali (si veda ad esempio la definizione di "valore"), ma indicano semplicemente il

senso che avranno alcuni termini nel seguito (a meno che non venga esplicitamente indicato diversamente).

Fonte (o sorgente) di dati: è inteso come un insieme di dati di composizione provenienti da una singola persona, un singolo gruppo di autori o una singola organizzazione.

Una fonte è considerata **primaria** se i suoi dati vengono direttamente inseriti nel magazzino a nome dell'ente che li ha rilasciati. Si parla invece di **fonti secondarie** per descrivere quelle fonti da cui provengono i dati delle primarie.

Alimento (cibo): nella datawarehouse vengono considerati distinti due alimenti con la stessa descrizione, ma provenienti da fonti primarie diverse. Questo perché si vogliono in qualche modo confrontare i dati e inoltre occorrenze diverse dello stesso cibo possono avere composizioni differenti.

Si fa anche notare che a caratterizzare principalmente l'alimento è la sua composizione e non la sua descrizione in linguaggio naturale: non si può presupporre che due compilatori di due fonti diverse volessero intendere lo stesso alimento semplicemente perché ne hanno dato lo stesso nome.

Componente: un componente è una qualsiasi proprietà del cibo che sia soggetta a misurazione scientifica. In particolare un componente comprende sia i **nutrienti** (come proteine, vitamine, minerali, etc.) che altre caratteristiche, come pH, parte edibile, etc..

Nel seguito comunque la parola nutriente verrà utilizzata anche come sinonimo di componente.

Come per gli alimenti, si assumono distinti due componenti provenienti da fonti primarie diverse, anche qualora questi abbiano descrizioni simili. Questo anche perché i metodi per ricavare la proprietà possono differire da fonte a fonte.

Valore (o composizione): quantità di un componente in un alimento e le sue proprietà statistiche.

Metodo: sistema tramite il quale viene determinato il valore di un componente in un alimento: un metodo può essere chimico, fisico, numerico, etc.

Unità di misura: unità con cui è espresso il valore di un componente (es. grammi, calorie, etc.). Per un valore che non presenti unità di misura si parla di "numero puro" o di "rapporto".

Modo di espressione della misura: quantità di alimento cui si riferisce il valore misurato (es. "per 100g di parte edibile", "per 100g di sostanza secca", etc.).

Viene sempre utilizzato assieme all'unità di misura (es. "g per 100 g di parte edibile").

1.3 Stato dell'Arte

Ciò che ha dato il via al progetto è stata la constatazione che, da quanto emerge dalle ricerche svolte, sono ben poche le risorse direttamente accessibili via web riguardanti dati di composizione di alimenti e che siano anche in italiano.

Dal sito dell' "Istituto Nazionale di Ricerca per gli Alimenti e la Nutrizione" (INRAN, vedi [4]) è comunque scaricabile un applicativo per macchine Windows che permette di consultare (in italiano) le tabelle di composizione fornite dall'ente. Tale software permette anche di effettuare ricerche per alimenti contenenti particolari valori dei componenti. Si comprende comunque come questo programma non sia direttamente fruibile da parte di tutta l'utenza web.

In ambito internazionale, i migliori siti trovati sono quello del "Nutrient Data Laboratory" (NDL, vedi [3]) statunitense e quello del "Danish Food Composition Databank". Entrambi sono in inglese (il secondo anche in danese) e ovviamente non contengono informazioni riguardanti cibi consumati esclusivamente in Italia.

Il sito statunitense permette solo una semplice ricerca per stringa tra i nomi degli alimenti, riservando comunque a dei report cartacei (disponibili sul sito in formato pdf) la presentazione di altre informazioni.

Il sito danese invece consente anche altri tipi di ricerca (oltre a quella per nome): gli alimenti possono essere visionati suddivisi per categoria oppure si può vedere una lista degli alimenti contenenti un particolare componente (in ordine crescente di valore).

In ogni caso i report prodotti dinamicamente dalle applicazioni web citate non contengono molte informazioni che possono essere utili agli esperti del settore, quali ad esempio: metodi utilizzati, dati statistici approfonditi, fonti, etc.

1.4 Linee Guida del Progetto

Il progetto tende ovviamente a migliorare sotto molti aspetti lo stato dell'arte descritto. In particolare si tende a costruire una datawarehouse che:

1) Integri le informazioni provenienti dalle varie fonti, inserendo tutti i dati presenti nelle stesse senza cambiarne il significato.

2) Mantenga i collegamenti dei dati con le fonti in maniera trasparente: deve sempre essere reperibile la provenienza di un'informazione.

3) Sia gestibile semplicemente, anche attraverso dei moduli software creati ad hoc.

4) Sia navigabile in maniera "elastica", con buoni collegamenti tra i frammenti di informazione.

5) Contenga un supporto multilingua: i dati devono essere consultabili sia nella lingua originaria che in italiano.

6) Sia accessibile via web tramite un'interfaccia amichevole.

Come si può vedere, queste semplici linee guida sono ben lungi dall'essere un rigoroso documento di specifiche: questo perché scopo della tesi è descrivere il progetto e non documentarlo approfonditamente, come invece si cerca di fare con il materiale allegato alla tesi stessa.

1.5 Passi Fondamentali

Nello svolgere il progetto sono stati seguiti gli step riportati di seguito (ovviamente la presentazione in serie dei passi non è indicativa di un processo di sviluppo sequenziale):

1) **Selezione delle fonti:** questa fase ha contemplato la ricerca e la scelta di sorgenti di dati da inserire nel magazzino finale.

2) **Reverse engineering delle fonti:** gli schemi iniziali delle fonti sono stati studiati e ampiamente rimaneggiati per meglio individuare le entità coinvolte e per facilitare la successiva fase di integrazione.

3) **Integrazione:** questo step comprende il progetto di un database che segua tutte le linee guida mostrate in precedenza. Da tale database è stata poi ricavata la datawarehouse finale.

4) **Sviluppo dei moduli di feeding:** sono stati progettati e implementati dei moduli automatici per l'inserimento, l'eliminazione e l'aggiornamento dei dati presenti nel database.

5) **Creazione di un prototipo di sito:** è stato sviluppato un sito per la consultazione dei dati presenti nel magazzino di dati.

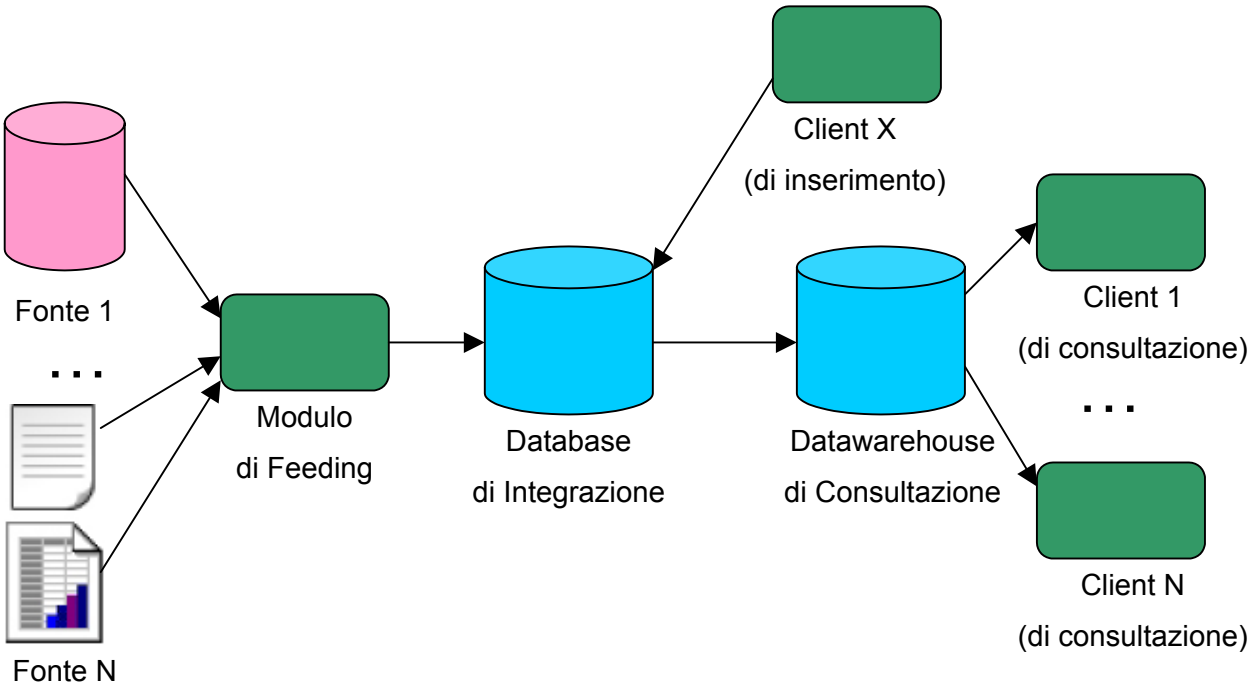
1.6 Architettura di Massima del Sistema

La figura seguente mostra la struttura di base del sistema: le frecce indicano il flusso dei dati tra i vari componenti dello stesso.

Lo schema mostra chiaramente come il modulo software automatico di feeding prelevi i dati dalle singole fonti e inserisca questi ultimi in quello che è stato chiamato "Database di Integrazione". Tale base di dati contiene già tutti i dati finali, ma in una suddivisione che facilita maggiormente l'inserimento (e quindi l'integrazione) che la consultazione: non sono presenti ridondanze e i dati non sono fortemente aggregati.

Da questo db, soprattutto tramite l'utilizzo di viste materializzate e di indici che facilitino le ricerche, è stata creata una "Datawarehouse di Consultazione".

Su questa architettura si poggiano poi i vari moduli client, che possono sia consultare le informazioni presenti nel magazzino, sia inserire dati nella database. Questo inserimento non automatico può per esempio rendersi necessario nel caso di lavori cartacei. La suddivisione tra "client per l'inserimento" e "client per la consultazione" è stata eseguita ovviamente solo a scopi illustrativi: i due moduli non devono per forza essere separati. Inoltre i client non si appoggiano direttamente sui dati, ma sono ovviamente presenti dei server, che per semplicità non sono schematizzati in figura.



1.7 Tecnologie Utilizzate

Il sistema operativo scelto per ospitare i server è **Linux** (in particolare la distribuzione Red Hat). Questa scelta è stata dettata soprattutto da motivi di sicurezza e di affidabilità: le alternative (i sistemi Microsoft) sono purtroppo rinomate per essere più vulnerabili. Inoltre, trattandosi di un progetto svolto nell'ambito di una tesi, si è sempre cercato di utilizzare software freeware (o al più shareware): sotto Linux è molto più facile reperire questo tipo di applicativi.

I server da realizzare sono ovviamente un DBMS server ed un web server (ovviamente possono anche risiedere entrambi sulla stessa macchina). La tecnologia utilizzata per il primo è **PostgreSQL**: si tratta di uno tra i migliori DBMS freeware disponibili su piattaforma Linux/Unix, con caratteristiche simili a ben più blasonati prodotti commerciali. Con il sistema operativo utilizzato si sarebbe potuto adoperare anche MySQL, ma si tratta di un prodotto con caratteristiche diverse e con un supporto peggiore per quanto riguarda trigger, query innestate, viste, proprietà acide, schemi e domini etc… Tutte queste feature sono state invece utilizzate nel sistema finale.

Per quanto riguarda il web server, la scelta invece non poteva che ricadere su **Apache**, semplicemente il più sicuro e testato in questo genere di applicativi. Per creare poi pagine dinamiche è stato utilizzato **PHP**, scelto anche perché si tratta di un linguaggio di scripting lato server, cosa che da sola rende molto più facile la compatibilità con qualsiasi browser e assai meno facile la vita agli hacker.

Durante tutto il processo di reverse engineering delle fonti, di integrazione dei dati e di progettazione del software è stata utilizzato il CASE **PowerDesigner**, uno strumento flessibile e con supporto per la stragrande maggioranza dei linguaggi di programmazione e dei DBMS presenti sul mercato.

Per la programmazione dei moduli di feeding la scelta è ricaduta sulla tecnologia di accesso ai dati **ODBC** e sul linguaggio di programmazione **C++** (di cui sono state utilizzate solo le librerie ANSI, oltre ovviamente alle API ODBC).

La tecnologia Open Data Base Connectivity è nata sì in ambiente Microsoft, ma è ormai diventata uno standard de facto ed è abbastanza matura anche in ambiente Linux: il suo utilizzo ha permesso di scrivere moduli di feeding indipendenti dal DBMS e un codice ampiamente portabile, qualora se ne presentasse la necessità, anche in ambiente Windows.

1.8 Ricerca delle Fonti

Il primo passo affrontato nel progetto è stato quello di cercare via web le possibili fonti da cui trarre i dati per il magazzino: i siti vengono presentati suddivisi tra gratuiti e a pagamento. Di questi ultimi viene fornita solo una descrizione delle eventuali parti dimostrative gratuite.

La ricerca si è mossa quasi esclusivamente su siti “ufficiali”, cioè siti di organizzazioni nazionali o internazionali. Questo allo scopo di ottenere dati iniziali che comunque presentino una buona qualità.

Nelle descrizioni che seguono si è cercato di condensare in poche parole sia le informazioni sui database reperibili dai siti, sia quelle sull’eventuale software di consultazione. Informazioni più dettagliate vengono fornite in seguito riguardo alle sole fonti selezionate per l’inserimento nella datawarehouse finale.

1.8.1 Siti Gratuiti

http://www.inran.it/

Sito dell’Istituto Nazionale di Ricerca per gli Alimenti e la Nutrizione.

Dal sito sono scaricabili un database (in formato Access) contenente informazioni su 790 alimenti consumati in Italia e un programma di consultazione dello stesso (entrambi aggiornati al 2000). Sono altresì scaricabili degli articoli di approfondimento (in formato pdf) su argomenti nutrizionali.

Il programma di consultazione, installabile su sistemi Windows, permette di sfogliare (come in un libro) le tabelle dei vari alimenti. E’ anche possibile cercare gli alimenti che contengono maggiormente un certo nutriente (o un certo intervallo di valori dello stesso).

http://www.ieo.it

Sito dell’Istituto Europeo di Oncologia.

Il sito contiene una “Banca dati di composizione degli alimenti per studi epidemiologici in Italia”. Il database scaricabile è un libro in formato pdf (i file ASCII si possono richiedere) pubblicato nel 1998 e contenente informazioni su 778 alimenti e 37 componenti alimentari. Gli alimenti inclusi sono per lo più alimenti semplici, crudi e quelli più frequentemente consumati dalla popolazione italiana.

Non sembra essere presente un software di consultazione.

http://www.nal.usda.gov/fnic/foodcomp

Sito del Nutrient Data Laboratory (NDL), facente capo all’United States Department of Agriculture (USDA).

Dal sito è scaricabile il “National Nutrient Database for Standard Reference, Release 16”: un database (disponibile in formato Access o ASCII) aggiornato annualmente e contenente informazioni per 125 componenti alimentari e 6661 alimenti consumati negli Stati Uniti.
Dal sito si possono scaricare anche dei reports fatti sul db e un applicativo Windows per fare semplici ricerche. Si può fare anche una ricerca per stringa on-line.
Altro materiale scaricabile riguarda studi più specifici su altri componenti alimentari e sui fattori di ritenzione degli stessi.

http://www.isa.cnr.it/PRODTIP/

Parte del sito dell’ISA (Istituto di Scienze dell’Alimentazione del CNR) dedicata ai prodotti caseari campani. I dati presenti sono in parte descrizioni e in parte analisi chimico-fisiche. Queste ultime sono presenti sul sito come immagini dovute alla scansione di un lavoro cartaceo (quattro pagine per alimento).

http://www.fao.org/infoods/data_en.stm

Sito dell’ “International Network of Food Data Systems (INFOODS)”, organizzazione della FAO il cui scopo è stimolare e coordinare i progetti regionali di studio sul cibo, al fine di ottenere dati di composizione alimentari adeguati e affidabili.
Il sito non contiene dati di composizione alimentare, ma è comunque presente moltissimo materiale, tra cui è sicuramente interessante la “Directory” dei db regionali, dalla quale si può accedere ai siti (e/o alle pubblicazioni) di composizione degli alimenti di tutto il mondo.
E’ presente anche una lista di pubblicazioni INFOODS, disponibili anche on-line, che tra l’altro danno indicazioni su come compilare db di composizione alimentare e su come descrivere i dati in modo che possano essere scambiati nella rete INFOODS (tramite l’utilizzo di tags).

http://www.foodcomp.dk/fcdb_aboutfooddata.htm

Sito del Danish Food Composition Databank danese, che contiene delle tabelle di composizione dettagliate (in inglese e danese) consultabili on-line. Il

database non sembra essere scaricabile, anche perché il sito comunque è "under construction" da moltissimo tempo.
Buono il software di consultazione via web, di cui si è già discusso in precedenza.

http://www.inta.cl/latinfoods/default.htm
Sito di Latinfoods, la "Rete Latinoamericana di composizione degli alimenti", costruita con gli auspici della FAO (in particolare del progetto INFOODS): il sito (completamente in spagnolo) contiene (per quanto si è compreso) un database di alimenti latinoamericani (consultabile (sembra) solo via Internet all'URL: http://www.rlc.fao.org/bases/alimento/busca.asp) e dei link ai siti dei db delle singole nazioni, al momento solo Argentina e Brasile (che sembrano essere inclusi nel database di Latinfoods).

http://www.unlu.edu.ar/~argenfoods/Tablas/Tabla.htm
Sito (in spagnolo) del db argentino di composizione: si possono scaricare le tabelle Excel di composizione (in spagnolo, una tabella per ognuna delle undici categorie).

http://www.fcf.usp.br/tabela/tbcamenu.php
Sito del db brasiliano, che sembra avere pochi dati e messi in formato html statico (anche questo sito è completamente in spagnolo, quindi difficile da comprendere).

http://www.ktl.fi/fineli/
Sito del database finlandese di composizione: è completamente in finlandese, a parte una pagina in inglese, che dà le istruzioni "di sopravvivenza" per visualizzare i dati on-line. Il db contiene 2500 alimenti e 290 nutrienti (di cui metà sono cibi semplici e l'altra metà complessi (ricette)). Su Internet sono consultabili però solo 39 nutrienti e 1633 alimenti e i valori dei nutrienti sul sito sono aggiornati al 2001.
Il db contiene tutti i nomi dei campi in finlandese, eccetto il nome dell'alimento, che è anche in inglese.

http://www.afssa.fr/ftp/basedoc/tablesaliments/Le_Ciqual.htm#ciqual
Sito del Centre Informatique sur la Qualité des Aliments (CIQUAL): contiene delle semplici tabelle statiche in francese (aggiornate al 2001), ognuna delle quali offre una lista degli alimenti ordinati in ordine crescente di contenuto di un particolare nutriente.

http://www.langual.org/
Sito di "Langua aLimentaria", un linguaggio di descrizione degli alimenti, che non descrive però la loro composizione.

1.8.2 Siti a Pagamento

http://food.ethz.ch/swifd/CHNWDB_it/HOME/home_it.htm
Sito della Banca Dati Svizzera di Composizione degli Alimenti. Il sito contiene tra l'altro due schede compositive (in formato pdf) di esempio (le schede sono in tedesco, con i nomi degli alimenti anche in italiano).
Il db contiene circa 700 prodotti che più rappresentano il consumo alimentare in Svizzera. Gli alimenti sono stati scelti in base a dati di mercato concernenti il consumo effettivo e a inchieste sulla nutrizione e sulle abitudini alimentari.

http://www.rsc.org/index.htm
Sito della Royal Society of Chemistry (UK), dove si può ordinare tra l'altro il libro "McCance and Widdowson's - The Composition of Foods Sixth Summary Edition", che contiene i valori nutrizionali di oltre 1200 alimenti consumati in UK. (si possono anche scaricare dei pdf di dimostrazione di tale libro).

http://www.sfk-online.net/cgi-bin/start.mysql?language=english
Sito del "Souci-Fachmann-Kraut Online-Database", database tedesco contente circa 800 alimenti e 260 componenti. Il sito è a pagamento e concede un periodo di prova gratuito di 30 giorni. Il sito comunque è un po' vecchio (l'ultimo

aggiornamento sembra risalire al 2000) e il database non è ovviamente scaricabile.

1.9 Fonti Selezionate

Tra le fonti menzionate ne sono state inizialmente selezionate quattro. La progettazione dei moduli di feeding ha comunque avuto tra gli obiettivi primari quello di garantire una facile estensione del numero di sorgenti di dati.

Le fonti selezionate sono quelle INRAN, IEO, NDL-USDA e ISA-CNR (per il significato di queste sigle si veda il paragrafo precedente): sono tutte istituzioni autorevoli, cosa che da sola garantisce una buona qualità delle informazioni, ed inoltre forniscono gratuitamente i propri dati.

Le prime due fonti sono state scelte perché contengono informazioni sui cibi effettivamente consumati in Italia e quindi sono di sicuro interesse in ambito nazionale.

La sorgente USDA è stata scelta perché è una vera e propria miniera di informazioni e contiene comunque molti alimenti che non sono troppo differenti da quelli consumati nel nostro paese.

Infine i dati ISA riguardanti i prodotti caseari campani sono stati selezionati in quanto caratterizzano una interessante realtà locale.

Tutte le fonti, tranne quella USDA (che è tutta in inglese), sono in buona parte in italiano. Una delle linee guida del progetto è quella di fornire un supporto per tradurre le parti esclusivamente in inglese in lingua italiana e viceversa. Ovviamente il lavoro di traduzione vero e proprio esula dagli scopi di questa tesi e deve comunque essere coadiuvato dalla presenza di un esperto del settore.

CAPITOLO DUE

REVERSE ENGINEERING DELLA FONTE INRAN

Inizia qui la serie di capitoli dedicati al reverse engineering delle varie fonti: questa fase è da sempre considerata uno dei passi principali per la comprensione degli schemi dei database e per la loro successiva integrazione.

2.1 Il Database

La base di dati (scaricata dal sito http://www.inran.it/) è in formato Access 97 e viene distribuita con un semplice programma di consultazione (entrambi gratuiti): il file Access sembra essere funzionale al programma, nel senso che molte tabelle contengono dati il cui scopo è l'utilizzo da parte del sw allegato (es. è presente una tabella con i nomi delle sezioni utilizzate nel programma, etc.). La base di dati è protetta da password.

Il database ospita informazioni nutrizionali su 790 alimenti consumati in Italia (si veda in seguito per la lista dei nutrienti presenti nel database). Le informazioni contenute in questa base di dati sono quasi esclusivamente in italiano (sono anche in inglese solo i dati di alcune appendici).

2.2 Il Software di Consultazione

Il software di interrogazione allegato al database INRAN è molto semplice da utilizzare e permette di consultare la tabella di un singolo alimento (come in un libro). Si può altresì fare una ricerca degli alimenti del db (o di una sola categoria) che contengono maggiormente un certo nutriente (o un certo intervallo di valori dello stesso).

L'applicativo permette di consultare, oltre al db Access, anche delle pagine html statiche con ulteriori informazioni (es. variazione di peso degli alimenti con la cottura, etc.).

2.3 Stato Iniziale del Database

Come si evince dal diagramma riportato di seguito, il db iniziale è più che altro un insieme di tabelle scollegate (non esistono vincoli di integrità referenziale) e

fortemente denormalizzate (in quanto le tabelle, come si è già detto, sembrano essere funzionali al "programma-libro" di consultazione). Si fa notare inoltre come tutti i campi siano di tipo testuale.

Nello schema riportato sono presenti solo le tabelle che contengono dati utili e non le tabelle di sistema né le tabelle create per utilizzo esclusivo da parte del software allegato.

La figura rappresenta lo schema concettuale ottenuto dal reverse engineering automatico fatto con il CASE PowerDesigner.

Composizione	
CAT_ID	A2
ALI_ID	A6
ALI_DESC	A200
NOME_SCIENT	A100
INFORMAZIONE_ALIM	A255
PER_EDI	A3
ACQUA	A10
PROTEINE	A10
NOTA_PROTEINE	A100
LIPIDI	A10
NOTA_LIPIDI	A100
COLESTEROLO	A10
CARBOIDRATI	A10
NOTA_CARBOIDRATI	A100
AMIDO	A10
NOTA_AMIDO	A100
ZUCC_SOL	A10
NOTA_ZUCC_SOL	A100
ALCOOL	A10
NOTA_ALCOOL	A100
FIBR_TOT	A10
FIBR_SOL	A10
FIBR_INS	A10
ENER_KCAL	A10
NOTA_KCAL	A100
ENER_KJOU	A10
PERC_PROT	A50
PERC_LIPI	A50
PERC_CARB	A50
PERC_ALCO	A50
SODIO	A10
POTASSIO	A10
FERRO	A10
NOTA_FERRO	A100
CALCIO	A10
FOSFORO	A10
Mg	A10
Zn	A10
Cu	A10
Se	A10
TIAMINA	A10
RIBOFLAVINA	A10
NIACINA	A10
VIT_A	A10
NOTA_VIT_A	A200
VIT_C	A10
NOTA_VIT_C	A200
VIT_E	A10
TOTA_SAT	A10
C4_0_C10_0	A10
C12_0	A10
C14_0	A10
C16_0	A10
C18_0	A10
C20_0	A10
C22_0	A10
TOTA_MON	A10
C14_1	A10
C16_1	A10
C18_1	A10
C20_1	A10
C22_1	A10
TOTA_POL	A10
C18_2	A10
C18_3	A10
C20_4	A10
C20_5	A10
C22_6	A10
RAPP_PS	A10
ACID_FIT	A10

Categorie	
CAT_ID	A2
CAT_DESC	A50

Alimenti	
CAT_ID	A2
ALI_ID	A6
ALI_DESC	A200

Aminoacidi	
ALI_ID	A6
Alimento	A200
Lisina	A10
Lisina1	A10
Istidina	A10
Istidina1	A10
Arginina	A10
Arginina1	A10
Ac_Aspa	A10
Ac_Aspa1	A10
Treonina	A10
Treonina1	A10
Serina	A10
Serina1	A10
Ac_Glut	A10
Ac_Glut1	A10
Prolina	A10
Prolina1	A10
Glicina	A10
Glicina1	A10
Alanina	A10
Alanina1	A10
Cistina	A10
Cistina1	A10
Valina	A10
Valina1	A10
Metionina	A10
Metionina1	A10
NOTA_metionina	A100
Isoleucina	A10
Isoleucina1	A10
Leucina	A10
Leucina1	A10
Tirosina	A10
Tirosina1	A10
Fenilalanina	A10
Fenilalanina1	A10
NOTA_Fenilalanina	A100
Triptofano	A10
Triptofano1	A10
IndChim	A10
AmiLimi	A10

2.4 Tabelle Escluse dal Processo di Reverse Engineering

Le tabelle non utilizzate dal processo di reverse engineering sono i cataloghi di sistema e le tabelle:

1) "Sezioni" (lista delle sezioni del software allegato).

2) "Appendici" (elenco delle appendici del sw allegato).

3) "TableDict" e "DataDict" (tabelle di metadatazione che non sono però quelle di sistema di Access, presenti probabilmente a causa di una migrazione da qualche altro DBMS).

E' da notare comunque che le tabelle del punto 3 sono state utilizzate per ricavare i nomi degli attributi del modello concettuale ristrutturato e le descrizioni dei nutrienti e delle unità di misura.

2.5 Considerazioni sull'Uso del Database Come Fonte per il Datawarehousing

Il db considerato è fornito con poca documentazione e quindi le considerazioni che seguono sono state tratte principalmente da un'analisi del database stesso e del suo software.

La base di dati è aggiornata al 2000 (e legata al sw di consultazione): questo è indice che gli aggiornamenti hanno frequenza bassa. Da ciò si evince una maggiore possibilità di cambiamenti radicali dello schema tra un aggiornamento e l'altro e quindi il modulo di feeding automatico potrebbe dover essere modificato per due versioni successive del database.

Servirebbe inoltre il parere di un esperto e maggiore documentazione per valutare la qualità dei dati: si ritiene comunque che, data l'autorevolezza della fonte, questi siano di buona qualità.

I dati sono forniti con poche indicazioni riguardo alle fonti secondarie: nella documentazione allegata si legge comunque "I dati riportati nel CD sono per il 70% dati originali ottenuti da studi programmati ad hoc e per il rimanente 30% provenienti da una accurata selezione bibliografica prevalentemente italiana. Le analisi sono state condotte nell'Unità di Chimica degli Alimenti dell'INRAN.".

Un aspetto sicuramente positivo del database è che contiene dati su cibi effettivamente consumati in Italia, cosa importante per un suo utilizzo per ricerche a livello nazionale.

2.6 Fonti di Dati INRAN Non Presenti nel Database

Sul sito dell'INRAN sono presenti degli articoli in formato pdf (es. carni.pdf) che presentano anche tabelle di composizione alimentare che potrebbero essere integrate col db, qualora non fossero già presenti in esso, con l'aiuto di un esperto.

Sempre sul sito è inoltre presente un file .doc con un errata corrige che è stato ovviamente integrato con i dati della base di dati.

Il programma di utilizzo del database fa uso anche di pagine html statiche: le informazioni presenti in molte di esse possono essere perfettamente integrate con quelle già presenti nel db (aggiungendo altri componenti o attributi quali "peso dopo la cottura", "nomi comuni", etc.).

2.7 Ristrutturazione dello Schema Concettuale

Di seguito riportiamo i diagrammi concettuali ristrutturati, creati sulla base dello schema concettuale iniziale visto in precedenza.

Essendo i codici di alcune colonne del diagramma iniziale poco significativi, sono stati sostituiti con le etichette ricavate dalla tabella di metadatazione "DataDict", al fine di rendere maggiormente leggibili gli schemi finali. La tabella seguente mostra le sostituzioni effettuate (che comunque hanno modificato le etichette e non il codice generato dal CASE):

ETICHETTA	CODICE
Alimento	ALI_DESC
Aminoacido Limitante	AMILIMI
Calorie da alcool (%)	PERC_ALCO
Calorie da carboidrati (%)	PERC_CARB
Calorie da lipidi (%)	PERC_LIPI
Calorie da proteine (%)	PERC_PROT
Categoria	CAT_DESC
Codice alimento	ALI_ID
Codice categoria	CAT_ID
Indice chimico	INDCHIM
Informazioni alimentari	INFORMAZIONE_ALIM
Nome scientifico	NOME_SCIENT
Parte edibile (%pe)	PER_EDI
Rapporto polinsaturi saturi	RAPP_PS

Questa operazione di aggiunta di nuovi nomi si è resa necessaria anche per la scarsa documentazione che accompagna il database.

Lo schema ER vero e proprio è nel diagramma "INRAN Final Conceptual", mentre il diagramma "Campi Eliminati" riporta la lista degli attributi dello schema concettuale iniziale che, a causa del processo di normalizzazione, non hanno un corrispettivo in nessun attributo del diagramma concettuale finale.

La normalizzazione si è resa necessaria sia perché nello schema iniziale erano presenti concetti mischiati tra di loro (es. componente e alimento), con le ovvie anomalie che questo comporta, sia per facilitare la successiva operazione di integrazione con gli altri database.

Le nuove entità inserite sono “Nutrienti”, che contiene informazioni generiche sui componenti, e “Note Nutriente”, che invece contiene delle note riguardanti il particolare valore di nutriente in un alimento.

Entrambe le entità aggiunte erano presenti come colonne nelle tabelle iniziali: nello schema “Campi Eliminati” vi sono due tabelle (“Nomi Nutrienti” e “Nomi Note Nutrienti”) che contengono le colonne rimosse dal diagramma iniziale e che sono occorrenze particolari delle due nuove entità create.

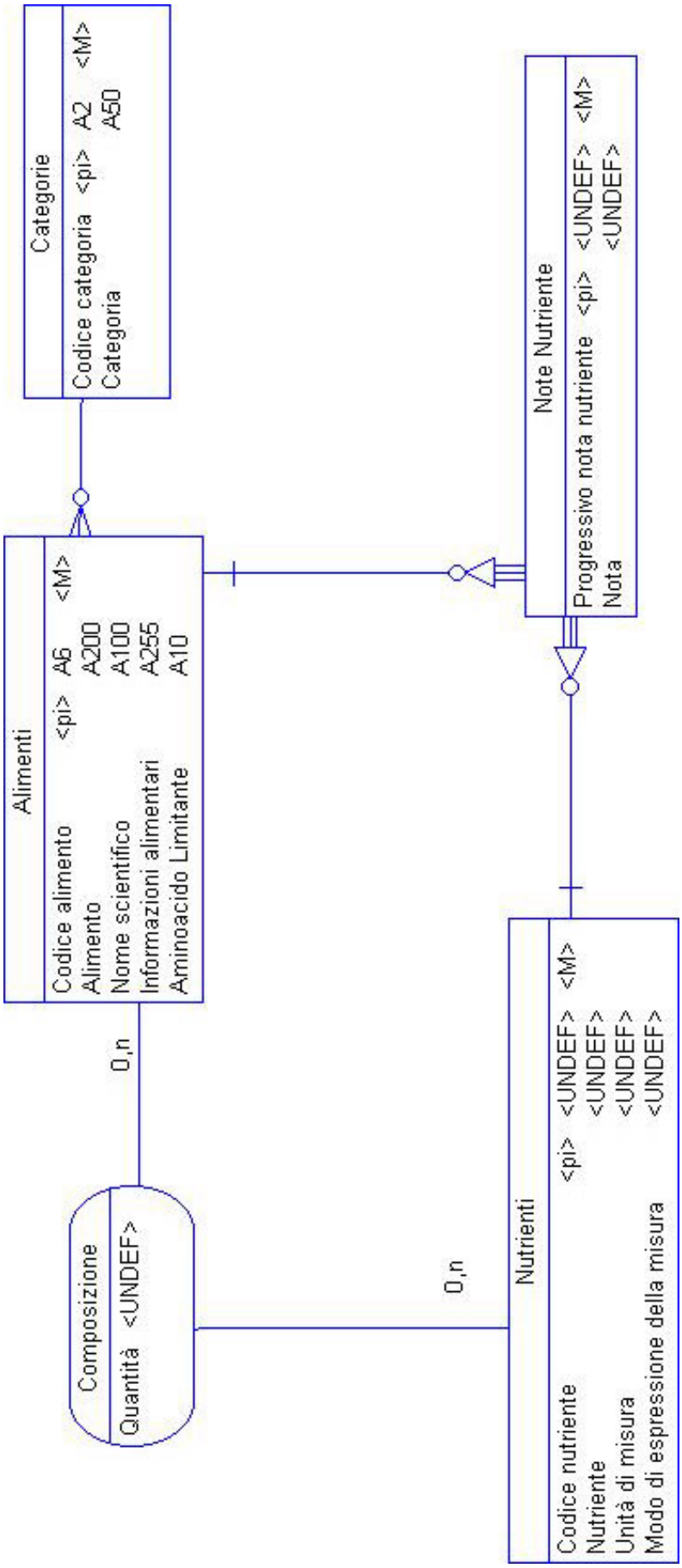
Sono state poi ovviamente costruite delle relazioni che legano tutte le entità e una di queste,”Composizione”, presenta anche l’attributo “Quantità”, che ovviamente denota l’ammontare di nutriente nell’alimento.

In fase di normalizzazione si è reso necessario creare anche degli identificativi per le nuove entità (“Nutrienti” e “Note nutriente”).

Si fa notare ancora che il campo "Unità di misura" dice l'unità (es. g, mg) con cui è espresso il nutriente, mentre il campo "Modo di espressione della misura" dice la quantità di alimento rispetto alla quale si riferisce la misura (es. 100g di parte edibile, 100g di proteine, etc.).

Tutti gli attributi inseriti ex novo non hanno definito un tipo (sono “<UNDEF>”), in quanto questa informazione verrà aggiunta in fase di integrazione nel solo diagramma di “unione” degli schemi di tutte le fonti.

2.7.1 Schema “INRAN Final Conceptual”



2.7.2 Schema “Campi Eliminati”

Nomi Nutrienti	
Triptofano	A10
Triptofano1	A10
ACQUA	A10
PROTEINE	A10
LIPIDI	A10
COLESTEROLO	A10
CARBOIDRATI	A10
AMIDO	A10
ZUCC_SOL	A10
ALCOOL	A10
FIBR_TOT	A10
FIBR_SOL	A10
FIBR_INS	A10
SODIO	A10
POTASSIO	A10
FERRO	A10
CALCIO	A10
FOSFORO	A10
Mg	A10
Zn	A10
Cu	A10
Se	A10
TIAMINA	A10
RIBOFLAVINA	A10
NIACINA	A10
VIT_A	A10
VIT_C	A10
VIT_E	A10
TOTA_SAT	A10
C4_0_C10_0	A10
C12_0	A10
C14_0	A10
C16_0	A10
C18_0	A10
C20_0	A10
C22_0	A10
TOTA_MON	A10
C14_1	A10
C16_1	A10
C18_1	A10
C20_1	A10
C22_1	A10
TOTA_POL	A10
C18_2	A10
C18_3	A10
C20_4	A10
C20_5	A10
C22_6	A10
ACID_FIT	A10
Lisina	A10
Lisina1	A10
Istidina	A10
Istidina1	A10
Arginina	A10
Arginina1	A10
Ac_Aspa	A10
Ac_Aspa1	A10
Treonina	A10
Treonina1	A10
Serina	A10
Serina1	A10
Ac_Glut	A10
Ac_Glut1	A10
Prolina	A10
Prolina1	A10
Glicina	A10
Glicina1	A10
Alanina	A10
Alanina1	A10
Cistina	A10
Cistina1	A10
Valina	A10
Valina1	A10
Metionina	A10
Metionina1	A10
Isoleucina	A10
Isoleucina1	A10
Leucina	A10
Leucina1	A10
Tirosina	A10
Tirosina1	A10
Fenilalanina	A10
Fenilalanina1	A10
ENER_KCAL	A10
ENER_KJOU	A10
Indice chimico	A10
Calorie da proteine (%)	A50
Calorie da lipidi (%)	A50
Calorie da carboidrati (%)	A50
Calorie da alcool (%)	A50
Rapporto polinsaturi saturi	A10
Parte edibile (%pe)	A3

Nomi Note Nutrienti	
NOTA_PROTEINE	A100
NOTA_LIPIDI	A100
NOTA_CARBOIDRATI	A100
NOTA_AMIDO	A100
NOTA_ZUCC_SOL	A100
NOTA_ALCOOL	A100
NOTA_FERRO	A100
NOTA_VIT_A	A200
NOTA_VIT_C	A200
NOTA_metionina	A100
NOTA_Fenilalanina	A100
NOTA_KCAL	VA100

2.8 Ulteriori Vincoli

Oltre ai vincoli visibili nello schema ristrutturato, ne sono stati individuati altri, che sono:

1) “Parte edibile (%)” <= 100

2) “Energia in kJ” = 4.184 * “Energia in kcal”

3) “Calorie da Proteine (%)” + “Calorie da Lipidi (%)” + “Calorie da Carboidrati (%)” + “Calorie da Alcool (%)” = 100

4) “Acqua (g)” + “Proteine (g)” + “Lipidi (g)” + “Carboidrati disponibili (g)” <= 100

5) “Amido (g)” + “Zuccheri solubili (g)” <= “Carboidrati disponibili (g)”

6) “Fibra insolubile (g)” + “Fibra solubile (g)” <= “Fibra totale (g)”

7) “Rapporto Polinsaturi/Saturi” = “Polinsaturi totali (g)” / “Saturi totali (g)”

8) “Monoinsaturi totali (g)” + “Polinsaturi totali (g)” + “Saturi totali (g)” <= “Lipidi (g)”

9) “C4:0-C10:0” + “C12:0” + “C14:0” + “C16:0” + “C18:0” + “C20:0” + “C22:0” <= “Saturi totali (g)”

10) “C14:1” + “C16:1” + “C18:1” + “C20:1” + “C22:1” <= “Monoinsaturi totali (g)”

11) “C18:2” + “C18:3” + “C20:4” + “C20:5”+ “C22:6” <= “Polinsaturi totali (g)”

Molti di questi vincoli non vengono comunque implementati nel sistema finale sia perché lo renderebbero più lento, sia perché a causa delle approssimazioni alcuni vincoli potrebbero perdere di validità.

2.9 Analisi delle Ridondanze

Nello schema finale sono presenti alcune istanze dell'entità "Nutrienti" che possono sembrare ridondanti, come ad esempio "Calorie da proteine (%)", "Calorie da lipidi (%)", etc.

Non essendo comunque in possesso degli elementi necessari per stabilire se ciò accade per ogni possibile istanza dell'entità, si è deciso di non eliminare nel database finale di integrazione tali dati.

CAPITOLO TRE

REVERSE ENGINEERING DELLA FONTE USDA

Descriviamo ora il processo di ristrutturazione della seconda fonte, quella USDA. Per favorire il reperimento di informazioni simili, si è cercato di mantenere il più possibile uguali i nomi dei paragrafi di questo capitolo e del precedente.

3.1 Il Database

La base di dati (scaricata dal sito http://www.nal.usda.gov/fnic/foodcomp) è in due formati: Access 2000 e ASCII.

Il database ha due semplici programmi di consultazione (entrambi gratuiti): un applicativo desktop per i sistemi operativi Windows o per personal digital assistant (PDA) e uno web based. E' presente una buona documentazione sia della base di dati che del sw allegato.

Il db (denominato “National Nutrient Database for Standard Reference, Release 16”) è aggiornato a luglio 2003 e contiene informazioni per 125 componenti alimentari e 6661 alimenti consumati negli Stati Uniti.

Le informazioni presenti sono esclusivamente in inglese.

3.2 Il Software di Consultazione

Il software di consultazione web based, disponibile on line allo stesso sito del db, permette di fare una veloce ricerca per stringa tra i nomi degli alimenti e di visualizzarne in seguito la scheda di composizione.

Tale scheda mostra le quantità di nutrienti per 100g di parte edibile o per altre unità di misura di uso comune a scelta (come una tazza o un cucchiaino).

Il programma di consultazione in ambiente Windows (e PDA) permette di fare la stessa ricerca per stringa, ma consente anche in seguito di restringere i risultati della ricerca scegliendo la sola categoria di cui si vogliono visualizzare i dati.

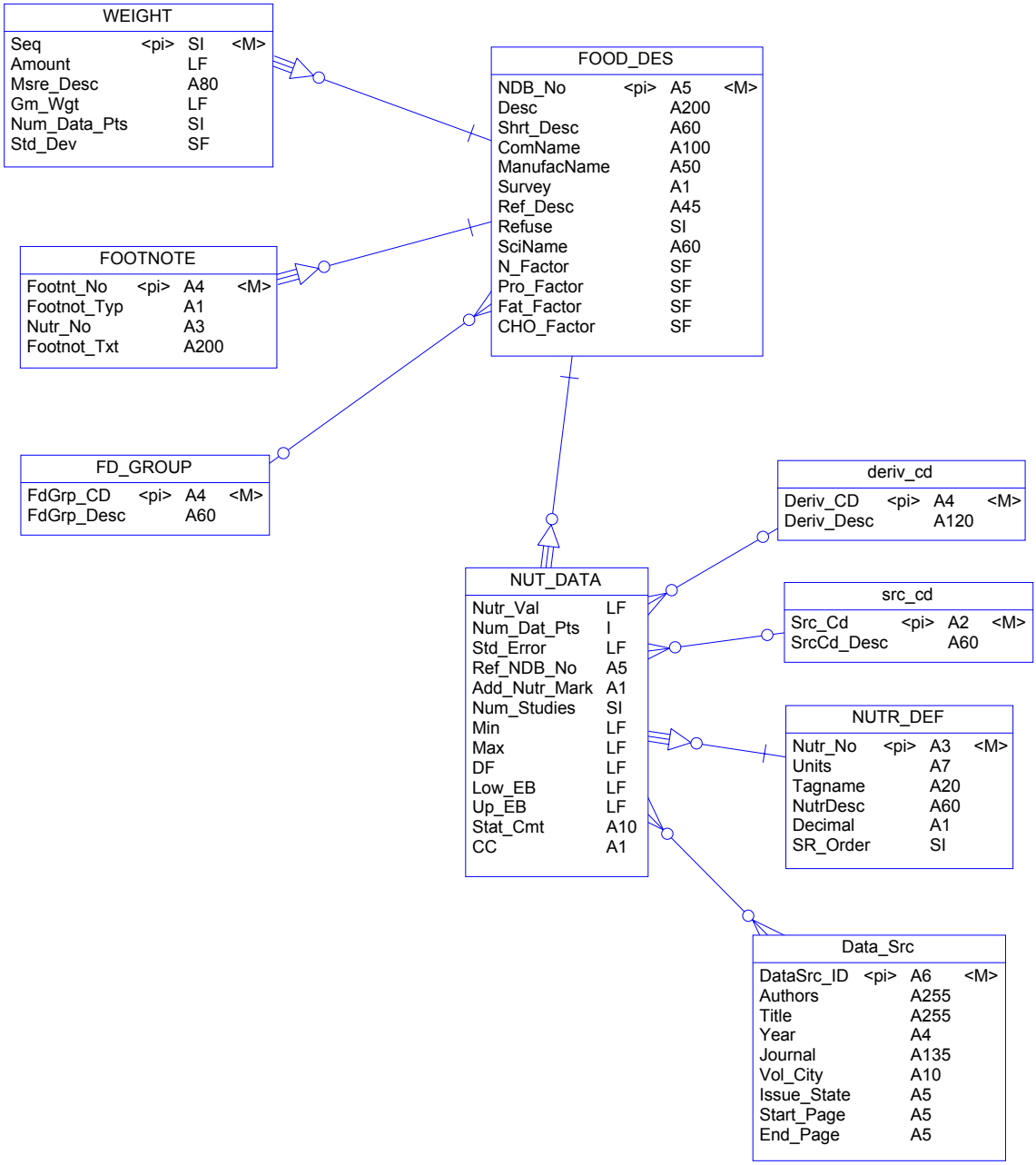
Inoltre, una volta mostrata la tabella di composizione dell'alimento scelto, l’applicativo desktop consente di cambiare la quantità a cui si riferiscono i valori (si può passare dal default di “100g di parte edibile” ad una qualsiasi quantità in grammi o ad unità di misura “comune”).

3.3 Stato Iniziale del Database

Nel diagramma seguente riportiamo lo schema concettuale iniziale del database, che, al di là di alcuni problemi illustrati nel seguito (si veda il paragrafo “Ristrutturazione dello schema concettuale”), si presenta già con una buona struttura.

Per la comprensione del significato degli attributi si veda l’ampia documentazione allegata alla base di dati ([3]).

La figura rappresenta, come al solito, lo schema concettuale ottenuto dal reverse engineering automatico fatto con il CASE PowerDesigner.



3.4 Tabelle Escluse dal Processo di Reverse Engineering

A parte le tabelle di sistema, l'unica altra tabella esclusa dal processo di reverse engineering (e quindi non presente nello schema precedente) è “ABBREV”, che contiene, per ogni alimento, un sunto delle informazione più importanti presenti nel database (come in una vista materializzata).

3.5 Considerazioni sull’Uso del Database Come Fonte per il Datawarehousing

La base di dati è aggiornata annualmente e vengono forniti ad ogni aggiornamento dei file “differenziali” in formato ASCII per le tabelle più importanti: quindi siamo di fronte ad una discreta fonte di datawarehousing, almeno dal punto di vista degli aggiornamenti. Una nota negativa è però data dal fatto che, saltuariamente, con gli aumenti di versione cambia anche lo schema del db e non ci è dato sapere se questi cambianti si ripercuotono anche nei file ASCII di aggiornamento.

Servirebbe il parere di un esperto e forse maggiore documentazione per valutare la qualità dei dati (anche se per molti dati sono fornite anche le fonti secondarie e il metodo di derivazione del valore). Si ritiene comunque che, data l’autorevolezza della fonte, i dati siano di buona qualità.

Un aspetto negativo del database è che contiene dati su cibi consumati negli Stati Uniti, anche se comunque molti di questi alimenti (seppur con differenze a volte significative nei componenti) sono consumati anche in Italia.

3.6 Fonti di Dati USDA-NDL Non Presenti nel Database

Sul sito USDA-NDL si possono scaricare dei report, in formato pdf, fatti sulla

base di dati: tali dati, essendo in realtà delle informazioni di riepilogo di quelli presenti nel database, sono ritenuti di scarso interesse.

Sempre sul sito è invece reperibile altro materiale compositivo riguardante studi più specifici su altri componenti alimentari e sui fattori di ritenzione degli stessi. Tali dati potrebbero essere integrati col db grazie anche all'ausilio di un esperto.

3.7 Ristrutturazione dello Schema Concettuale

Di seguito riportiamo i diagrammi concettuali ristrutturati, creati sulla base del diagramma concettuale iniziale visto in precedenza.

Si fa notare che, sebbene i nomi degli attributi degli schemi non siano abbastanza significativi (sono stati lasciati i codici dei campi della base di dati), un'ampia spiegazione di tali nomi può essere trovata nella documentazione allegata al database.

Lo schema ER vero e proprio è nel diagramma "USDA Final Conceptual", mentre il diagramma "Campi Eliminati" riporta la lista degli attributi dello schema concettuale iniziale che, a causa del processo di ristrutturazione, non hanno un corrispettivo in nessun attributo del diagramma concettuale finale.

Durante la ristrutturazione sono state introdotte due nuove entità: “NUT_FOOTNOTE” e “Stat_Cmt”. La prima contiene delle note riguardanti uno specifico valore di un componente in un alimento, ed è un’entità che è stata divisa dalle note riguardanti il singolo alimento, che restano invece presenti nell’entità “FOOTNOTE”.

L’entità “Stat_Cmt” contiene invece dei commenti statistici sul valore: questi commenti erano presenti solo nella documentazione allegata al database, mentre in quest’ultimo c’era solo un attributo di “NUT_DATA” contenente un elenco, separato da virgole, dei codici dei commenti che si applicano al dato valore.

In fase di normalizzazione si è reso necessario creare degli identificativi per le nuove entità (NUT_FOOTNOTE e Stat_Cmt).

Altra modifica allo schema è stata quella di ricavare dai campi "doppi" (nel senso che, a seconda del contesto, contengono una tra due informazioni completamente differenti) “Vol_City” e “Issue_State” dell’entità “Data_Src” altri quattro campi: “Volume”, “City”, “Issue” e “State”. Questo aumenta di molto i valori NULL, ma rende sicuramente più leggibile lo schema.

Si noti anche che l'attributo "Footnt_Typ" dell'entità "FOOTNOTE" era a tre valori: "D" indicava nota sull'alimento, "M" indicava nota sulla misura dell'alimento, "N" indicava nota sul nutriente. Il valore "N" è stato eliminato dallo schema ristrutturato perché le note sull'alimento e sulla misura sono rimaste in FOOTNOTE, mentre quelle sul nutriente sono state spostate in NUT_FOOTNOTE.

In "Campi Eliminati" sono presenti i seguenti attributi, che vengono riportati assieme alla motivazione della loro eliminazione:

- **"Survey":** campo flag che indica se l'alimento parteciperà o meno al sondaggio "National Health and Nutrition Examination Survey: What We Eat in America". Eliminato perché di scarso interesse e perché ridondante (un alimento partecipa al sondaggio se ha non nulli determinati nutrienti).

- **"Refuse":** percentuale di parte non edibile dell'alimento. Eliminato perché è meglio individuarlo come un'occorrenza particolare dell'entità "NUT_DEF".

- **"Stat_Cmt" e "FOOTNOTE.Nutr_No":** eliminati perché sono stati individuati come entità a sé stanti.

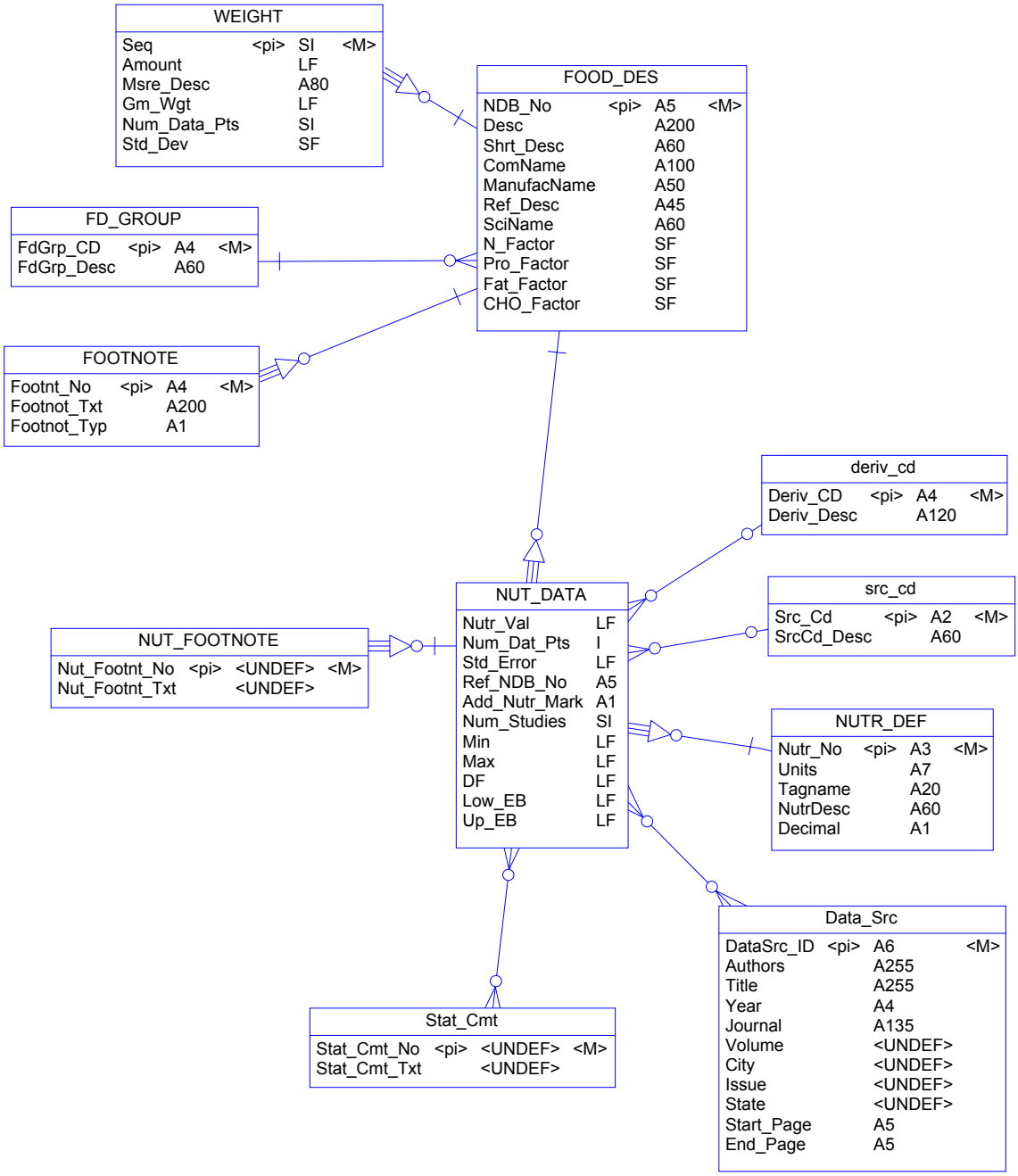
- **"SR_Order":** campo utilizzato per ordinare i record dei nutrienti nello stesso ordine nei vari report pubblicati. Eliminato perché application dependent.

- **"CC":** Confidence Code, dovrebbe indicare la qualità del dato. Eliminato perché ha tutti valori NULL e perché sembra poter essere calcolato in base ad altri campi (come numero di data point, metodo, etc.).

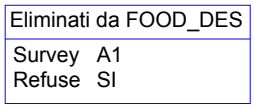
- **"Nutr_no":** eliminato da FOOTNOTE perché ora c'è un'entità a parte (NUT_FOOTNOTE) per le note sui nutrienti.

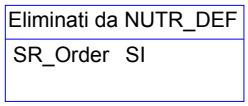
Come sempre, tutti gli attributi inseriti ex novo non hanno definito un tipo (sono "<UNDEF>"), in quanto questa informazione verrà aggiunta in fase di integrazione nel solo diagramma di "unione" degli schemi di tutte le fonti.

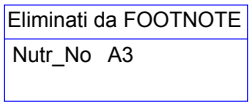
3.7.1 Schema “USDA Final Conceptual”

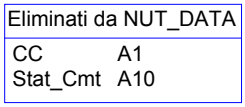


3.7.2 Schema “Campi Eliminati”









3.8 Problemi di Ristrutturazione Risolti Parzialmente

Gli attributi di testo “Deriv_Desc” e “SrcCd_Desc”, che sono campi descriventi le metodologie utilizzate per ricavare un valore, fanno riferimento ai codici presenti in “Src_Cd”.

Si dovrebbero “esplodere" tali codici presenti nei campi descrittivi (bisognerebbe cioè sostituire ai “Src_Cd” le “Src_Cd_Desc” corrispondenti), così da non dover dipendere da tali codici.

Questo lavoro non è automatizzabile e potrebbe richiedere il parere di un esperto, quindi viene rimandato ad una fase successiva.

3.9 Ulteriori Vincoli

Un primo vincolo, non presentato nello schema finale solo per motivi di leggibilità del diagramma stesso, è rappresentato dal fatto che l’attributo “Ref_NDB_No” dell’entità “NUT_DATA” deve avere, se diverso da NULL, un corrispettivo nel campo “NDB_No” di “FOOD_DES”.

Esistono poi vincoli simili a quelli di INRAN e IEO tra i nutrienti (si vedano i

paragrafi “Ulteriori Vincoli” relativi alle due fonti), che però vanno trovati e verificati con un esperto. Si fa comunque presente che nella documentazione USDA è scritto che, a causa delle approssimazioni, molti vincoli in cui dovrebbe comparire il simbolo di “minore o uguale” a volte non vengono rispettati.

3.10 Analisi delle Ridondanze

Nello schema finale non sembrano esserci dati ridondanti, anche se non è ben chiaro se il campo “Shrt_Desc” dell’entità “FOOD_DES” sia ricavato o meno in modo automatico dal campo “Desc” della stessa entità, cosa che potrebbe essere attuata utilizzando la lista di abbreviazioni presente nella documentazione allegata alla base di dati.

CAPITOLO QUATTRO

REVERSE ENGINEERING DELLE FONTI IEO E ISA-CNR

E' questo l'ultimo capitolo dedicato al reverse engineering: le due fonti esplorate nel seguito sono accomunate dal fatto che per entrambe, in questa fase, è stato adoperato quasi esclusivamente del materiale in struttura non relazionale.

4.1 REVERSE ENGINEERING DELLA FONTE IEO

4.1.1 Il Database

La base di dati utilizzata nella fase di reverse engineering (scaricata dal sito http://www.ieo.it/) è in realtà un libro (dal titolo “Banca Dati di Composizione degli Alimenti per Studi Epidemiologici in Italia”) in formato pdf.

In questa fase è stata utilizzata la versione “cartacea” e non il file ASCII corrispondente per il semplice motivo che quest’ultima si è resa disponibile solo in seguito. Comunque, a parte delle lievi differenze illustrate nei paragrafi successivi, le due versioni sono pressoché coincidenti.

La base di dati non sembra possedere programmi di consultazione.

Nella documentazione allegata (e sul sito web) si legge che il libro è stato pubblicato nel 1998 e contiene informazioni su 778 alimenti e 37 componenti alimentari: gli alimenti inclusi sono per lo più alimenti semplici, crudi e quelli più frequentemente consumati dalla popolazione italiana.

Sempre dalla documentazione si apprende che la banca dati può essere definita “compilativa”, nel senso che “i dati inseriti sono stati ricavati da fonti preesistenti (tabelle di composizione, articoli scientifici, etc.) e non da analisi eseguite ad hoc”.

Sebbene una delle fonti principali del database siano le tabelle dell’INN (Istituto Nazionale della Nutrizione, oggi INRAN), cioè una delle altre sorgenti selezionate per far parte della base di dati finale, i dati IEO possiedono comunque caratteristiche molto interessanti, ad esempio la presenza di componenti assenti in INRAN e la piccola percentuale di valori NULL.

Le informazioni presenti nella banca dati sono in italiano e in buona parte anche in inglese (nel file ASCII però sono solo in italiano).

4.1.2 Stato Iniziale del Database

Come si evince dalla figura riportata di seguito, il “cuore” della banca dati è un insieme di tabelle di composizione, una per ogni alimento.

Per la comprensione del significato dei campi e delle abbreviazioni si veda l’ampia documentazione allegata ([2]).

Composizione per 100g di parte edibile

AGNELLO [OVIS AGNUS] **codice 1060**

Categoria merceologica 10080

alimento interamente sostituito nota INN: pronto da cuocere, lo scarto e' costituito dalle ossa del busto

Componenti alimentari, unità	Valori	Fonte	Codice	Note e valori originali
Parte edibile, g	83	H		
Acqua, g	70.1	H		
Proteine totali, g	20.8	H		
Proteine animali, g	20.8	H		
Proteine vegetali, g	0.0	H		
Lipidi totali, g	8.8	H		
Lipidi animali, g	8.8	H		
Lipidi vegetali, g	0.0	H		
Acidi grassi:				
Saturi totali, g	4.15	02	APP	RIPR riferito a agnello (nota INN: valori medi relativi al totale della parte edibile) lip=30.5g saturi=14.4g 18:1=10.89g mono=11.26g 18:2=0.71g 18:3=0.71g altri PUFA=0 PUFA=1.42g
Acido oleico, g	3.14	02	APP	
Monoinsaturi totali, g	3.25	02	APP	
Acido linoleico, g	0.20	02	APP	
Acido linolenico, g	0.20	02	APP	
Altri polinsaturi, g	0.00	02	APP	
Polinsaturi totali, g	0.41	02	APP	
Colesterolo, mg	71	02	APP	riferito a Agnello parte edibile totale cruda
Glucidi Disponibili, g	0.0	H		
Amido, g	0.0	H		
Glucidi solubili, g	0.0	H		
Fibra alimentare, g	0.0	H		
Alcool, g	0.0	H		
Energia, kcal	162	H		
Energia, kJ	678	86		
Minerali:				
Ferro, mg	1.6	H		
Calcio, mg	7	H		
Sodio, mg	88	H		
Potassio, mg	350	H		
Fosforo, mg	190	H		
Zinco, mg	3.3	T	96	per tutti i nutrienti fonte T cod 96 ALIMENTO SIMILE Lamb, average, trimmed lean, raw acqua=70.6g, prot=20.2g, lip=8.3g
Vitamine:				
Tiamina, mg	0.14	H		
Riboflavina, mg	0.28	H		
Niacina, mg	6.0	H		
Vitamina C, mg	0	H		
Vitamina B6, mg	0.30	T	96	
Acido Folico, µg	6	T	96	
Retinolo eq, µg	tr	H		
Retinolo, µg	tr	H		CALC
ß-carotene eq, µg	0	T	96	
Vitamina E, mg	0.09	T	96	non RIPR
Vitamina D, µg	0.40	T	96	non RIPR

S Salvini, M Parpinel, P Gnagnarella, P Maisonneuve, A Turrini 388

4.1.3 Considerazioni sull'Uso del Database Come Fonte per il Datawarehousing

Come si deduce dalla documentazione, siamo di fronte ad una base di dati "statica", nel senso che non sembrano previsti ulteriori aggiornamenti. Servirebbe comunque il parere di un esperto per valutare la qualità dei dati, che al momento è ritenuta buona semplicemente basandosi sull'autorevolezza della fonte.

Anche questo database, al pari di quello INRAN, contiene dati su cibi effettivamente consumati in Italia, cosa importante per un suo utilizzo per ricerche a livello nazionale.

4.1.4 Fonti di Dati IEO Non Presenti nel Database

Sul sito IEO è presente un file .pdf con un errata corrige: i dati nel file ASCII hanno già apportate queste modifiche.

Oltre ai dati contenuti nelle tabelle di composizione, ci sono molti dati nelle appendici che possono essere perfettamente integrati col database. Ad esempio, in appendice sono presenti le descrizioni in inglese degli alimenti: tali informazioni sono assenti nel file ASCII e dovrebbero quindi essere inserite manualmente in un secondo momento.

4.1.5 Creazione dello Schema Concettuale

Di seguito riportiamo il diagramma concettuale finale, creato sulla base dell'esame di svariate tabelle di composizione del libro.

Lo schema ER è denominato "IEO Final Conceptual": in fase di creazione dello schema, i dati delle schede presenti nella versione pdf sono stati calati nella struttura relazionale mostrata nella figura successiva. La corrispondenza tra i campi delle

schede e quelli del modello relazionale è abbastanza immediata e viene data una spiegazione solo per gli attributi il cui nome potrebbe non essere autoesplicativo.

Innanzitutto si è reso necessario creare degli identificativi per le entità che non avevano codici nella fonte (“Nutrienti” e “Categorie”).

Si nota inoltre che l'entità "Categorie" è ricavata dall'indice della documentazione allegata ([2], pag. 1 e 2) e non è altro che una classificazione meno specifica rispetto a quella fornita da "Categorie Merceologiche".

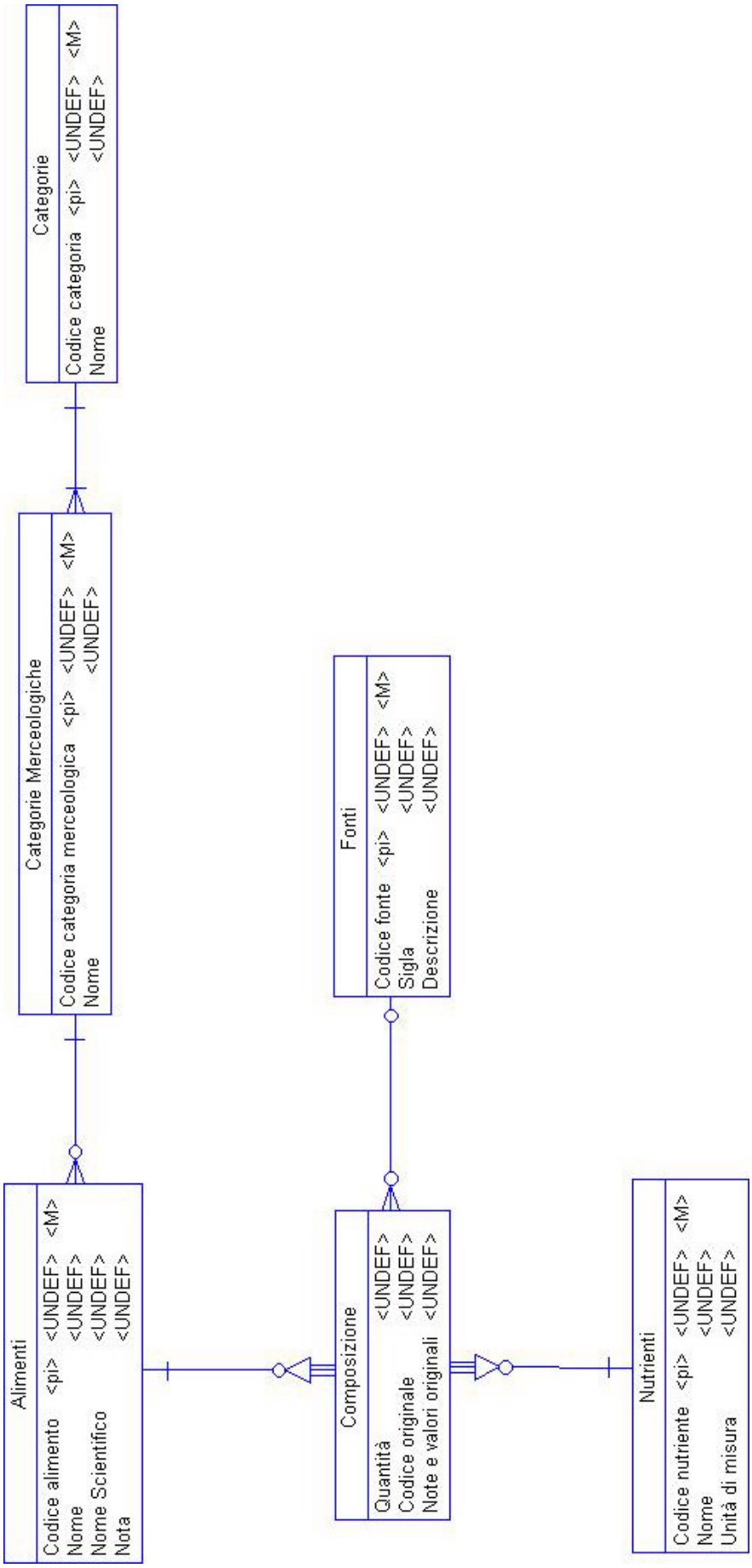
L’attributo “Codice Originale” di “Composizione” non è altro che l’informazione presente nella colonna “Codice” delle schede originali: indica il codice dell’alimento da cui è stato ricavato il valore in questione.

Altro attributo che merita un commento è “Nota” dell’entità “Alimenti”: contiene il testo presente sotto la categoria merceologica nelle tabelle IEO (si veda l’esempio presentato in precedenza).

Esistono poi informazioni che non vengono prese direttamente dalle schede: il campo ”Nome” delle “Categorie Merceologiche” si ricava dall'Appendice D di [2], mentre i campi “Sigla” e “Descrizione” di “Fonti” possono essere presi dalla Bibliografia a pag.954 e seguenti di [2].

Infine è da notare che tutti gli attributi non hanno definito il tipo (sono “<UNDEF>”), in quanto questa informazione, ovviamente non presente nella fonte, verrà aggiunta in fase di integrazione nel solo diagramma di “unione” degli schemi di tutte le fonti.

4.1.5.1 Schema “IEO Final Conceptual”



4.1.6 Informazioni Non Presenti nel File ASCII

Nel file ASCII utilizzato per fare il feeding della datawarehouse finale non sono presenti tutti gli attributi visti in precedenza. In particolare mancano l'attributo "Nota" di "Alimento" e i campi "Codice originale" e "Note e valori originali" di "Composizione".

Queste informazioni non sono state quindi inserite in modo automatico nella datawarehouse, anche se comunque sono state tenute in conto nella creazione dello schema della stessa, in quanto potrebbero essere aggiunte in un secondo momento.

4.1.7 Ulteriori Vincoli

I vincoli che seguono sono presi dalla documentazione ([2], pag.36) e vengono riportati senza alcuna modifica alla scrittura originale:

1) Somma degli acidi grassi saturi, monoinsaturi e polinsaturi minore dei lipidi totali.

2) Acido oleico minore o uguale ai monoinsaturi totali.

3) Somma di acido linoleico, linolenico e degli altri acidi grassi polinsaturi minore o uguale ai polinsaturi totali.

4) Somma dei glucidi solubili e amido uguale ai glucidi disponibili.

5) Somma delle proteine (o lipidi) vegetali ed animali, uguali alle proteine (o lipidi) totali.

6) Retinolo equivalente uguale alla somma del retinolo più un sesto del ß-carotene.

I vincoli seguenti sono invece abbastanza banali e sono stati ricavati da una

semplice analisi delle schede:

1) “Parte edibile, g” <= 100.

2) “Energia, kJ” = 4.184 * “Energia, kcal”.

3) “Acqua, g” + “Proteine totali, g” + “Lipidi totali, g” + “Glucidi disponibili, g” <= 100.

Come sempre, non tutti questi vincoli vengono in realtà implementati, principalmente per motivi di prestazioni. Trattandosi di una datawarehouse, infatti, risulta molto più conveniente gestire questi vincoli con controlli una tantum.

4.1.8 Analisi delle Ridondanze

L'unico dato che sembra essere ridondante è il nutriente "Retinolo equivalente" che, come si evince dal paragrafo "Ulteriori vincoli" (e ovviamente dalla documentazione allegata alle schede), è calcolato in base ad altri nutrienti.

4.2 REVERSE ENGINEERING DELLA FONTE ISA-CNR

4.2.1 Il Database

La base di dati (scaricata dal sito http://www.isa.cnr.it/PRODTIP/) è in realtà una ricerca cartacea di cui viene distribuita gratuitamente una scansione in immagini jpg: non è presente alcun programma di ricerca, eccettuato il sito tramite il quale si possono richiamare le suddette immagini.

Nel database sono presenti informazioni nutrizionali su prodotti caseari campani: sono disponibili sia varie descrizioni, sia quattro pagine di composizione chimica. Di queste ultime le prime tre sono di composizione nutrizionale, mentre la quarta contiene il ciclo di produzione dell'alimento. Nel seguito saranno quindi considerati solo i dati presenti nelle tre pagine in questione.

Le informazioni presenti sono esclusivamente in italiano.

4.2.2 Stato Iniziale del Database

Nelle tre immagini successive si vede un esempio di scheda di un singolo alimento. Da tale lavoro cartaceo è stato ricavato lo schema ER ristrutturato.

N.0R.CM

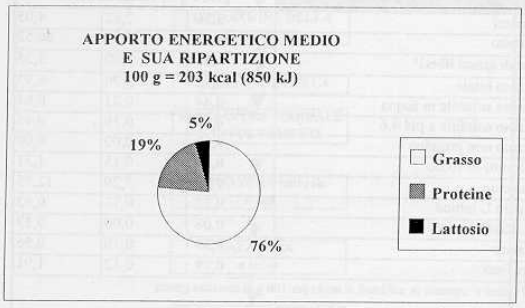
Ricotta

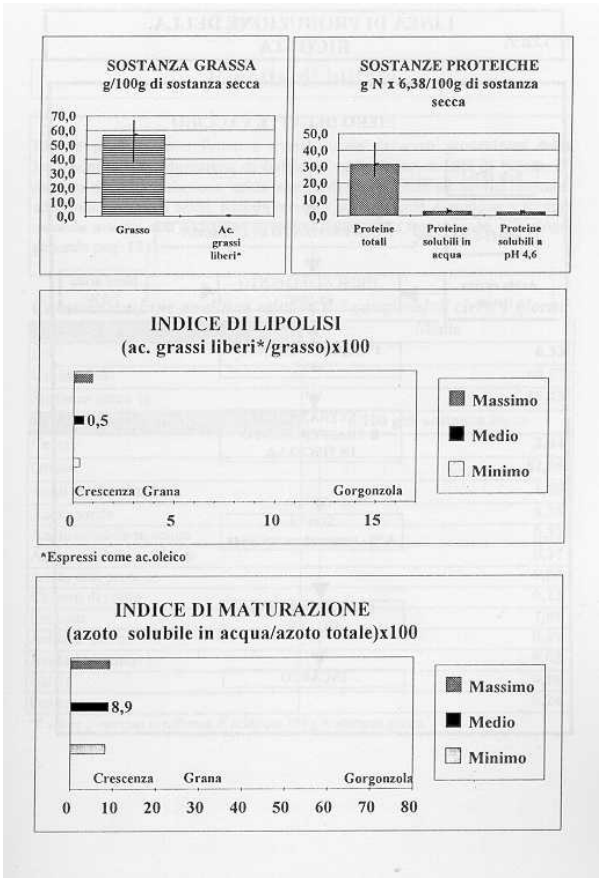
Ricotta prodotta tutto l'anno a livello regionale da siero proveniente dalla lavorazione della Mozzarella ed addizionato di latte. E' ottenuta dalla coagulazione acido-termica di siero, intero ad acidità naturale, addizionato di siero acido o talvolta di acido citrico o acido lattico. Ricotta lavorata manualmente, sottoposto a spurgo naturale e salata con addizione di sale al siero; senza maturazione. (Catalogo generale pag. 187)

Caratterizzazione analitica relativa a 3 campioni di 1 giorno

	Media	Minimo	Massimo
pH	**6,25**	5,87	6,48
Umidita' %	**69,55**	67,28	73,32
Sostanza secca %	**30,45**	26,68	32,72
	g/100 g di sostanza secca		
Ceneri	**3,30**	2,82	4,05
Grasso	**56,67**	38,50	66,52
Acidi grassi liberi*	**1,89**	1,20	3,35
Azoto totale	**4,93**	3,76	6,93
Azoto solubile in acqua	**0,44**	0,31	0,64
Azoto solubile a pH 4.6	**0,35**	0,19	0,45
Azoto non proteico	**0,00**	0,00	0,00
Cloruro di sodio	**0,84**	0,15	1,31
Lattosio	**7,82**	3,29	12,35
Acido L-lattico	**0,27**	0,11	0,43
Acido D-lattico	**0,08**	0,00	0,13
Calcio	**0,39**	0,19	0,56
Fosforo	**0,79**	0,12	1,91

*Il valore e' espresso in millimoli di acido per 100 g di sostanza grassa





4.2.3 Considerazioni sull'Uso del Database Come Fonte per il Datawarehousing

I dati sono presentati come un lavoro esclusivamente cartaceo e non sembrano essere previsti aggiornamenti periodici. Quindi non ha senso costruire un modulo di feeding automatico per tale sorgente.

Servirebbe il parere di un esperto (e forse maggiore documentazione) per valutare la qualità dei dati. Si ritiene comunque che, data l'autorevolezza della fonte, i dati siano di buona qualità.

Un aspetto certamente positivo del database è che contiene dati su cibi effettivamente prodotti in Campania, caratterizzando così un'importante realtà locale:

gli alimenti considerati vengono comunque ampiamente consumati anche a livello nazionale, cosa che dà alla fonte ancora maggior rilievo.

4.2.4 Creazione dello Schema Concettuale

Di seguito riportiamo il diagramma concettuale ristrutturato, creato sulla base dell'analisi delle schede cartacee di diversi alimenti.

Lo schema ER è denominato "ISA-CNR Final Conceptual": durante la fase di creazione di quest'ultimo, i dati delle immagini prelevate dal sito sono stati calati nella struttura relazionale mostrata nella figura successiva.

L'entità "Alimento" contiene informazioni prelevate dalla prima immagine (o pagina) relativa a ogni cibo: l'attributo "Nome alimento" è il titolo della prima pagina, mentre "Nota" è il testo immediatamente successivo. I campi "Numero campioni" ed "Età campioni" contengono poi i dati presenti nel rigo immediatamente precedente alla tabella compositiva di pagina uno (per l'alimento di esempio valgono rispettivamente "3" e "1 giorno").

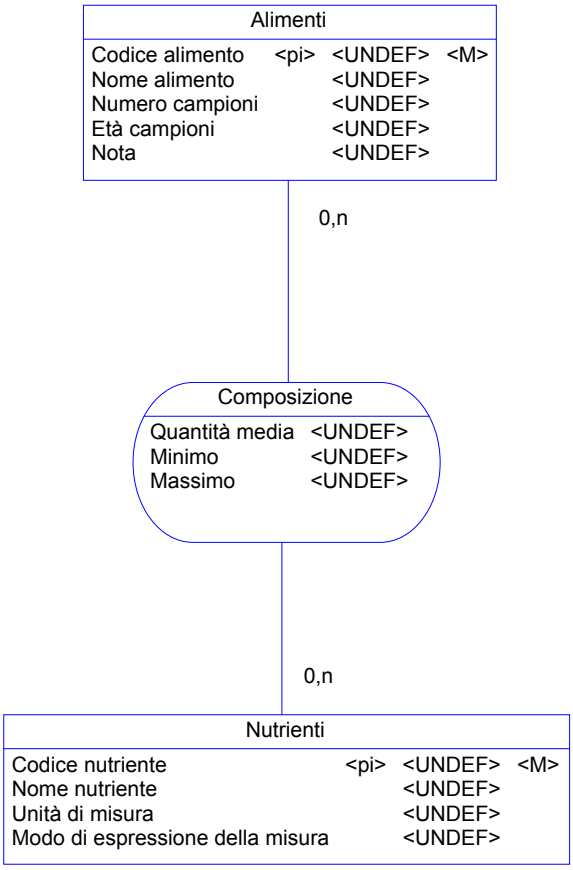
In fase di normalizzazione si è reso poi necessario creare degli identificativi per tutte le entità create ("Alimenti" e "Nutrienti").

L'entità "Nutrienti" contiene informazioni generiche (valide cioè per qualsiasi alimento) sui singoli componenti presenti a pag.1,2 e 3: il campo "Unità di misura" dice l'unità (es. "g", "millimoli") con cui è espresso il nutriente, mentre il campo "Modo di espressione della misura" dice la quantità di alimento rispetto alla quale si riferisce la misura (es. "100g di parte edibile", "100g di sostanza secca", etc.).

L'associazione "Composizione" contiene invece informazioni sullo specifico valore: oltre alla media, sono presenti anche due campi per il minimo e il massimo, che però non sono riempiti per tutti i valori, ma solo per quelli della prima pagina.

Infine è da notare che, anche in questo caso, tutti gli attributi non hanno definito il tipo (sono "<UNDEF>"), in quanto questa informazione, ovviamente non presente nella fonte, verrà aggiunta in fase di integrazione nel solo diagramma di "unione" degli schemi di tutte le fonti.

4.2.4.1 Schema “ISA-CNR Final Conceptual”



4.2.5 Analisi delle Ridondanze

Nello schema finale sono presenti alcune istanze dell’entità “Nutrienti” che possono sembrare ridondanti, come ad esempio le percentuali di calorie (da proteine, da grasso, etc.), le sostanze proteiche, l’indice di lipolisi e l’indice di maturazione.

Non essendo comunque in possesso degli elementi necessari per stabilire se ciò accade per ogni possibile istanza dell’entità, si è deciso di non eliminare nel database finale di integrazione tali dati.

CAPITOLO CINQUE

IL DATABASE DI INTEGRAZIONE

Inizia qui la fase di forward engineering, cioè la costruzione del sistema finale. Questo importantissimo capitolo descrive in cuore dell'applicazione in esame: i dati.

5.1 Le Entità Fondamentali

Dall'analisi degli schemi ristrutturati relativi alle quattro fonti (quelli che nei capitoli precedenti sono stati denominati "NOME_FONTE Final Conceptual") si ricava facilmente che le entità principali in gioco sono **"Alimento"**, **"Componente"** e **"Valore"** (per il significato di questi nomi si vedano le definizioni date nel primo capitolo).

Affianco a queste va ovviamente inserita una entità creata appositamente per tenere traccia della sorgente del singolo dato: **"Fonte primaria"**.

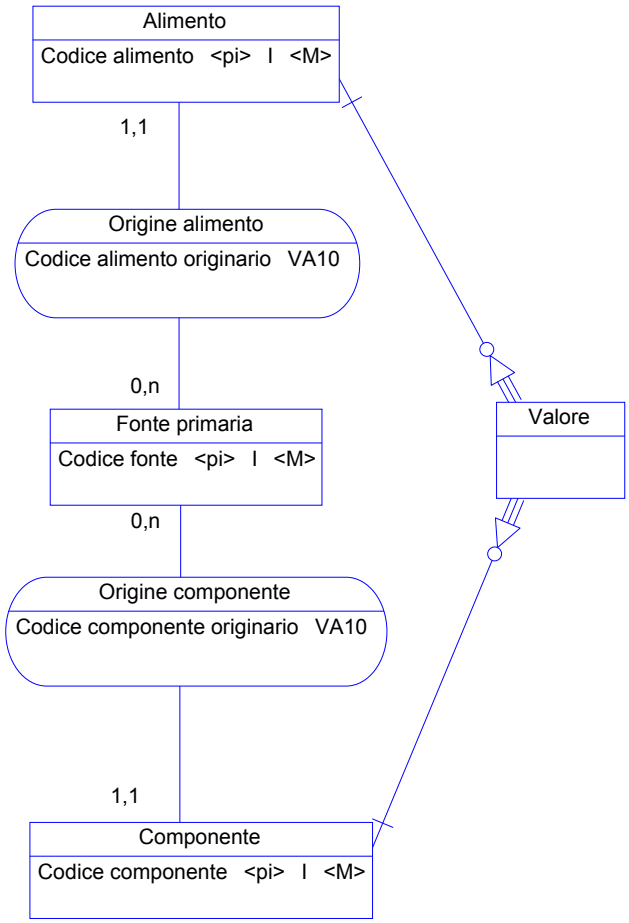
Insieme, questi quattro pezzi di informazione esprimono il concetto basilare dello schema, che si può così brevemente riassumere: **"L'alimento X, secondo la fonte primaria Y, ha un valore Z del componente W"**. In prima battuta infatti le restanti informazioni (es. categorie, note, etc.) possono essere considerate come semplice metadatazione di una delle suddette entità fondamentali.

Si noti poi che i nomi che vengono dati alle entità in questo schema integrato sono stati assegnati seguendo le definizioni al capitolo uno, mentre nei singoli schemi ristrutturati delle fonti tali nomi possono differire. Questo perché si è cercato, in fase di reverse engineering, di non discostarsi troppo dalle denominazioni degli schemi originali e/o della documentazione allegata agli stessi.

Per attuare le doverose corrispondenze tra lo schema di questo capitolo e quelli dei precedenti si tenga quindi in conto che quello che qui viene chiamato "Componente" è sinonimo di "Nutriente" ("NUT_DEF" in USDA), mentre "Valore" ha potuto assumere in precedenza il nome "Composizione" ("NUT_DATA" in USDA).

Per non generare confusione, nel grafico vengono presentati i soli attributi essenziali. Questo diagramma è una semplice base su cui viene poi costruito il ben più complesso schema finale.

Nella figura si possono notare anche le associazioni tra le entità: due di queste ("Origine alimento" e "Origine Componente") contengono anche degli attributi, che indicano rispettivamente il codice che avevano nella fonte primaria l'alimento e il componente.

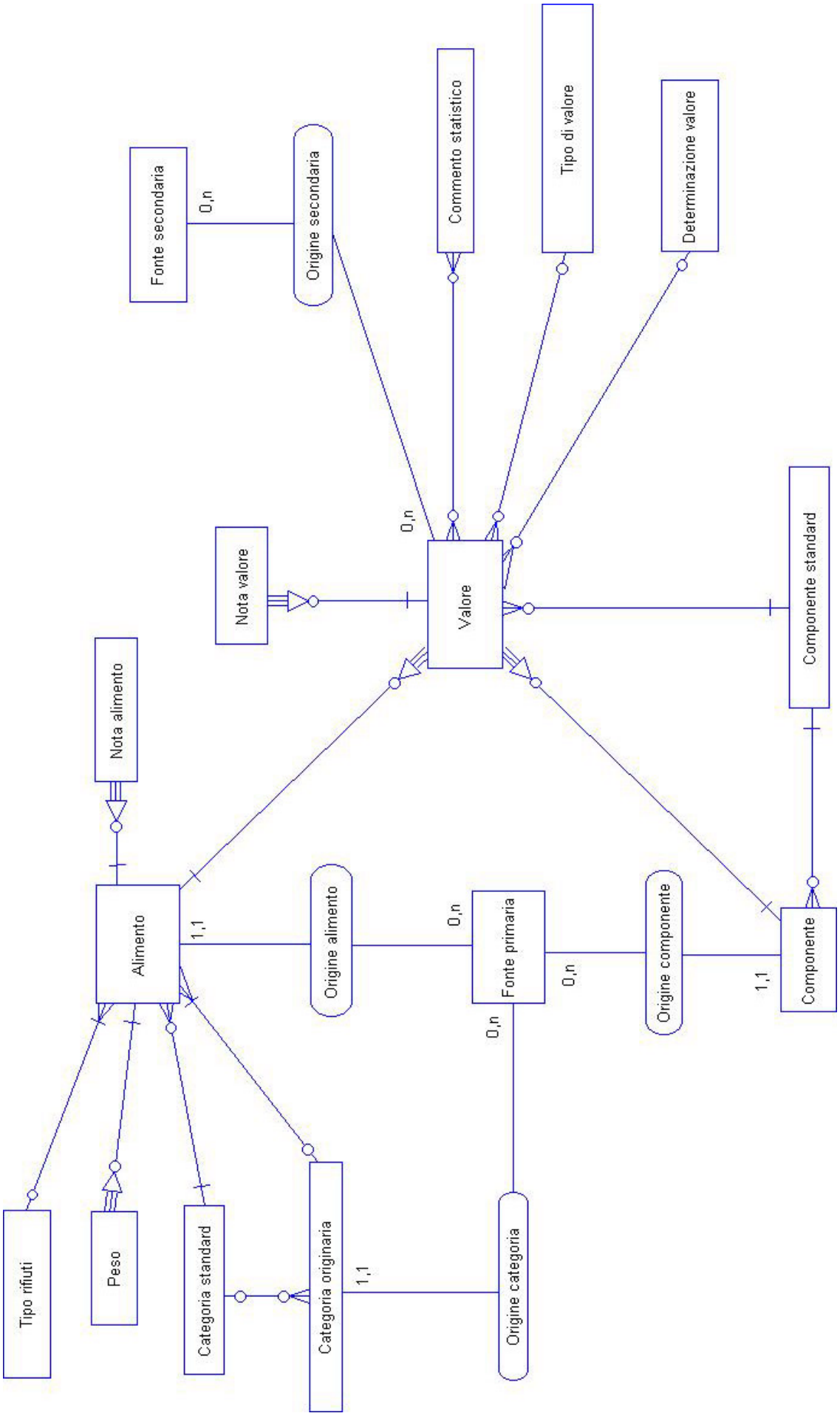


5.2 Le Altre Entità

Sono state poi aggiunte tutte le entità atte ad ospitare le informazioni presenti nei vari schemi ristrutturati delle singole fonti.

La figura seguente mostra molto bene come ognuna di queste ulteriori informazioni vada a specificare meglio i dati di una o più entità “fondamentali”. A differenza di queste ultime, presenti in tutte le fonti, non tutte le altre entità contengono informazioni provenienti da tutte le sorgenti di dati selezionate: molte anzi provengono dalla sola fonte USDA.

Come già anticipato nelle linee guida presentate nel capitolo uno, si è cercato di non perdere nessuna informazione riguardo ai database iniziali, pur sapendo che questo comporterà attributi con molti valori NULL.


Fonte secondaria
0,n
Origine secondaria
Commento statistico
Tipo di valore
Determinazione valore
Nota valore
0,n
Valore
Componente standard
Nota alimento
Alimento
1,1
Origine alimento
0,n
Fonte primaria
0,n
Origine componente
1,1
Componente
Tipo rifiuti
Peso
Categoria standard
Categoria originaria
1,1
Origine categoria

Anche se il significato di molte delle entità presentate nel diagramma può essere facilmente intuibile dal loro nome, per avere più informazioni si veda il paragrafo “Informazioni sulle singole entità” in questo stesso capitolo: quello che si vuol segnalare ora è la struttura di base dello schema.

5.2.1 Le Entità “Standard”

Occorre notare inoltre che, nella figura precedente, sono state introdotte due entità i cui dati non provengono da nessuna fonte: “Categoria standard” e “Componente standard”. La presenza di questi due nuovi concetti è dovuta alla necessità di confrontare i dati delle singole fonti.

Ci si ricorderà infatti che ogni fonte porta con sé i propri componenti e le proprie categorie di alimenti: questi vengono quindi visti come diversi anche quando in realtà esprimono informazioni molto simili (es. un software che consulta il database assume come due componenti distinti le proteine provenienti da due fonti differenti). Per permettere i confronti sono quindi state introdotte delle semplici etichette aggiuntive nella forma delle due entità “standard” (es. i componenti “proteine” di due fonti diverse sono entrambi associati allo stesso componente standard).

Gli standard utilizzati sono la classificazione dei cibi "Eurocode 99/2" (solo i gruppi principali) e la denominazione dei componenti del vocabolario standardizzato "COST Action 99 - EUROFOODS" (si vedano rispettivamente le fonti bibliografiche [6] e [1]).

Bisogna far notare che, nell’ambito nutrizionale, non esistono standard universalmente adottati e quindi quelli utilizzati sono stati scelti principalmente per la loro flessibilità e facilità di utilizzo: un’alternativa al vocabolario EUROFOODS sarebbero stati infatti i cosiddetti tagname INFOODS (si veda il sito relativo e [8]), che però risultano molto meno pratici da gestire.

In ogni caso si tenga presente che le corrispondenze tra componenti originari e standard (e tra categorie originarie e standard) sono del tutto sperimentali, in quando non sono state ancora validate da un esperto. Le suddette correlazioni sono comunque facilmente modificabili e si incoraggiano anzi gli utilizzatori dei dati a segnalare prontamente qualsiasi errore presente.

Data la provvisorietà di queste informazioni, esse non vanno assolutamente utilizzate per scambiare informazioni tra questo database e altri. Qualsiasi software che si appoggi sulla base di dati dovrebbe limitare l'utilizzo degli "standard" al solo loro scopo iniziale, cioè quello di semplice confronto (sperimentale) tra dati di fonti diverse.

Va detto poi che il vocabolario EUROFOODS è stato anche opportunamente allargato per comprendere alcuni componenti presenti nelle fonti ma non nello standard.

5.3 Il Supporto Multilingua

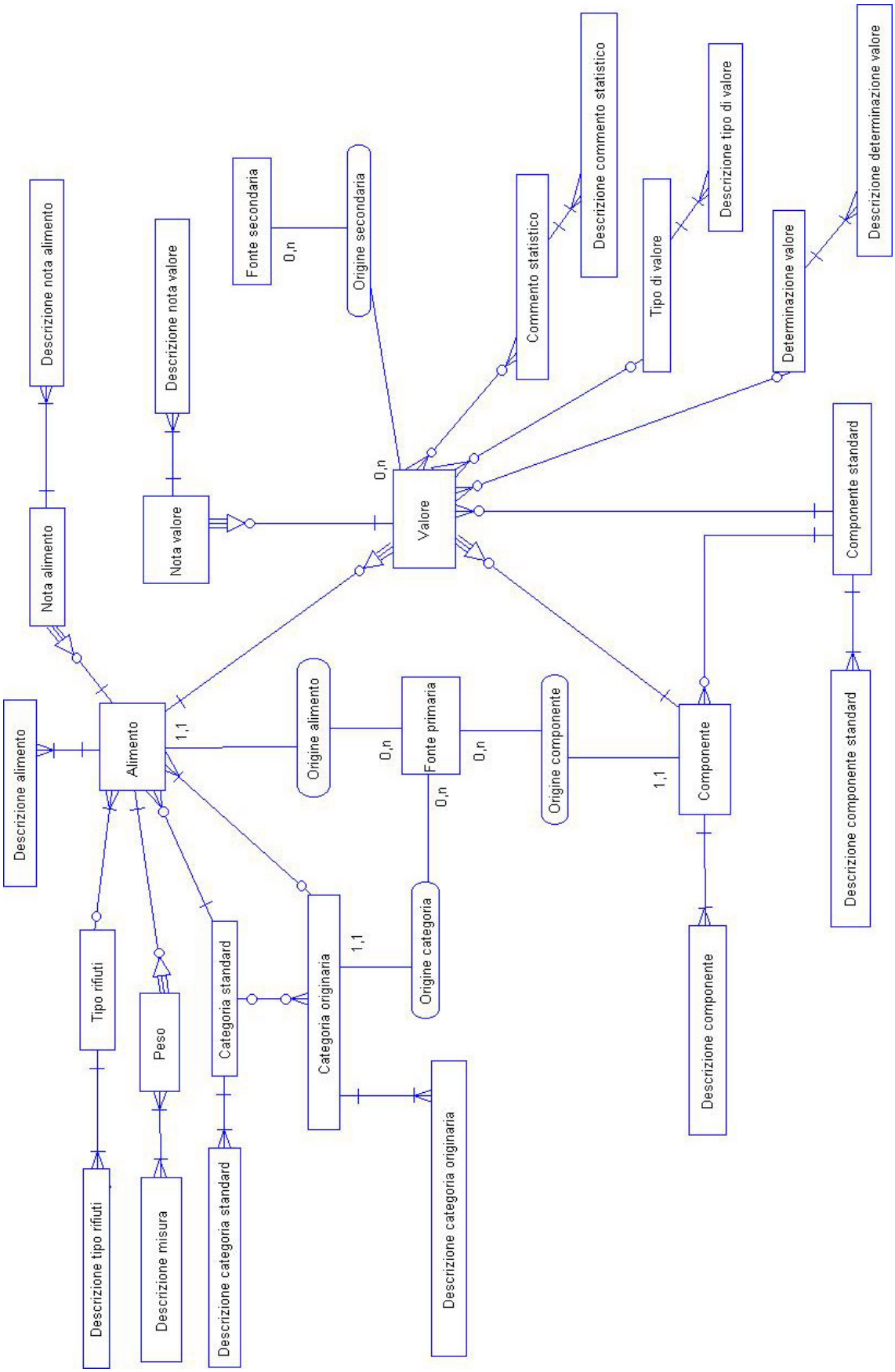
Sono state infine aggiunte molte entità (praticamente tutte quelle il cui nome inizia per "Descrizione") per fornire un supporto multilingua (inizialmente solo italiano e inglese).

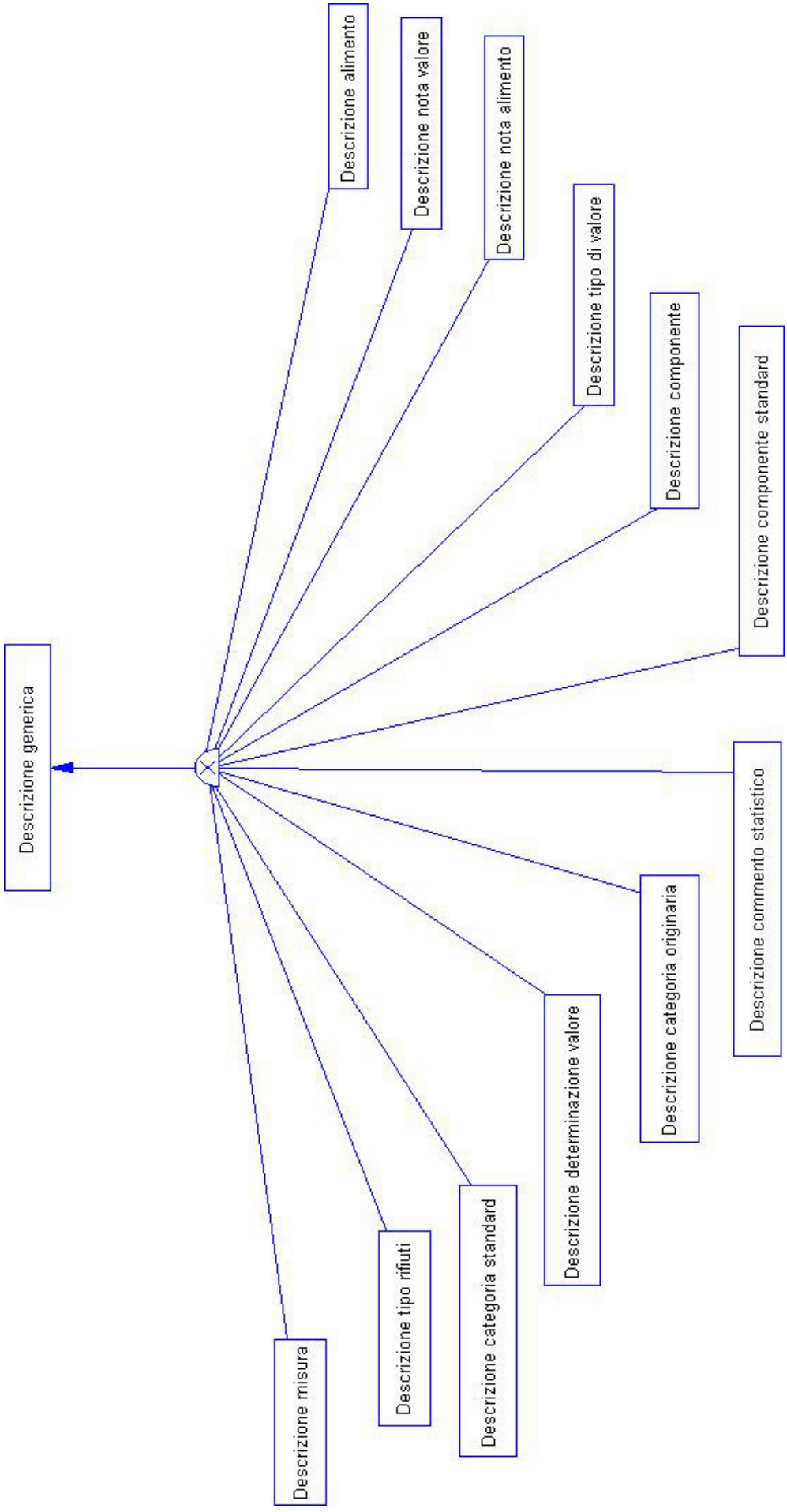
Ogni volta che ci si è trovati di fronte ad un testo che si voleva presente nel database in più lingue, si è scorporato quest'ultimo in un'ulteriore entità che, oltre ad un identificativo univoco, porti anche l'indicazione della lingua della descrizione in questione.

Data la somiglianza tra le varie entità "descrizione", queste sono state organizzate in una gerarchia, di cui comunque l'entità padre (denominata "Descrizione generica") non viene effettivamente tradotta in una tabella dello schema logico derivato dallo schema ER.

Nelle due figure che seguono viene quindi presentata la struttura finale del database: nella prima vi è lo schema ER principale e nella seconda la semplice gerarchia delle descrizioni.

Si fa infine presente che, durante tutto il processo di creazione dello schema, si è cercato di tener conto delle raccomandazioni EUROFOODS, a loro volta basate su varie raccomandazioni INFOODS.


Descrizione nota alimento
Descrizione nota valore
Fonte secondaria
0,n
Origine secondaria
Commento statistico
Descrizione commento statistico
Tipo di valore
Descrizione tipo di valore
Determinazione valore
Descrizione determinazione valore
Componente standard
Nota alimento
Nota valore
0,n
Valore
Descrizione componente standard
Descrizione alimento
Alimento
1,1
Origine alimento
0,n
Fonte primaria
0,n
Origine componente
1,1
Componente
Tipo rifiuti
Peso
Categoria standard
Categoria originaria
1,1
Origine categoria
0,n
Descrizione componente
Descrizione tipo rifiuti
Descrizione misura
Descrizione categoria standard
Descrizione categoria originaria


Descrizione generica
Descrizione alimento
Descrizione nota valore
Descrizione nota alimento
Descrizione tipo di valore
Descrizione componente
Descrizione componente standard
Descrizione commento statistico
Descrizione categoria originaria
Descrizione determinazione valore
Descrizione categoria standard
Descrizione tipo rifiuti
Descrizione misura

5.4 Dizionario dei Dati

Presentiamo ora delle piccole schede delle singole entità. Per ognuna si dà una breve descrizione e, laddove necessario, delle spiegazioni aggiuntive sugli attributi.

5.4.1 Alimento

Alimento			
Codice alimento	<pi>	I	<M>
Marca		VA100	
Nome scientifico		VA150	
N Factor		SF	
Pro Factor		SF	
Fat Factor		SF	
CHO Factor		SF	
Aminoacido limitante		VA15	

Contiene informazioni generali sull'alimento:

- **Marca:** nome del produttore dell'alimento.
- **N Factor:** fattore utilizzato per convertire l'azoto in proteine.
- **Pro Factor:** fattore utilizzato per calcolare le calorie da proteine.
- **Fat Factor:** fattore utilizzato per calcolare le calorie da grassi.
- **CHO Factor:** fattore utilizzato per calcolare le calorie da carboidrati.
- **Aminoacido limitante:** campo presente nella sola fonte INRAN e che non è ben documentato.

5.4.2 Categoria originaria

Categoria originaria			
Codice categoria originaria	<pi>	I	<M>
Categoria standard corrispondente		I	

Contiene informazioni sulla categoria utilizzata nella fonte primaria:

- **Codice categoria originaria:** codice utilizzato in questo database per identificare univocamente la categoria originaria.

- **Categoria standard corrispondente:** eventuale categoria standard in cui la categoria originaria in esame è inclusa.

5.4.3 Categoria standard

Categoria standard			
Codice categoria standard	<pi>	I	<M>

Contiene i codici delle categorie "Eurocode 99/2" (gruppi principali).

Oltre a quelli dello standard (vedi [6]), è stato aggiunto il solo codice “0” per indicare alimenti della base di dati che non sono stati ancora classificati.

5.4.4 Commento statistico

Commento statistico			
Codice commento statistico	<pi>	I	<M>
Codice originario commento statistico		I	

Informazioni sui commenti statistici riguardanti i valori.

Tale entità proviene al momento dalla sola fonte USDA ed è presa dalla documentazione ([3], pag.24).

- **Codice originario commento statistico:** codice che identifica il commento nella fonte primaria.

5.4.5 Componente

Componente			
Codice componente	<pi>	I	<M>
Unità di misura		VA10	<M>
Modo di espressione misura		VA10	<M>
Cifre decimali		SI	
Tagname		VA20	

Informazioni sul componente preso dalla fonte primaria:

- **Unità di misura:** unità con cui è espressa la misura (es. g, mg, etc.).
- **Modo di espressione misura:** quantità di cibo rispetto alla quale si riferisce la misura (es. 100g di parte edibile, 100g di sostanza secca, etc.).
- **Cifre decimali:** numero di cifre decimali alle quali è arrotondato il valore del nutriente. Al momento è disponibile per la sola fonte USDA.
- **Tagname:** tagname utilizzato dall' “International Network of Food Data Systems” (INFOODS): un abbreviazione univoca sviluppata da INFOODS per aiutare nello scambio di dati. Al momento è presente per i soli componenti della fonte USDA ed è riportato così come era nella fonte.

5.4.6 Componente standard

Componente standard			
Sigla componente standard	<pi>	VA12	<M>

Contiene le sigle dei componenti del vocabolario standardizzato "COST Action 99 - EUROFOODS" (vedi [1]).

Sono stati aggiunte delle sigle (e le corrispondenti descrizioni) per i componenti presenti nelle fonti ma non nel vocabolario.

5.4.7 Determinazione valore

Determinazione valore			
Codice determinazione valore	<pi>	I	<M>
Codice originario determinazione valore		VA5	

Informazioni su come il valore è stato ricavato (al momento proviene solo da USDA e corrisponde al suo “Data Derivation Code File”):

- **Codice originario determinazione valore:** codice utilizzato nella fonte primaria (“DERIV_CD” in USDA).

5.4.8 Fonte primaria

Fonte primaria			
Codice fonte	<pi>	I	<M>
Descrizione fonte		VA150	<M>
Ente		VA150	
Sigla ente		VA20	
Indirizzo web		VA100	
Data pubblicazione fonte		VA10	

Informazioni sulle fonti da cui vengono direttamente presi i dati inseriti nel database:

- **Descrizione fonte:** denominazione della fonte. Non è vista come entità a parte perché questa informazione viene riportata come è presentata nella fonte stessa, quindi solo in lingua originale.
- **Ente:** ente fornitore dei dati.
- **Sigla ente:** eventuale acronimo dell'ente.
- **Indirizzo web:** url da cui si possono reperire i dati originali e/o informazioni aggiuntive su di essi.
- **Data pubblicazione fonte:** data della pubblicazione delle fonte da parte dell'organizzazione competente.

5.4.9 Fonte secondaria

Fonte secondaria			
Codice fonte secondaria	<pi>	I	<M>
Codice originario fonte secondaria		VA10	
Sigla originaria fonte secondaria		VA20	
Autori		VA254	
Titolo		VA254	<M>
Editore		VA50	
Anno		SI	
Nome rivista		VA100	
Numero rivista		VA10	
Volume		VA10	
Città		VA20	
Stato		VA20	
Pagina iniziale		SI	
Pagina finale		SI	

Informazioni sulle fonti secondarie, cioè le fonti utilizzate dalle fonti primarie per compilare i rispettivi database:

- **Codice originario fonte secondaria:** codice utilizzato nella fonte primaria per designare univocamente la fonte secondaria.

- **Sigla originaria fonte secondaria:** sigla utilizzata nella fonte primaria per descrivere la fonte secondaria. Il campo è presente solo nei pdf IEO e quindi al momento non è riempito.
- **Autori:** autori del documento o organizzazione sponsorizzante lo stesso.
- **Anno:** anno di pubblicazione del documento.
- **Volume:** volume del libro o dell'articolo di giornale (in inglese: volume).
- **Numero rivista:** numero di rivista in cui compare l'articolo (in inglese: issue).
- **Città:** città dove risiede l'organizzazione sponsorizzante (o l'editore).
- **Stato:** stato dove risiede l'organizzazione sponsorizzante (o l'editore).

5.4.10 Nota alimento

Nota alimento			
Progressivo nota alimento	<pi>	I	<M>
Codice originario nota alimento		VA5	
Nota misura		BL	

Informazioni sulla nota relativa all'alimento.

- **Codice originario nota alimento:** codice (o numero di sequenza) della nota nella fonte primaria.
- **Nota misura:** campo booleano che, se alto, indica che la nota si riferisce ad una misura dell'alimento (si veda l'entità “Peso”). Campo solo USDA.

5.4.11 Nota valore

Nota valore			
Progressivo nota valore	<pi>	I	<M>
Codice originario nota valore		VA5	

Informazioni sulla nota relativa al valore.

- **Codice originario nota valore:** codice (o numero di sequenza) della nota nella fonte primaria.

5.4.12 Peso

Peso			
Progressivo peso	<pi>	NO	<M>
Progressivo peso originario		VA5	
Quantità pesata		SF	<M>
Peso in grammi		SF	<M>
No of data points		I	
Standard Deviation		SF	

Informazioni sul peso in grammi di misure comuni (cioè utilizzate nella vita quotidiana, come ad esempio "una tazza" o "un cucchiaio") dell'alimento. Le misure sono fatte considerando la sola parte edibile del cibo in questione.

Questa entità al momento proviene solo da USDA.

- **Progressivo peso originario:** progressivo (o codice) utilizzato nella fonte primaria per designare univocamente il peso.
- **Quantità pesata:** modificatore della quantità pesata (es. se è “0,5” e la misura è "tazza", allora è stato misurato il peso del contenuto di mezza tazza).
- **Peso in grammi:** media del peso in grammi risultante dalla misura della quantità comune (es. peso del latte presente in una tazza).
- **No of data points:** numero n di osservazioni che hanno contribuito al calcolo della media.
- **Standard deviation:** deviazione standard della media.

5.4.13 Tipo di valore

Tipo di valore			
Codice tipo di valore	<pi>	I	<M>
Codice originario tipo di valore		VA3	

Informazioni sul tipo di valore (es. analitico, calcolato, assunto pari a zero, etc.). Al momento proviene solo da USDA e corrisponde al “Source Code File”.

- **Codice originario tipo di valore:** codice utilizzato nella fonte primaria (“SRC_CD” in USDA).

5.4.14 Tipo rifiuti

Tipo rifiuti			
Codice tipo rifiuti	<pi>	I	<M>

Informazioni sul tipo di scarti (parte non edibile) dell'alimento (es. “ossa”, “semi”, etc.). Al momento questa entità è solo USDA.

5.4.15 Valore

Valore		
Valore	SF	<M>
Tracce	BL	
Standard error	SF	
Min	SF	
Max	SF	
No of data points	I	
Number of studies	I	
Degrees of freedom	SF	
Lower 95% error bound	SF	
Upper 95% error bound	SF	
Alimento di riferimento	I	
Nutriente arricchito	BL	
Età data points(ore)	I	

Informazioni (prevalentemente statistiche) sul valore di un dato componente in un dato alimento:

- **Valore:** quantità media del componente presente nell'alimento.
- **Tracce:** flag che indica, nel caso in cui sia alto, che il componente è presente in tracce (se il flag è alto il campo “Valore” deve essere zero o NULL). (campo solo INRAN e IEO)
- **Standard error:** errore standard della media. (campo solo USDA)
- **Min:** valore minimo osservato. (campo solo USDA e ISA-CNR)
- **Max:** valore massimo osservato. (campo solo USDA e ISA-CNR)
- **No of data points:** numero n di osservazioni del cibo considerato (non necessariamente campioni differenti) utilizzate per calcolare i valori statistici. (campo solo USDA e ISA-CNR)
- **Number of studies:** numero di studi analitici adoperati per calcolare la media. Uno studio è un progetto di analisi dei cibi di una certa rilevanza. (campo solo USDA)

- **Degrees of freedom:** Numero di valori che sono liberi di variare dopo che si sono fissate certe restrizioni sui dati. Utilizzato in calcoli di probabilità. (campo solo USDA)
- **Lower 95% error bound:** valore al di sopra del quale ci si aspetta ricada la media, con livello di confidenza del 95%. (campo solo USDA)
- **Upper 95% error bound:** valore al di sotto del quale ci si aspetta ricada la media, con livello di confidenza del 95%. (campo solo USDA)
- **Alimento di riferimento:** codice dell'alimento che è stato usato per imputare il valore del componente all'alimento considerato. (campo solo USDA)
- **Nutriente arricchito:** flag che, se alto, indica che il nutriente in questione è stato in parte o totalmente aggiunto artificialmente. (campo solo USDA)
- **Età data points (ore):** età (in ore) dei data points considerati. (campo solo ISA-CNR).

5.4.16 Descrizione generica

Descrizione generica		
Lingua	A2	<M>

Descrizione generica: tutte le altre descrizioni derivano gli attributi di questa entità, che comunque non ha un corrispettivo nello schema logico.

- **Lingua:** sigla ISO 639:1998 della lingua in cui è espressa la descrizione.

5.4.17 Descrizione alimento

Descrizione alimento			
Progressivo descrizione alimento	<pi>	NO	<M>
Descrizione alimento		VA250	<M>
Descrizione breve		VA70	
Nomi comuni		VA200	

Contiene le denominazioni dell'alimento:

- **Descrizione breve:** descrizione con al più 70 caratteri; coincide con la descrizione normale se quest'ultima è minore di 70 caratteri (al momento

proviene solo da USDA).

- **Nomi comuni:** altri nomi comunemente usati per descrivere l'alimento.

5.4.18 Altre Descrizioni

Descrizione categoria standard			
Progressivo descrizione categoria standard	<pi>	NO	<M>
Descrizione categoria standard		VA100	<M>

Tutte le altre entità descrittive contengono un campo identificativo (dal nome “Progressivo descrizione NOME_ENTITA’”) e un campo con la descrizione vera e propria (denominato “Descrizione NOME_ENTITA’”).

5.5 Informazioni sulle Singole Associazioni

Le schede mostrate in questo paragrafo riguardano le sole associazioni che presentano degli attributi.

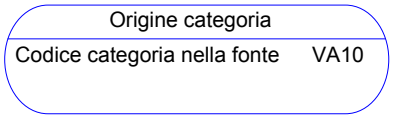
5.5.1 Origine alimento

Origine alimento	
Codice alimento originario	VA10

Lega le entità “Fonte primaria” e “Alimento”:

- **Codice alimento originario:** codice utilizzato nella fonte primaria per identificare univocamente l'alimento.

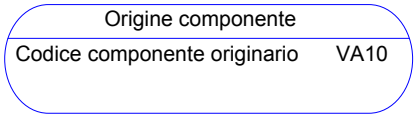
5.5.2 Origine categoria



Lega le entità “Fonte primaria” e “Categoria originaria”:

- **Codice categoria nella fonte:** codice utilizzato nella fonte primaria per identificare univocamente la categoria.

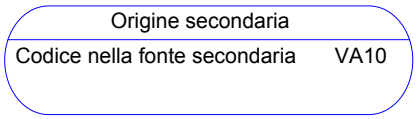
5.5.3 Origine componente



Lega le entità “Fonte primaria” e “Componente”:

- **Codice componente originario:** codice utilizzato nella fonte primaria per determinare univocamente il componente.

5.5.4 Origine secondaria



Lega le entità “Valore” e “Fonte secondaria”:

- **Codice nella fonte secondaria:** codice che identifica univocamente nella fonte secondaria l'alimento utilizzato per determinare il valore in questione.

5.6 Assegnazione del Tipo agli Attributi

Sono stati stabiliti poi i tipi degli attributi, basandosi su quelli già presenti negli schemi da integrare e seguendo le semplici regole illustrate di seguito:

- I campi testuali sono stati tutti tradotti in stringhe a lunghezza variabile con lunghezza massima maggiore o uguale dell'originale. In particolare sono state aumentate sostanzialmente le lunghezze dei campi che erano solo in inglese, in previsione di un aumento di caratteri per eventuali traduzioni.

- Tutti i campi floating point sono stati portati a singola precisione, anche se molti erano originariamente a doppia precisione: questo perché non c'è bisogno dell'enorme quantità di cifre significative fornita dalla precisione doppia.

- I nuovi identificativi sono stati creati come interi lunghi, tranne quelli delle entità "Descrizione", che sono stati creati come numeri progressivi aumentati automaticamente (cioè sono di tipo “serial number”).

5.7 Ulteriori Vincoli

Oltre ai vincoli visibili nello schema ER e nelle schede delle singole entità, sono presenti (e implementati) i seguenti vincoli:

1) L’attributo “Lingua” di “Descrizione generica” può assumere solo un set definito di valori, presi dallo standard ISO 639:1998 "Code for the representation of names of languages". Al momento l’insieme di valori validi comprende “it” e “en”, rispettivamente per italiano e inglese.

2) Il campo “Alimento di riferimento” di "Valore" deve avere, se diverso da NULL, un corrispettivo in "Codice alimento" dell’entità “Alimento”. Questo vincolo non è presente nello schema per non generare confusione.

3) Il campo “Categoria standard corrispondente” dell’entità "Categoria originaria" deve avere, se non NULL, un corrispettivo in “Codice categoria standard” di "Categoria standard".

4) “Unità di misura”, presente nell’entità “Componente”, può assumere solo un set di valori presi dal vocabolario delle raccomandazioni EUROFOODS ([1], pag.56), più eventuali necessarie integrazioni. L’insieme di valori possibili è: “g” (grammi); “mg” (milligrammi); “ug” (microgrammi); “mmol” (millimoli); “R” (rapporto o numero puro); “kcal” (chilocalorie); “kJ” (chiloJoule); “IU” (International Units, aggiunto rispetto allo standard).

5) “Modo di espressione misura” di “Componente” può assumere solo un insieme di valori presi dal vocabolario EUROFOODS ([1], pag.57), più eventuali necessarie integrazioni. I valori utilizzati al momento sono: “W” (che significa “per 100g di parte edibile”); "T" (=”per 100g di cibo totale”); "P" (“per 100g di proteine”, aggiunto rispetto allo standard); "D" (=”per 100g di sostanza secca”); "G" (“per 100g di sostanza grassa”, aggiunto allo standard).

6) Tutte le entità sono legate alla rispettiva entità “Descrizione” con una associazione di cardinalità (1,n): n in questo caso è fissato ed è il numero di lingue supportate nella base di dati (inizialmente due: italiano e inglese).

7) Nell’entità “Valore”, se il campo booleano “Tracce” è vero, allora l’attributo “Valore” deve essere zero o NULL.

5.8 Diagrammi Completi

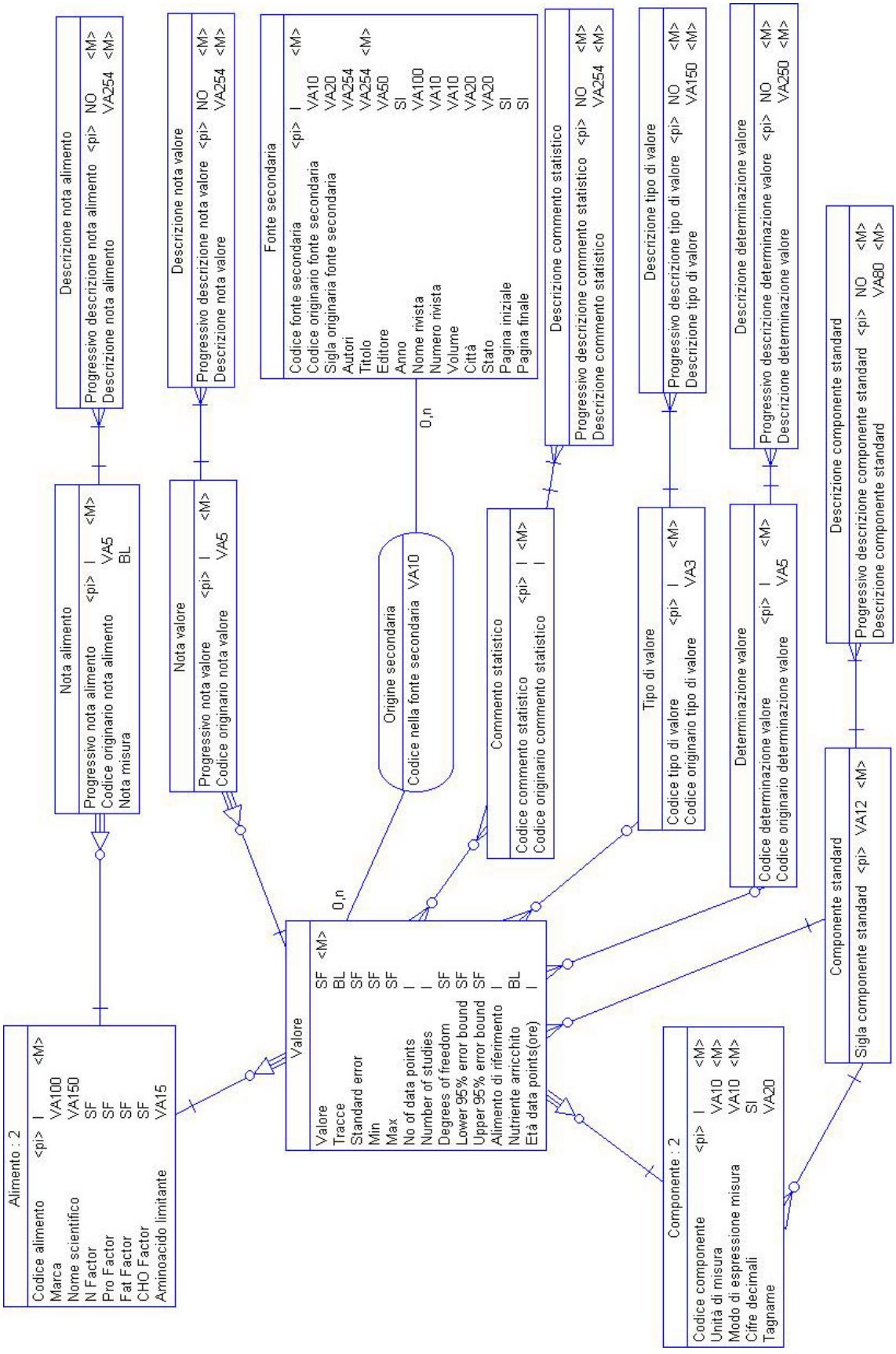
Vengono ora presentati, a scopo di riepilogo e per mostrare come tutti i pezzi si incastrano insieme, lo schema ER completo (dal nome “Final Conceptual”) e il corrispettivo schema logico. Quest’ultimo è stato ottenuto dalla traduzione automatica effettuata dal CASE, specificando che il DBMS utilizzato è PostgreSQL.

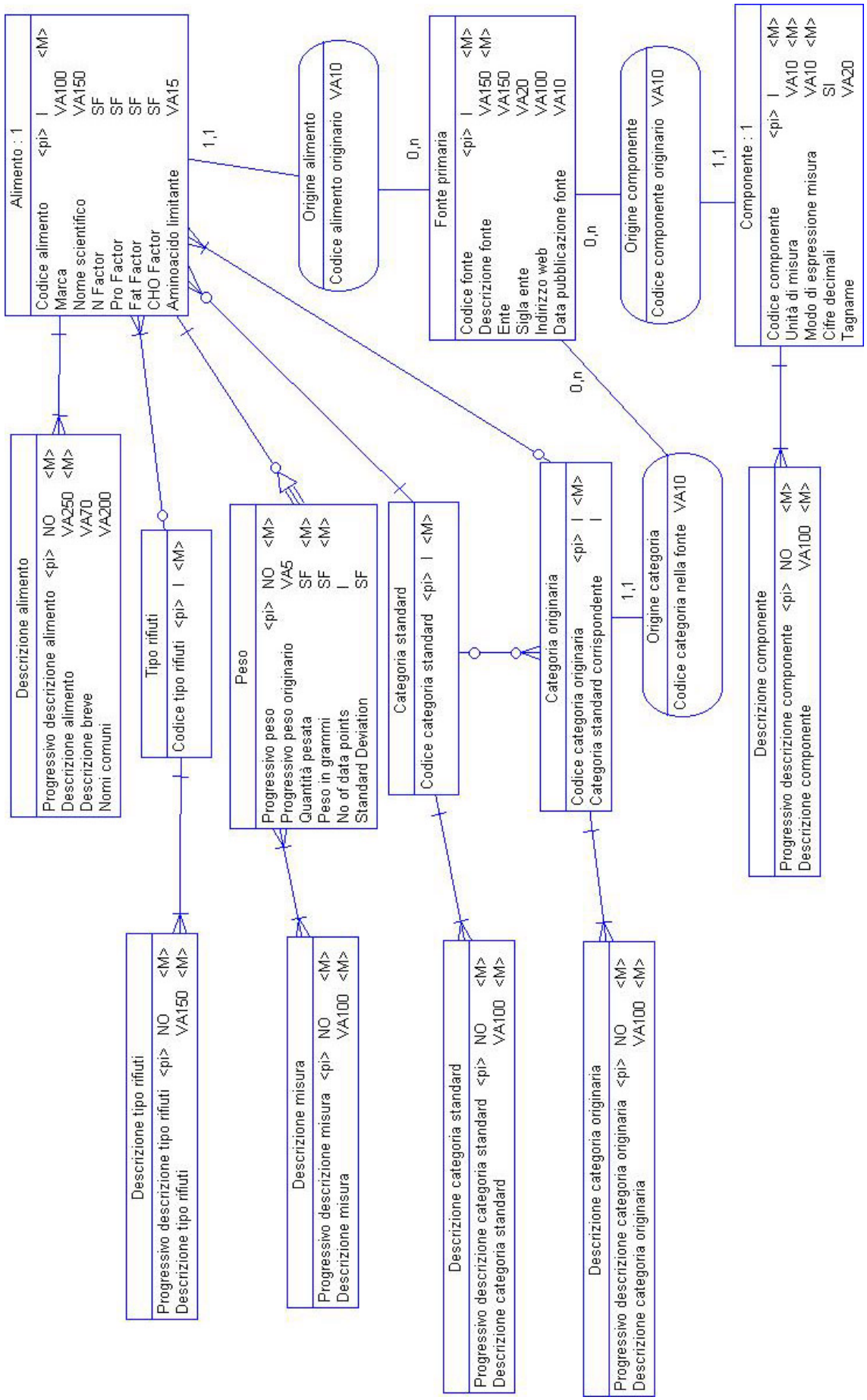
Entrambi i diagrammi occupano due pagine ciascuno e quindi si è fatto utilizzo di sinonimi grafici: la stessa entità compare due volte per mostrare i collegamenti tra le due pagine.

A completamento dello schema ER bisogna ovviamente anche consultare il diagramma con la gerarchia delle descrizioni già presentato in precedenza.

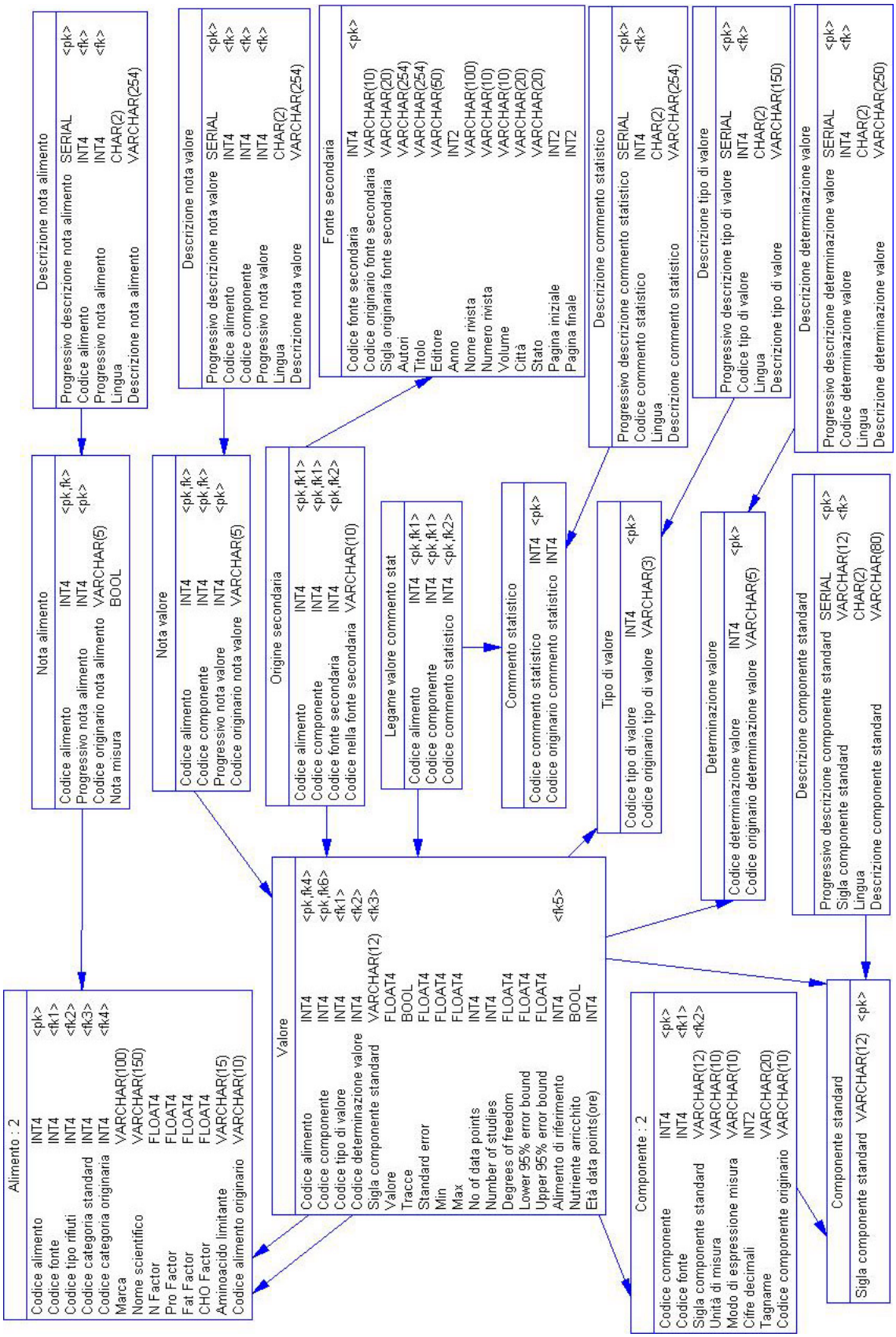
Da notare anche che nello schema logico (che nel CASE è stato denominato “Final Physical”) i tipi non sono quelli dello standard SQL, ma vengono usati degli alias tipici di PostgreSQL (es. “INT4” al posto di “INTEGER”). Il DBMS in questione infatti può utilizzare entrambe le notazioni e il CASE preferisce, per motivi di compatibilità con le versioni precedenti, adoperare la notazione non standard: si veda [13] per ulteriori spiegazioni sui tipi impiegati.

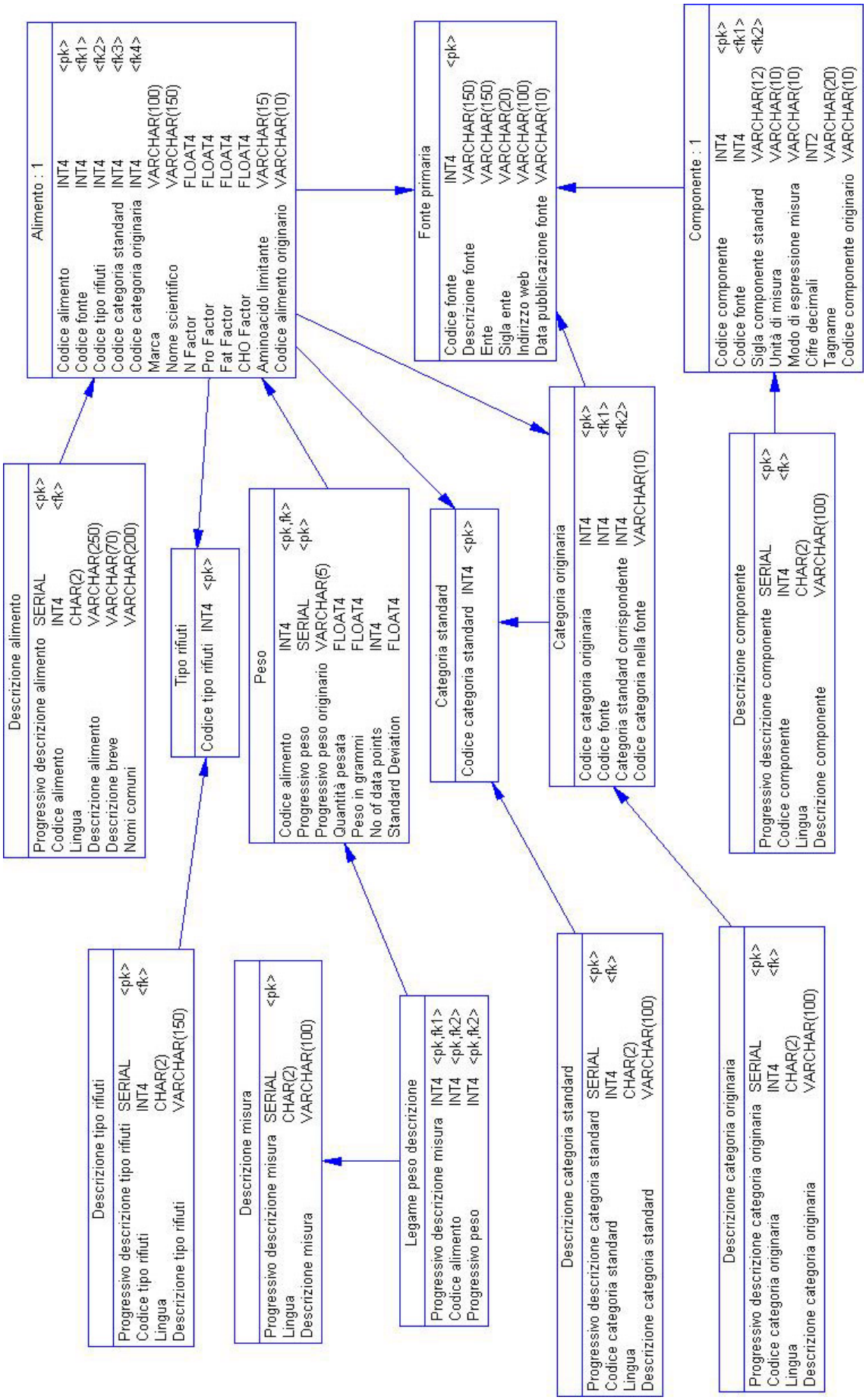
5.8.1 Schema “Final Conceptual”




Alimento : 1
Codice alimento <pi> I <M>
Marca VA100
Nome scientifico VA150
N Factor SF
Pro Factor SF
Fat Factor SF
CHO Factor SF
Aminoacido limitante VA15
1,1
Origine alimento
Codice alimento originario VA10
0,n
Fonte primaria
Codice fonte <pi> I <M>
Descrizione fonte VA150 <M>
Ente VA150
Sigla ente VA20
Indirizzo web VA100
Data pubblicazione fonte VA10
0,n
Origine componente
Codice componente originario VA10
1,1
Componente : 1
Codice componente <pi> I <M>
Unità di misura VA10 <M>
Modo di espressione misura VA10 <M>
Cifre decimali SI
Tagname VA20
Descrizione alimento
Progressivo descrizione alimento <pi> NO <M>
Descrizione alimento VA250 <M>
Descrizione breve VA70
Nomi comuni VA200
Tipo rifiuti
Codice tipo rifiuti <pi> I <M>
Peso
Progressivo peso <pi> NO <M>
Progressivo peso originario VA5
Quantità pesata SF <M>
Peso in grammi SF <M>
No of data points I
Standard Deviation SF
Categoria standard
Codice categoria standard <pi> I <M>
Categoria originaria
Codice categoria originaria <pi> I <M>
Categoria standard corrispondente I
1,1
Origine categoria
Codice categoria nella fonte VA10
Descrizione componente
Progressivo descrizione componente <pi> NO <M>
Descrizione componente VA100 <M>
Descrizione tipo rifiuti
Progressivo descrizione tipo rifiuti <pi> NO <M>
Descrizione tipo rifiuti VA150 <M>
Descrizione misura
Progressivo descrizione misura <pi> NO <M>
Descrizione misura VA100 <M>
Descrizione categoria standard
Progressivo descrizione categoria standard <pi> NO <M>
Descrizione categoria standard VA100 <M>
Descrizione categoria originaria
Progressivo descrizione categoria originaria <pi> NO <M>
Descrizione categoria originaria VA100 <M>

5.8.2 Schema “Final Physical”




Alimento : 1
Codice alimento INT4 <pk>
Codice fonte INT4 <fk1>
Codice tipo rifiuti INT4 <fk2>
Codice categoria standard INT4 <fk3>
Codice categoria originaria INT4 <fk4>
Marca VARCHAR(100)
Nome scientifico VARCHAR(150)
N Factor FLOAT4
Pro Factor FLOAT4
Fat Factor FLOAT4
CHO Factor FLOAT4
Aminoacido limitante VARCHAR(15)
Codice alimento originario VARCHAR(10)
Fonte primaria
Codice fonte INT4 <pk>
Descrizione fonte VARCHAR(150)
Ente VARCHAR(150)
Sigla ente VARCHAR(20)
Indirizzo web VARCHAR(100)
Data pubblicazione fonte VARCHAR(10)
Componente : 1
Codice componente INT4 <pk>
Codice fonte INT4 <fk1>
Sigla componente standard VARCHAR(12) <fk2>
Unità di misura VARCHAR(10)
Modo di espressione misura VARCHAR(10)
Cifre decimali INT2
Tagname VARCHAR(20)
Codice componente originario VARCHAR(10)
Descrizione alimento
Progressivo descrizione alimento SERIAL <pk>
Codice alimento INT4 <fk>
Lingua CHAR(2)
Descrizione alimento VARCHAR(250)
Descrizione breve VARCHAR(70)
Nomi comuni VARCHAR(200)
Tipo rifiuti
Codice tipo rifiuti INT4 <pk>
Peso
Codice alimento INT4 <pk,fk>
Progressivo peso SERIAL <pk>
Progressivo peso originario VARCHAR(5)
Quantità pesata FLOAT4
Peso in grammi FLOAT4
No of data points INT4
Standard Deviation FLOAT4
Categoria standard
Codice categoria standard INT4 <pk>
Categoria originaria
Codice categoria originaria INT4 <pk>
Codice fonte INT4 <fk1>
Categoria standard corrispondente INT4 <fk2>
Codice categoria nella fonte VARCHAR(10)
Descrizione componente
Progressivo descrizione componente SERIAL <pk>
Codice componente INT4 <fk>
Lingua CHAR(2)
Descrizione componente VARCHAR(100)
Descrizione tipo rifiuti
Progressivo descrizione tipo rifiuti SERIAL <pk>
Codice tipo rifiuti INT4 <fk>
Lingua CHAR(2)
Descrizione tipo rifiuti VARCHAR(150)
Descrizione misura
Progressivo descrizione misura SERIAL <pk>
Lingua CHAR(2)
Descrizione misura VARCHAR(100)
Legame peso descrizione
Progressivo descrizione misura INT4 <pk,fk1>
Codice alimento INT4 <pk,fk2>
Progressivo peso INT4 <pk,fk2>
Descrizione categoria standard
Progressivo descrizione categoria standard SERIAL <pk>
Codice categoria standard INT4 <fk>
Lingua CHAR(2)
Descrizione categoria standard VARCHAR(100)
Descrizione categoria originaria
Progressivo descrizione categoria originaria SERIAL <pk>
Codice categoria originaria INT4 <fk>
Lingua CHAR(2)
Descrizione categoria originaria VARCHAR(100)

CAPITOLO SEI

I MODULI DI FEEDING

Si descrive ora il software che alimenta il database di integrazione oggetto del capitolo precedente.

6.1 Nota Iniziale

Prima di iniziare la disamina del progetto, si ricorda che vengono illustrati gli aspetti essenziali alla comprensione dello stesso, senza approfondire le tecnologie utilizzate e senza documentare in maniera pedante tutto. Per maggiori delucidazioni si veda la bibliografia fornita e il materiale allegato alla tesi.

Inoltre, qualora fosse necessario, ogni volta che viene nominata per la prima volta una tecnologia freeware verrà fornito anche il sito dalla quale questa è reperibile.

Si fa anche presente che nel seguito ci si riferirà a quello che finora è stato chiamato “database di integrazione” anche col termine datawarehouse. Questo perché i dati presenti in entrambi sono gli stessi e quindi, in senso lato, il modulo di feeding alimenta anche il magazzino di dati ricavato dal database di integrazione.

6.2 I Requisiti Software

Il modulo software, denominato semplicemente “Feeder”, è progettato per assolvere alle operazioni di gestione della datawarehouse che siano automatizzabili e batch, cioè non richiedano la presenza di un operatore durante il loro completamento.

L’utente dell’applicativo è il Data Base Administrator (DBA), quindi ci si rivolge ad un’utenza esperta, cui siano ben chiare le conseguenze di certe azioni critiche.

Le operazioni implementate dal modulo sono fondamentalmente tre:

1) **Inserimento** dei dati da una fonte al database.

2) **Aggiornamento** dei dati da una fonte al database.

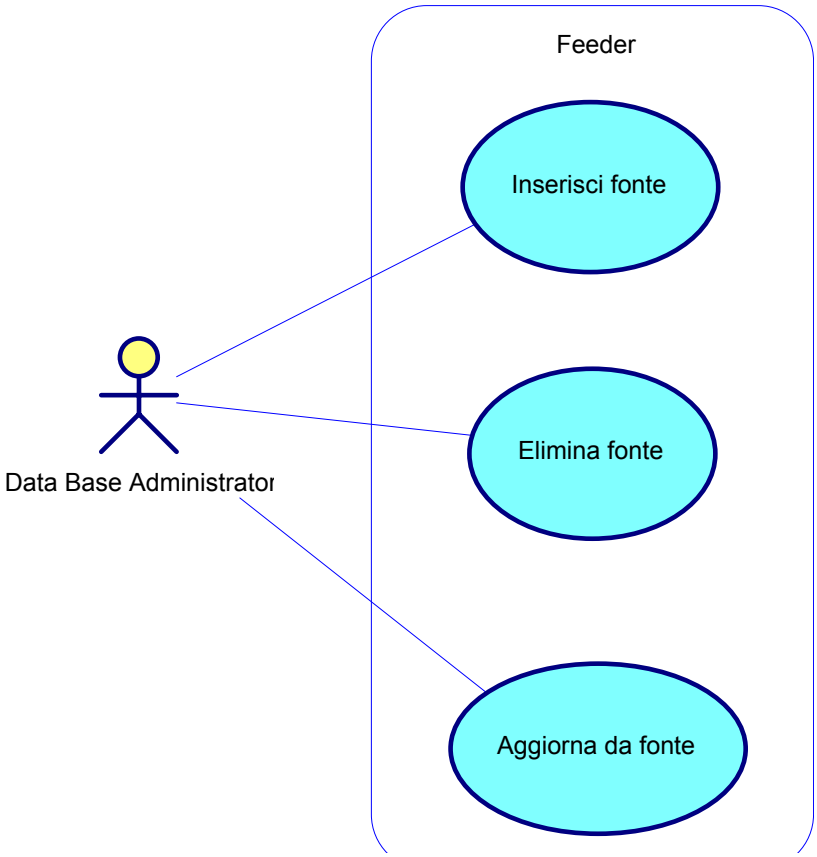
3) **Eliminazione** dei dati relativi ad una singola fonte dal database.

Le operazioni suddette vanno svolte in maniera atomica, nel senso che, ad esempio, anche un errore su una singola tupla deve far annullare l’intero inserimento di una fonte. Questo per non avere incongruenze all’interno della datawarehouse: è facile immaginare le conseguenze di una diversa scelta progettuale.

Inoltre, trattandosi spesso di grossi lavori batch da svolgere una tantum, il tempo di esecuzione può oscillare tra qualche minuto a poche ore. Ovviamente tale tempo sarà sempre proporzionale alle dimensioni dei dati in gioco (es. un'operazione di inserimento di pochi megabyte di dati non può impiegare delle ore!).

Altro importante requisito è che il software permetta una facile introduzione di ulteriori sorgenti di dati: deve essere possibile riusare il codice e scrivere per ogni fonte solo le linee necessarie all'estrazione dei dati dalla specifica sorgente.

Per meglio illustrare quanto detto finora, presentiamo un semplice use case diagram in UML (nel seguito saranno spesso mostrati diagrammi in tale linguaggio di modellazione visuale, per un approfondimento del quale si consiglia di consultare [15] e [16]).



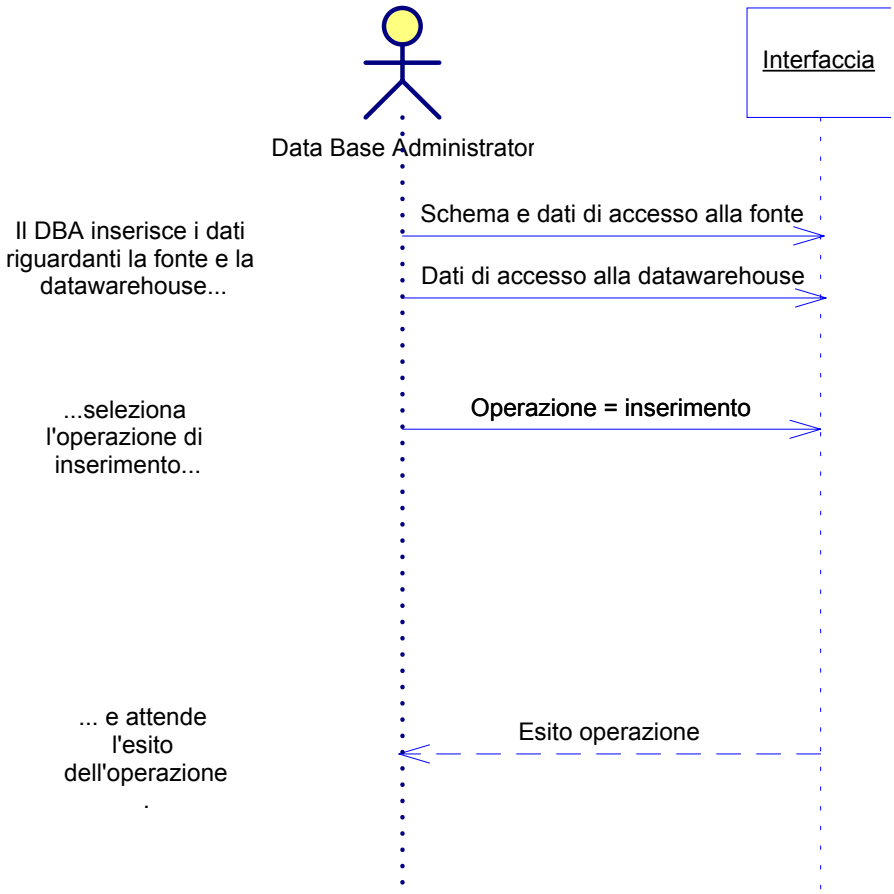
Presentiamo quindi una semplice descrizione dei singoli casi d'uso.

6.2.1 Inserisci fonte

Tramite questa funzionalità l'utente effettua un primo inserimento dei dati relativi ad una fonte. Le informazioni non introducibili con un inserimento automatico, ma indispensabili a quest'ultimo, devono essere ovviamente immesse prima dal DBA, ad esempio tramite l'utilizzo di script (così che la procedura risulti comunque ripetibile).

Il sequence diagram mostrato di seguito illustra un esempio dell'andamento, che si ripete essere atomico, dell'operazione.

L'oggetto "Interfaccia" indica una generica interfaccia utente: il software progettato infatti è indipendente da quest'ultima, che può essere così modificata a piacimento in un secondo momento.



Si noti che tra i dati inseriti dall'utente è presente lo schema della fonte: il DBA deve cioè specificare se la fonte è INRAN, USDA, etc.

Naturalmente quello mostrato è solo un possibile andamento (uno scenario) di questo caso d'uso: scenari alternativi prevedono ad esempio errori nella connessione alla fonte o alla datawarehouse.

6.2.2 Elimina fonte

Questa operazione permette all'utente di eliminare tutti i dati facenti riferimento ad una singola fonte, quindi anche quelli introdotti dopo il primo inserimento automatico.

Si capisce dunque come questa procedura, come spesso accade per le eliminazioni, sia estremamente rischiosa e vada adoperata raramente, soprattutto vista la natura poco dinamica del database di integrazione.

L'andamento tipico di questa operazione è simile a quello visto per l'inserimento, con la sola differenza che ora l'utente non deve inserire i dati di accesso alla fonte, in quanto l'operazione si svolge esclusivamente sulla datawarehouse.

6.2.3 Aggiorna da fonte

Dopo il primo inserimento, possono avvenire degli aggiornamenti successivi qualora la sorgente di dati fornisca i file "differenziali": tra le fonti selezionate solo USDA garantisce questo tipo di servizio.

Anche questa operazione ha un andamento simile a quello dell'inserimento: si veda quindi il paragrafo relativo a tale caso d'uso.

6.3 La Struttura delle Classi

6.3.1 Un Ulteriore Sguardo alle Tecnologie

Come già preannunciato nel primo capitolo, si è deciso di progettare e implementare i moduli di feeding utilizzando metodologie di progettazione e di programmazione ad oggetti.

Per l'implementazione si è scelto il linguaggio C++, corredato delle API ODBC: questa combinazione è facilmente portabile in ambiente Windows ed inoltre garantisce efficienza.

Il programma si interfaccia alla datawarehouse tramite ODBC, cosa che garantisce indipendenza dal DBMS: per cambiare quest'ultimo, infatti, l'utente deve semplicemente creare uno script di generazione del database (con PowerDesigner) per il nuovo DBMS e dare al programma un Data Source Name (DSN) diverso.

Esula dagli scopi di questo documento illustrare le API ODBC: all'uopo si veda l'ampia letteratura in materia oppure il sito della Microsoft (alla pagina: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odintropr.asp).

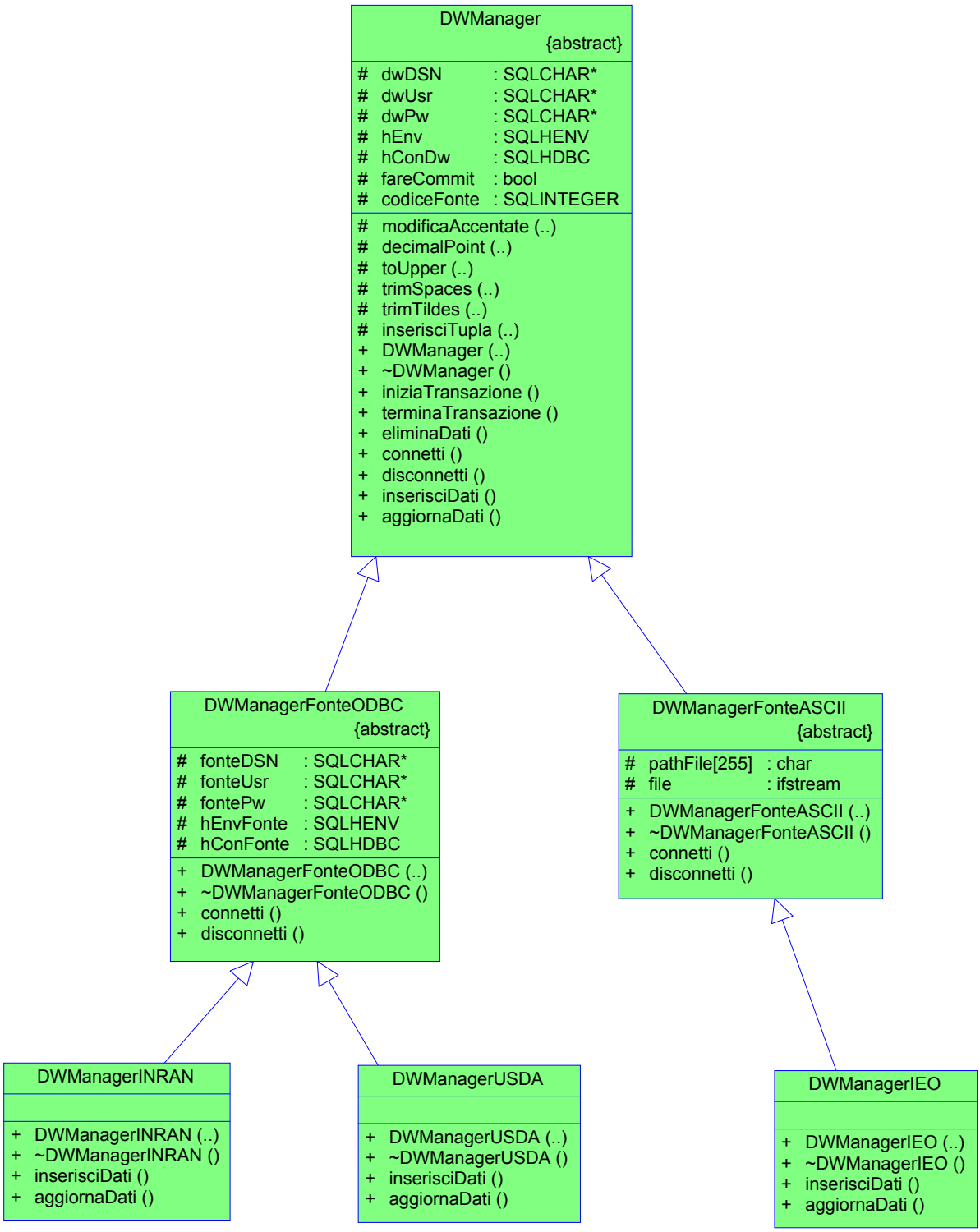
La particolare implementazione di ODBC impiegata è unixODBC (http://www.unixodbc.org/), arricchita con i driver per Access forniti da MDB Tools (http://mdbtools.sourceforge.net/).

Si fa presente sin da ora che la struttura a oggetti non è stata utilizzata per creare un incapsulamento delle API ODBC (un "wrapper", come ad esempio quello fornito dalle MFC di Microsoft): il programma è scritto, per motivi di semplicità e di efficienza, in C, utilizzando le estensioni ad oggetti solo per le funzionalità proprie dell'applicazione.

6.3.2 La Gerarchia Principale

Data la natura relativamente semplice del problema, si è deciso di adottare una gerarchia unica, senza particolari collaborazioni, che viene mostrata nel class diagram seguente.

Nella figura, per motivi di chiarezza, non vengono mostrati né i parametri né il valore ritornato dai singoli metodi, che saranno comunque illustrati in seguito.



Scopo primario della gerarchia è ovviamente la gestione della datawarehouse, cioè l'implementazione delle regole alla base dei casi d'uso. Per fare ciò queste classi devono ovviamente collaborare con un interfaccia utente, che ha il semplice compito di richiedere e fornire i dati al DBA.

Viste nel loro insieme, le classi della figura precedente possono essere concettualmente raggruppate in un **package** denominato **"Data Warehouse Management"**.

6.3.3 La Gerarchia dei Buffer

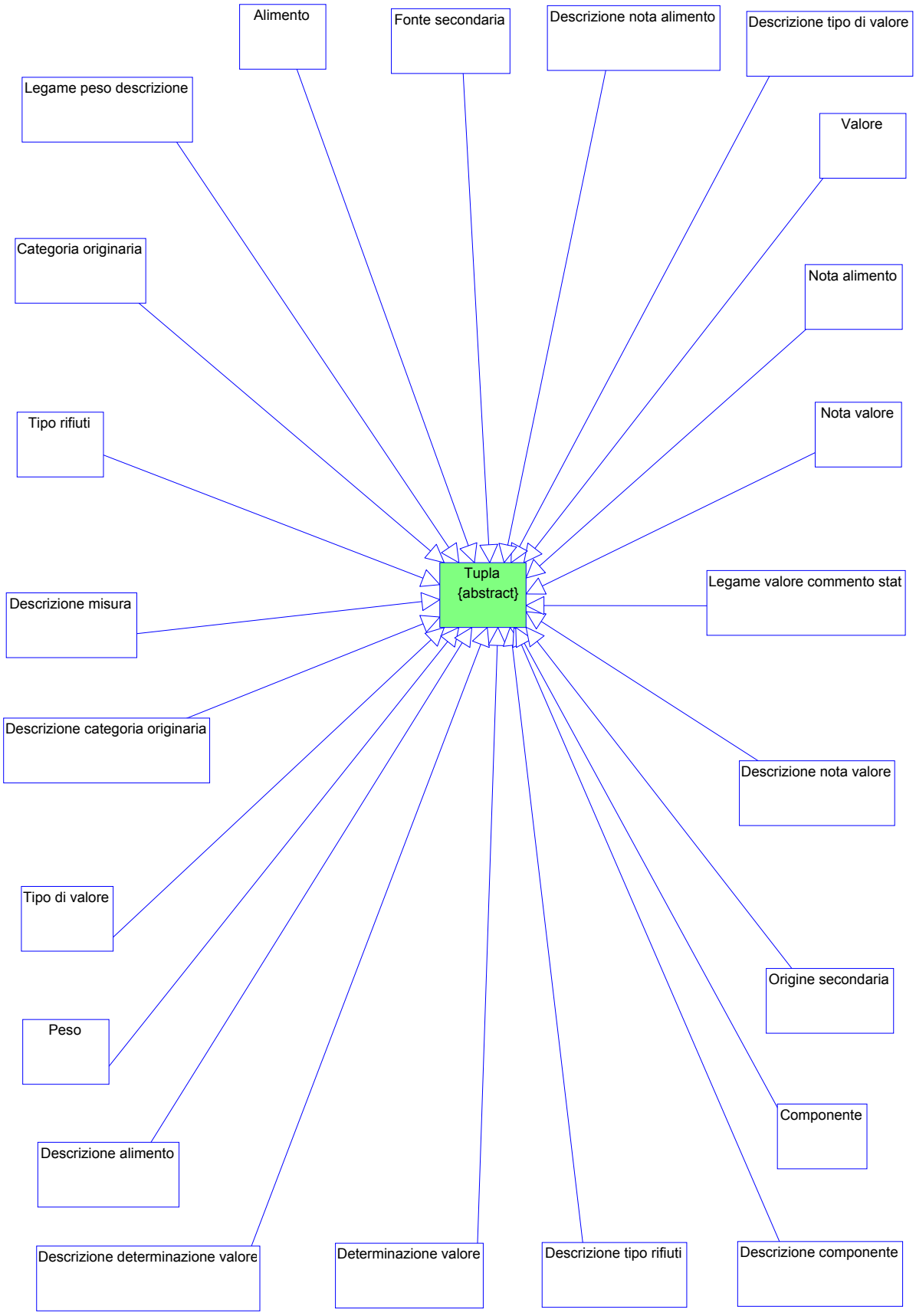
Oltre all'interfaccia utente, l'unico altro elemento dell'applicazione è un ulteriore package, contenente delle classi con semplice funzione di storage delle tuple (sono cioè dei veri e propri buffer).

La relazione di dipendenza tra i due package è mostrata nel diagramma seguente:

La dipendenza è dovuta essenzialmente al fatto che le classi nel primo package istanziano e utilizzano i buffer presenti nel secondo.

Il contenuto del **package "Tuple Buffer"** è illustrato nel class diagram seguente, dove si può notare una classe base astratta ("Tupla") e le classi derivate atte ad ospitare una tupla delle varie tabelle del database di integrazione (es. "Alimento").

Come sempre, in figura mancano molti particolari, omessi per non rendere confuso il diagramma: per ora si vuole semplicemente mostrare il modo in cui i vari elementi si relazionano tra di loro.


Alimento
Fonte secondaria
Descrizione nota alimento
Descrizione tipo di valore
Legame peso descrizione
Valore
Categoria originaria
Nota alimento
Tipo rifiuti
Nota valore
Tupla
{abstract}
Descrizione misura
Legame valore commento stat
Descrizione categoria originaria
Descrizione nota valore
Tipo di valore
Origine secondaria
Peso
Componente
Descrizione alimento
Descrizione determinazione valore
Determinazione valore
Descrizione tipo rifiuti
Descrizione componente

6.4 Informazioni sulle Singole Classi

Presentiamo ora delle schede sulle varie classi, illustrando ovviamente dove necessario anche le relazioni che le legano con le altre. La trattazione parte ovviamente dal package “Data Warehouse Management”, che costituisce l’essenza dell’applicazione.

6.4.1 DWManager

DWManager {abstract}	
# dwDSN : SQLCHAR*	
# dwUsr : SQLCHAR*	
# dwPw : SQLCHAR*	
# hEnv : SQLHENV	
# hConDw : SQLHDBC	
# fareCommit : bool	
# codiceFonte : SQLINTEGER	
# modificaAccentate (char* stringa)	: void
# decimalPoint (char* stringa)	: void
# toUpper (char* stringa)	: void
# trimSpaces (char* stringa)	: void
# trimTildes (char* stringa)	: void
# inserisciTupla (Tupla* tupla)	: SQLRETURN
+ DWManager (SQLCHAR* DSNdw, SQLCHAR* Utentedw, SQLCHAR* Pwdw)	
+ ~DWManager ()	
+ iniziaTransazione ()	: SQLRETURN
+ terminaTransazione ()	: SQLRETURN
+ eliminaDati ()	: SQLRETURN
+ connetti ()	: SQLRETURN
+ disconnetti ()	: SQLRETURN
+ inserisciDati ()	: SQLRETURN
+ aggiornaDati ()	: SQLRETURN

Questa classe ha la fondamentale responsabilità di fornire alle classi da essa derivate tutti i metodi per operare sulla datawarehouse (tramite ODBC). Tale classe cioè ha piena conoscenza della struttura della datawarehouse, ma ignora completamente invece lo schema delle fonti.

Questa classe inoltre impone, attraverso la definizione di alcuni metodi virtuali puri, la semplice interfaccia cui tutte le classi derivate devono conformarsi: si tratta dunque di una classe astratta.

Attributi:

- **dwDSN, dwUsr, dwPw:** immagazzinano rispettivamente DSN, user name e password della sorgente di dati ODBC corrispondente alla datawarehouse.

- **hEnv:** handle dell'environment ODBC all'interno del quale vengono definite le connessioni alla datawarehouse.

- **hConDw:** handle della connessione ODBC alla datawarehouse.

- **fareCommit:** campo booleano che indica lo stato corrente della transazione: se è vero la transazione va terminata con commit, se è falso con rollback.

- **codiceFonte:** codice della fonte nella datawarehouse.

Metodi:

- **DWManager(SQLCHAR* DSNdw, SQLCHAR* Utentedw, SQLCHAR* Pwdw):** il costruttore inizializza gli attributi dwDSN, dwUsr e dwPw con i parametri forniti dall'utente. Inoltre in tale metodo fareCommit viene settato a falso.

- **connetti():** istanzia gli handle hEnv e hConDw e connette alla datawarehouse.

- **disconnetti():** disconnette dalla datawarehouse e dealloca gli handle dell'ambiente e della connessione.

- **iniziaTransazione():** comincia una transazione sulla connessione alla datawarehouse.

- **terminaTransazione():** conclude la transazione sulla datawarehouse, facendo commit o rollback a seconda del flag fareCommit.

- **inserisciDati():** funzione virtuale pura il cui corpo deve essere definito nelle classi derivate: questo metodo ha infatti il compito di inserire i dati relativi alla fonte nella datawarehouse, e quindi richiede la conoscenza dello schema della specifica fonte. Per raggiungere lo scopo vengono utilizzati i buffer del package “Tuple Buffer” e il metodo protetto inserisciTupla().

- **aggiornaDati():** funzione virtuale pura il cui corpo deve essere definito nelle classi derivate: questo metodo ha infatti il compito di aggiornare i dati relativi alla fonte nella datawarehouse, e quindi richiede la conoscenza dello schema della specifica fonte. Per raggiungere lo scopo vengono utilizzati i buffer del package “Tuple Buffer” e il metodo protetto inserisciTupla().

- **eliminaDati():** questo metodo elimina tutti i dati relativi alla fonte. E’ implementato in DWManager perché effettua operazioni esclusivamente sulla datawarehouse.

- **inserisciTupla():** metodo che accetta come parametro un puntatore alla classe astratta Tupla: all’interno, tramite l’utilizzo del supporto Run Time Type Identification (RTTI) del C++, si stabilisce quale è il reale aspetto del buffer e si inserisce la tupla nella corrispettiva tabella. Alla prima invocazione prepara le query, che verranno poi semplicemente eseguite con i parametri inviati nel buffer di volta in volta.

I rimanenti metodi lavorano tutti sulla stringa passata loro come parametro e compiono operazioni di formattazione del testo sui buffer prima che questi vengano inseriti nella datawarehouse:

- **toUpper(char* stringa):** porta tutte le lettere della stringa da minuscole a maiuscole. Nella datawarehouse infatti, per velocizzare le ricerche, tutti caratteri sono memorizzati come maiuscoli.

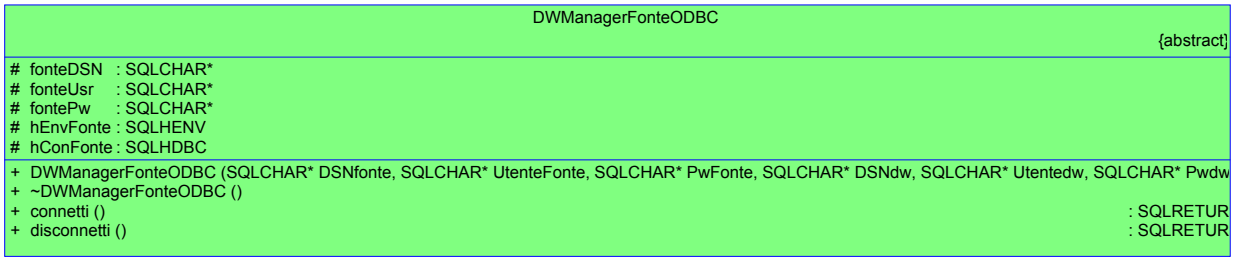
- **modificaAccentate(char* stringa):** sostituisce tutte le occorrenze di lettere accentate (es. “ è ”) con la corrispettiva lettera minuscola più l’apostrofo (es. “ e’ ”). Se si vogliono caratteri tutti maiuscoli, va utilizzata prima di toUpper().

- **decimalPoint(char* stringa):** sostituisce tutte le virgole in una stringa con dei punti. E' adoperata qualora la fonte fornisca valori numerici in campi di testo e usi la virgola come separatore delle cifre decimali: molti dei driver ODBC adoperati, infatti, richiedono che si usi il punto.

- **trimSpaces(char* stringa):** elimina gli spazi presenti all'inizio e alla fine di una stringa.

- **trimTildes(char* stringa):** elimina i tutti i caratteri tilde (cioè "~") presenti all'inizio e alla fine di una stringa. Viene al momento impiegata nel processamento dei soli file di aggiornamento USDA, in cui i campi di testo iniziano e terminano appunto con delle tilde.

6.4.2 DWManagerFonteODBC



Questa classe fornisce i metodi per effettuare la connessione alle fonti utilizzando ODBC. Per far ciò vengono semplicemente aggiunti degli attributi e ridefiniti dei metodi della classe base.

Poiché tale classe eredita ma non implementa i metodi virtuali puri del padre, anche essa è astratta.

Attributi:

- **fonteDSN, fonteUsr, fontePw:** immagazzinano rispettivamente DSN, user name e password della sorgente di dati ODBC cui corrisponde la fonte.

- **hEnvFonte:** handle dell'environment ODBC all'interno del quale vengono definite le connessioni alla fonte.

- **hConFonte:** handle della connessione ODBC alla fonte.

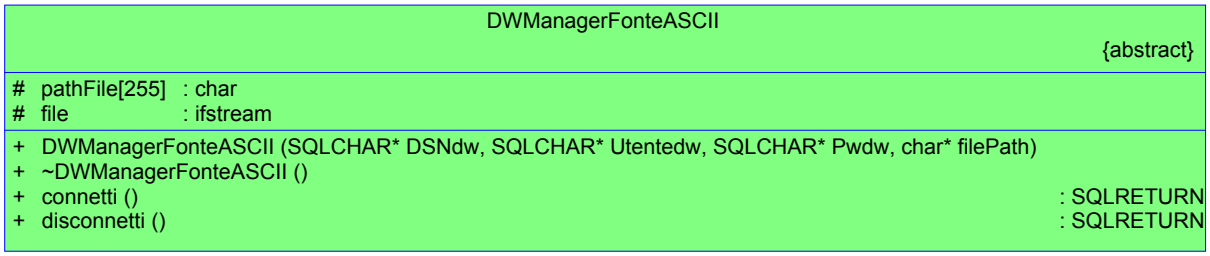
Metodi:

- **DWManagerFonteODBC(SQLCHAR* DSNfonte, SQLCHAR* UtenteFonte, SQLCHAR* PwFonte, SQLCHAR* DSNdw, SQLCHAR* Utentedw, SQLCHAR* Pwdw):** questo costruttore assegna il valore iniziale agli attributi fonteDSN, fonteUsr e fontePw adoperando i corrispettivi parametri forniti dall'utente. Gli altri parametri vengono invece passati al costruttore della classe base.

- **connetti():** ridefinizione di connetti() della classe padre. Invoca il metodo corrispondente della classe base (per connettersi alla datawarehouse). Dopo, se il DSN della fonte è diverso dalla stringa vuota, istanzia gli handle hEnvFonte e hConFonte e si connette alla fonte.

- **disconnetti():** ridefinizione di disconnetti() della classe padre. Invoca il metodo corrispondente della classe base (per disconnettersi dalla datawarehouse). Dopo si disconnette dalla fonte e dealloca gli handle hEnvFonte e hConFonte.

6.4.3 DWManagerFonteASCII



Questa classe fornisce i metodi per effettuare la connessione alle fonti i cui dati risiedono in semplici file di testo ASCII. Per far ciò vengono aggiunti degli attributi e ridefiniti dei metodi della classe base.

Per usufruire di file ASCII come fonti si può adoperare anche la classe DWManagerFonteODBC, in quanto è possibile connettersi ad un file ASCII con campi delimitati da un separatore utilizzando per l'appunto ODBC. Per file con campi non delimitati si hanno due possibilità: o si trasforma il file, inserendo dei separatori, o si utilizza questa classe.

Poiché tale classe eredita ma non implementa i metodi virtuali puri del padre, anche essa è astratta.

Attributi:

- **pathFile:** stringa contenente il percorso che denota la posizione del file ASCII, compreso il nome del file stesso. Questo path non può essere più lungo di 254 caratteri e non deve includere spazi.

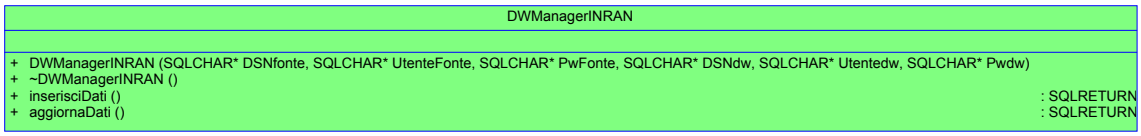
- **file:** oggetto di tipo ifstream attraverso il quale viene gestito lo stream di input dal file.

Metodi:

- **connetti():** ridefinizione di connetti() della classe padre. Invoca il metodo corrispondente della classe base (per connettersi alla datawarehouse). Dopo, se la path del file è diversa dalla stringa vuota, lo apre e lo lega all'attributo file.

- **disconnetti():** ridefinizione di disconnetti() della classe padre. Invoca il metodo corrispondente della classe base (per disconnettersi dalla datawarehouse). Dopo chiude il file.

6.4.4 DWManagerINRAN, DWManagerUSDA, DWManagerIEO



Queste classi, che rappresentano le foglie della gerarchia, sono a conoscenza sia dello schema della fonte sia di quello della datawarehouse. Grazie a ciò esse possono implementare la semplice interfaccia definita in DWManager e assolvere quindi allo scopo primario della gerarchia: alimentare il database.

Va comunque detto che le implementazioni di aggiornaDati() di DWManagerINRAN e DWManagerIEO sono vuote. Questo è dovuto al fatto che tali fonti non solo non forniscono periodicamente dei file di aggiornamento, ma inoltre non sembrano garantire la persistenza dello schema tra due aggiornamenti: tutto ciò rende ovviamente un inutile esercizio di programmazione l'implementare i rispettivi metodi aggiornaDati().

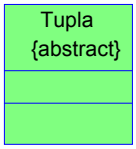
Metodi:

- **DWManagerINRAN(SQLCHAR* DSNfonte, SQLCHAR* UtenteFonte, SQLCHAR* PwFonte, SQLCHAR* DSNdw, SQLCHAR* Utentedw, SQLCHAR* Pwdw):** il costruttore passa i parametri al costruttore della classe base e scrive in codiceFonte il codice che identifica la specifica fonte nella datawarehouse.

- **inserisciDati():** questa funzione preleva i dati dalla fonte, copiandoli nei buffer del package "Tuple Buffer", e li inserisce nella datawarehouse tramite inserisciTupla(). Tale metodo viene utilizzata per un primo inserimento dei dati da una fonte: per ulteriori introduzioni si veda aggiornaDati().

- **aggiornaDati():** questo metodo inserisce i nuovi dati come accade per inserisciDati() e effettua anche le eventuali cancellazioni e aggiornamenti dei dati preesistenti.

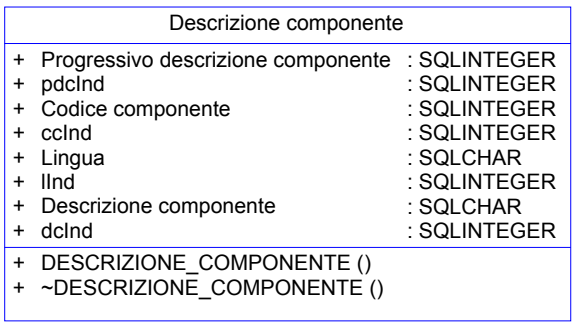
6.4.5 Tupla



Questa banale classe astratta serve sia a realizzare il vincolo della realtà che indica che tutti i buffer sono accomunati dal contenere una tupla, sia a fornire un ulteriore tipo alle classi dei buffer. Infatti il tipo “Tupla” viene usato nei metodi che accolgono come parametro il generico buffer per poi identificarne a run time l’effettivo tipo specifico (es. “Alimento”, etc.).

L’utilizzo di tale classe è dettato da questioni di stile più che di praticità: si è semplicemente deciso di utilizzare Tupla* piuttosto di un generico void*.

6.4.6 Le Classi Buffer



Queste classi ospitano ognuna una tupla di una tabella diversa della datawarehouse. I loro campi corrispondono alle colonne del database ed infatti sono queste classi sono state ricavate dalla traduzione automatica, effettuata dal CASE,

dello schema “Final Physical” del capitolo precedente. Ogni attributo ha poi associato un campo indicatore tipico di tutti i buffer ODBC.

Essendo l’unico scopo di queste classi quello di funzionare da buffer, è stato ritenuto inutile utilizzare metodi di get e set e si è quindi deciso di mantenere pubblici i vari attributi.

L’unico metodo di tali classi è il costruttore, che semplicemente inizializza a NULL i vari campi, scrivendo SQL_NULL_DATA nell’indicatore corrispondente.

6.5 L’Interfaccia Utente

Scopo dell’interfaccia utente è richiedere i dati, richiamare i metodi del package “Data Warehouse Management” e infine presentare i risultati all’utente.

Questi semplici compiti possono essere realizzati sia da un’interfaccia grafica che testuale: trattandosi comunque di un applicativo per utenti esperti (il DBA), si è deciso inizialmente di costruire un semplice software testuale. Per avere informazioni aggiuntive su questa interfaccia si veda anche l’Appendice A.

Nulla vieta comunque di legare al package menzionato un’interfaccia di tipo grafico, magari implementata con filosofia di progettazione e programmazione ad oggetti.

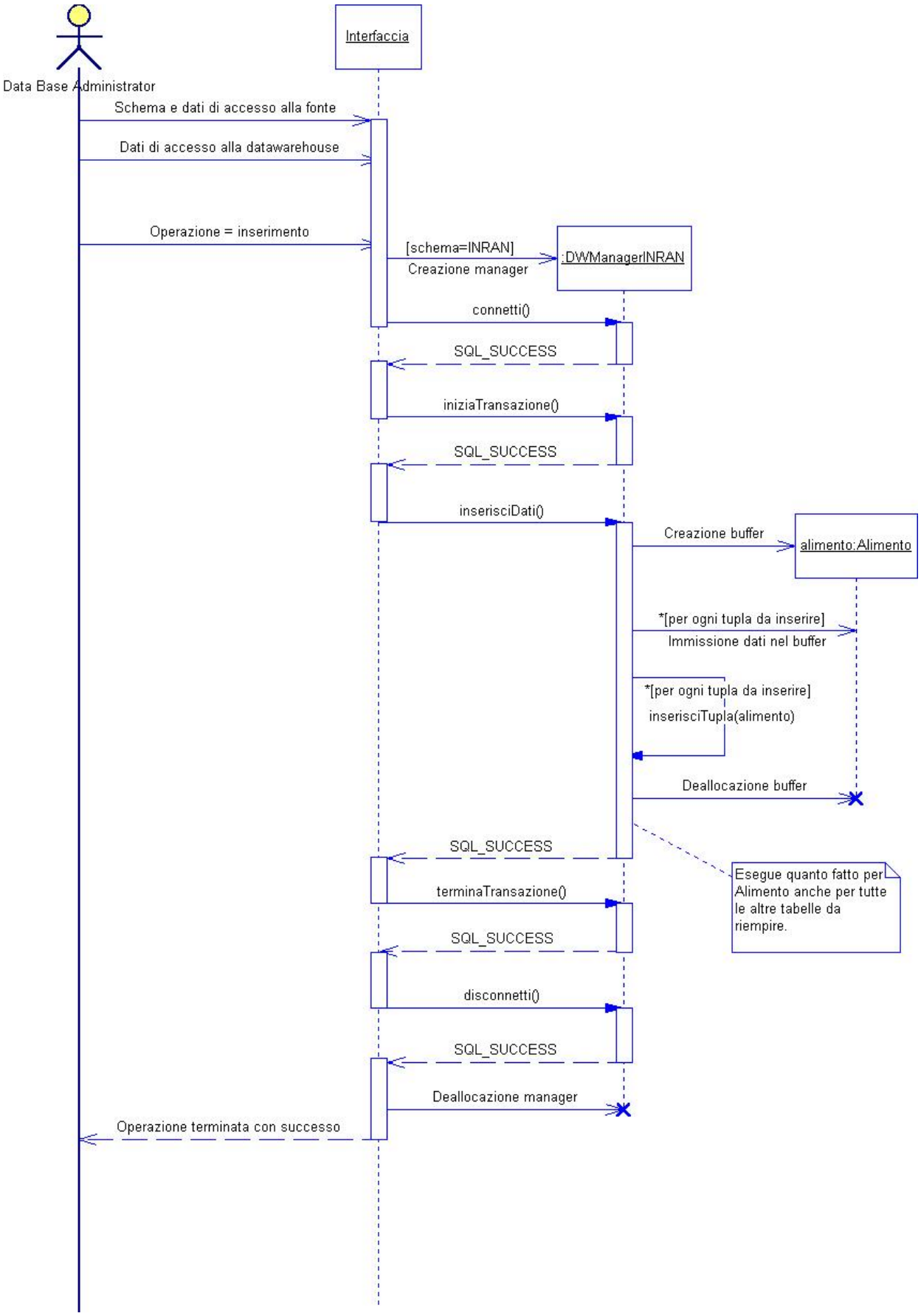
6.6 Il Comportamento Dinamico

Vediamo ora come tutte le parti mostrate in precedenza collaborano per realizzare i casi d’uso descritti all’inizio del capitolo.

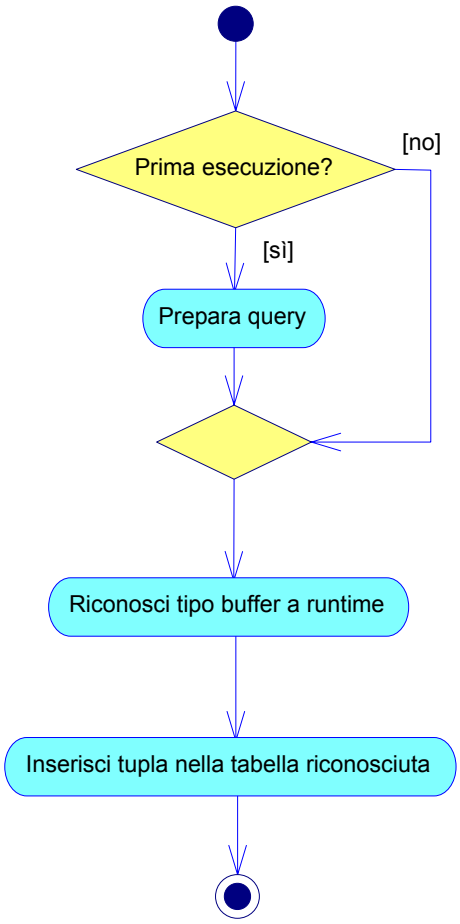
Nella figura seguente è mostrato un sequence diagram che presenta in modo abbastanza dettagliato un possibile scenario del caso d’uso “Inserisci fonte”.

I restanti due casi d’uso hanno svolgimenti analoghi e quindi per essi non vengono riportati esempi illustrativi.

Si noti anche come lo schema seguente fornisca ulteriori dettagli al sequence diagram mostrato in precedenza per lo stesso caso d’uso.


Data Base Administrator
Interfaccia
Schema e dati di accesso alla fonte
Dati di accesso alla datawarehouse
Operazione = inserimento
[schema=INRAN]
Creazione manager
:DWManagerINRAN
connetti()
SQL_SUCCESS
iniziaTransazione()
SQL_SUCCESS
inserisciDati()
Creazione buffer
alimento:Alimento
*[per ogni tupla da inserire]
Immissione dati nel buffer
*[per ogni tupla da inserire]
inserisciTupla(alimento)
Deallocazione buffer
SQL_SUCCESS
Esegue quanto fatto per Alimento anche per tutte le altre tabelle da riempire.
terminaTransazione()
SQL_SUCCESS
disconnetti()
SQL_SUCCESS
Deallocazione manager
Operazione terminata con successo

A completamento del diagramma precedente presentiamo anche un activity diagram che illustra il comportamento della funzione **inserisciTupla()**: si fanno notare in particolar modo l'utilizzo della preparazione delle query prima della loro esecuzione e il riconoscimento a run time del tipo di buffer passato come parametro (RTTI).



6.7 Aggiunta di Altre Fonti

Uno dei requisiti più importanti cui il software risponde è quello dell'estensione ad ulteriori sorgenti di dati.

Per venire incontro a questa esigenza, la gerarchia del package "Data Warehouse Management" è estendibile in due punti: innanzitutto si può aggiungere un ulteriore classe derivata da DWManagerFonteODBC o da DWManagerFonteASCII. Questa classe deve ridefinire semplicemente le due funzioni inserisciDati() e aggiornaDati(), utilizzando tra l'altro i metodi protetti messi a disposizione dalle classi base.

Oltre a ciò, si può estendere la gerarchia aggiungendo una classe derivata da "DWManager": si può infatti voler accedere alla fonte in modo differente da ODBC o dallo strema di input per file del C (ad esempio, si potrebbe decidere di utilizzare le API C di un particolare DBMS, cosa possibile anche per PostgreSQL). Questa estensione "al secondo livello" della gerarchia dovrebbe comunque uniformarsi alle classi già create: bisognerebbe cioè che questa nuova classe ridefinisca semplicemente i metodi connetti() e disconnetti(), fornendo nel contempo alle classi derivate degli attributi che facciano da handle per la fonte. Il restante comportamento (la definizione di inserisciDati() e aggiornaDati()) è invece compito del terzo livello della gerarchia e differisce ovviamente da fonte a fonte.

6.8 Problema Implementativo: i Driver ODBC per Access

Un problema, riscontrato in fase di implementazione, è stata la sperimentalità dei driver ODBC per Access sotto Linux. Il progetto MDB Tools infatti è ancora in essere e i driver menzionati non permettono molte importanti operazioni SQL (una per tutte: non è stato ancora implementato il join).

Tale problema ha dato vita, tra l'altro, a due interessanti soluzioni, che vengono illustrate nei paragrafi seguenti. La soluzione più banale, cioè quella di salvare in altro formato le tabelle tramite Access stesso, non è stata esplorata, in quanto anche i formati alternativi possibili non possedevano driver ODBC per Linux stabili.

6.8.1 Prima Soluzione: ODBC-ODBC Bridge

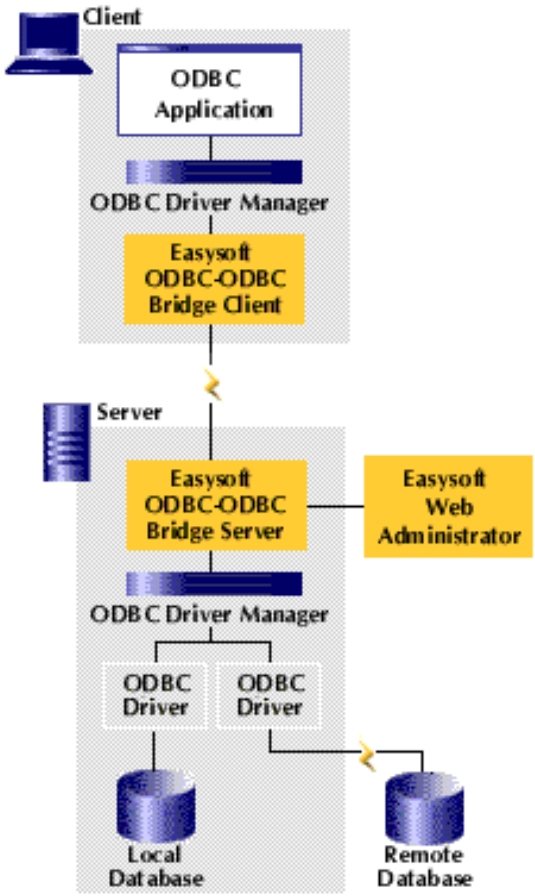
La prima interessantissima soluzione al problema descritto in precedenza è l'utilizzo di un programma shareware denominato ODBC-ODBC Bridge (OOB) distribuito da Easysoft (partner del progetto unixODBC) e reperibile al sito: http://www.easysoft.com/products/oob/main.phtml.

Come ampiamente documentato sul sito della compagnia (e anche su quello di unixODBC) questo "ponte" è un'applicazione client-server per Windows e Linux che

permette di accedere a fonti ODBC risiedenti su macchine in reti TCP/IP utilizzando il driver risiedente sul server.

Per risolvere il nostro problema è stata quindi configurata una macchina Windows come server e una Linux come client. In questo modo si è potuto accedere ai database Access utilizzando il ben più stabile driver costruito da Microsoft.

La figura seguente (presa dal sito Easysoft) mostra lo schema generico dell'applicazione OOB, ma illustra abilmente anche il nostro caso specifico qualora si ricordi che il SO del client è Linux e quello del server Windows. Si ricordi anche che i collegamenti possono avvenire su qualsiasi rete TCP/IP.



La configurazione presentata risolve a pieno questo problema e molti altri: ad esempio si possono utilizzare fonti i cui driver non siano affatto disponibili per Linux (come accade per SQLServer) oppure, viceversa, si possono alimentare datawarehouse risiedenti su Windows utilizzando dei moduli di feeding su macchina Linux.

Questa soluzione comunque ha il grosso svantaggio che il lato server dell'applicativo è shareware, con licenza per il pieno utilizzo solo per trenta giorni.

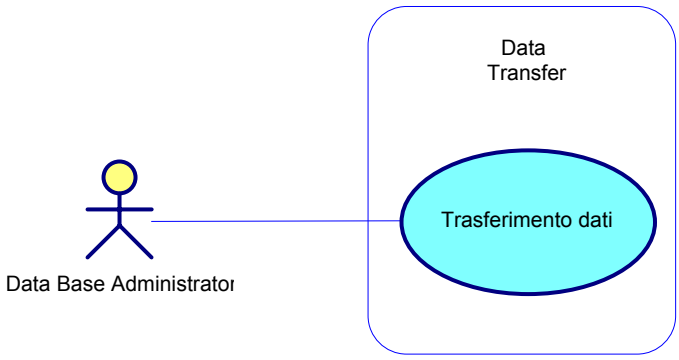
Essendo il nostro un progetto freeware, non è stata in definitiva applicata questa risoluzione, anche se la stessa è stata ampiamente esplorata date le sue enormi potenzialità.

6.8.2 Seconda Soluzione: un Applicativo di Trasferimento Dati

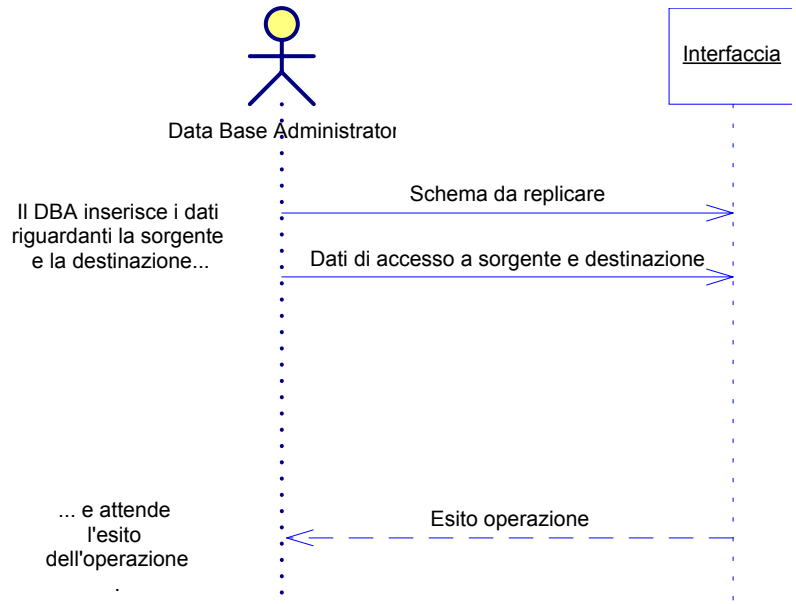
La seconda soluzione, che è poi quella adottata nel sistema attuale, è quella di scrivere un semplice applicativo di trasferimento dei dati tra due fonti ODBC di DBMS diversi.

Così infatti si può passare ad un DBMS che, come PostgreSQL, abbia dei driver per unixODBC stabili: questo è possibile perché l'applicazione compie sulla sorgente, che è quella possedente dei driver sperimentali, delle query semplicissime e quindi comunque funzionanti (qualsiasi driver ODBC che voglia essere definito tale deve supportare interrogazioni del tipo "SELECT lista_colonne FROM nome_tabella").

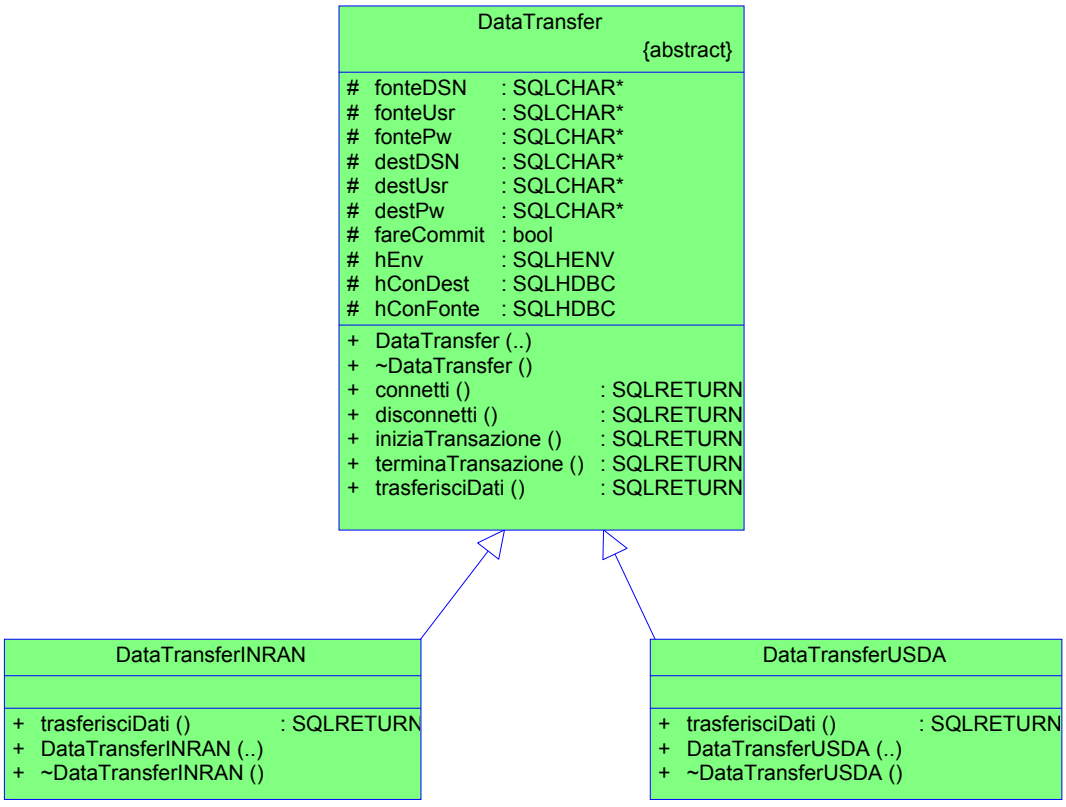
L'applicazione in questione è stata denominata "DataTransfer" ed ha il solo semplice caso d'uso mostrato in figura.



Il comportamento atteso dall'applicazione ad alto livello viene invece illustrato nel sequence diagram seguente: l'utente specifica il DSN della sorgente, quello della destinazione e lo schema da replicare (es. INRAN, etc.) e quindi semplicemente attende la fine del trasferimento.



Per raggruppare le funzionalità comuni a tutti gli schemi da replicare è stata infine costruita la semplice gerarchia mostrata nel class diagram seguente, per un approfondimento del quale si rimanda al materiale allegato alla tesi, in quanto il significato dei metodi e degli attributi dovrebbe essere chiaro alla luce di quanto detto per l'applicativo "Feeder".



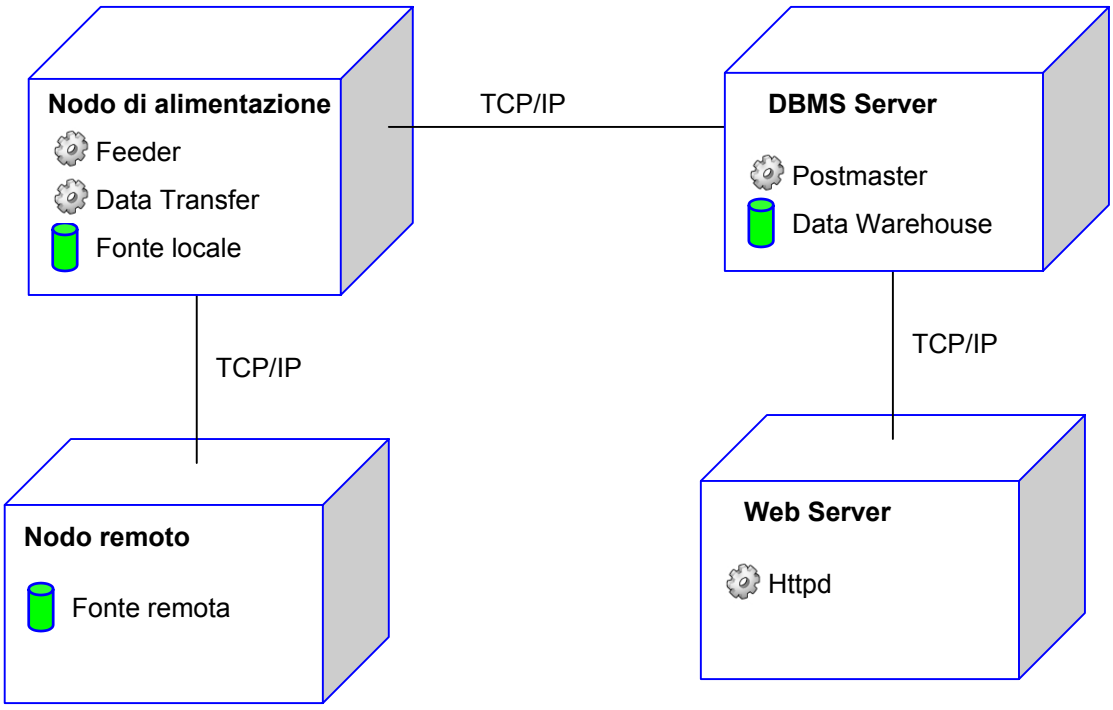
E' importante notare che il cambio di DBMS non influenza minimamente il modulo di feeding illustrato in precedenza: basterà semplicemente indicargli come fonte il Data Source Name ODBC dello schema replicato.

Questa soluzione al problema è comunque da ritenersi provvisoria, in attesa dell'acquisto del software Easysoft o nella speranza che termini lo sviluppo dei driver ODBC per Access sotto Linux.

Anche questo software è dotato di una semplice interfaccia testuale, di cui si parla anche nell'Appendice A.

6.9 Architettura del Sistema

A completamento di quanto detto finora, si mostra di seguito un deployment diagram con una possibile disposizione dei componenti tra le varie unità di calcolo. Le relazioni tra i componenti non vengono illustrate in quanto dovrebbero risultare chiare alla luce della discussione precedente.



Nel diagramma precedente non compaiono i client, che ovviamente sono connessi (sempre tramite una rete TCP/IP) con il web server, che nell'esempio mostrato è Apache.

Naturalmente questa disposizione prevede la disponibilità di un buon numero di calcolatori: tutto il sistema può comunque risiedere al limite anche su un solo computer, con la sola eccezione che serve ovviamente una rete se su due nodi devono risiedere sistemi operativi differenti (cosa necessaria ad esempio per l'utilizzo dell'ODBC-ODBC Bridge visto prima).

CAPITOLO SETTE

IL PROTOTIPO DI SITO DI CONSULTAZIONE

Questo capitolo è interamente dedicato all'unico aspetto del progetto pienamente visibile all'utente finale. Grande cura è stata quindi posta nel presentare le informazioni seguenti con chiarezza e semplicità.

7.1 Caratteristiche di Base del Sito

Il primo software progettato per appoggiarsi sui dati di cui si è discusso finora è un applicativo di consultazione, basato su interfaccia web.

Il sito è da considerarsi ancora un prototipo, non tanto per la mancanza di alcune funzionalità minori, ma soprattutto perché è privo della parte in lingua inglese. Inoltre manca del tutto una veste grafica accattivante: per ora si ha solo uno spartano testo nero su sfondo bianco.

Questo capitolo mira fondamentalmente a spiegare la struttura e l'utilizzo del sito, accennando solo alla descrizione della fase di progettazione che, trattandosi di un prototipo, è stata comunque abbastanza rapida.

Come anticipato nel primo capitolo, è stato utilizzato il linguaggio di scripting server-side PHP. Questo garantisce, tra l'altro, anche una notevole compatibilità con tutti i browser disponibili sul mercato: il sito ha dimostrato di funzionare senza problemi sia con Internet Explorer e Opera sotto Windows, sia con Mozilla nel sistema operativo Linux (si ricorda a tal proposito che Mozilla è basato sul motore di Netscape Navigator).

L'utilizzo di PHP ha inoltre permesso di instaurare delle vere e proprie sessioni utente: esistono dati che accompagnano il singolo utilizzatore durante tutta la propria permanenza sulle pagine web del sito, consentendo così una maggiore personalizzazione dello stesso. Per ulteriori informazioni su questo e altri aspetti di PHP si veda la documentazione ufficiale (al sito http://www.php.net) oppure l'ampia manualistica disponibile sul mercato, tra cui segnaliamo [14].

7.2 La Data Warehouse

Una volta stabilite le funzionalità del sito, è stata creata una data warehouse, basandosi sui dati del database di integrazione.

Sono stati innanzitutto aggiunti alla base di dati quattro schemi (intesi come "schema" SQL), denominati "italiano", "italianopiu", "inglese" e "inglesepiu". Il primo è atto a ospitare le informazioni presenti nel database nella sola lingua italiana, mentre il secondo contiene i dati che sono in italiano, più eventuali dati in altre lingue qualora

non esistesse di queste informazioni una versione italiana. Analogo significato hanno i due schemi in inglese: a tal proposito si fa notare che per la creazione del sito in lingua inglese esiste già il substrato di dati.

Ognuno degli schemi creati ospita poi delle viste materializzate, contenenti i dati nella lingua corrispondente, su cui è stato realizzato un buon numero di indici, per favorire le ricerche.

In ultimo è stato creato un utente con privilegi di sola lettura ("select") sulle tabelle dei quattro schemi citati: è questa infatti l'unica operazione che il web server deve poter compiere sui dati.

7.3 La Scelta della Fonti

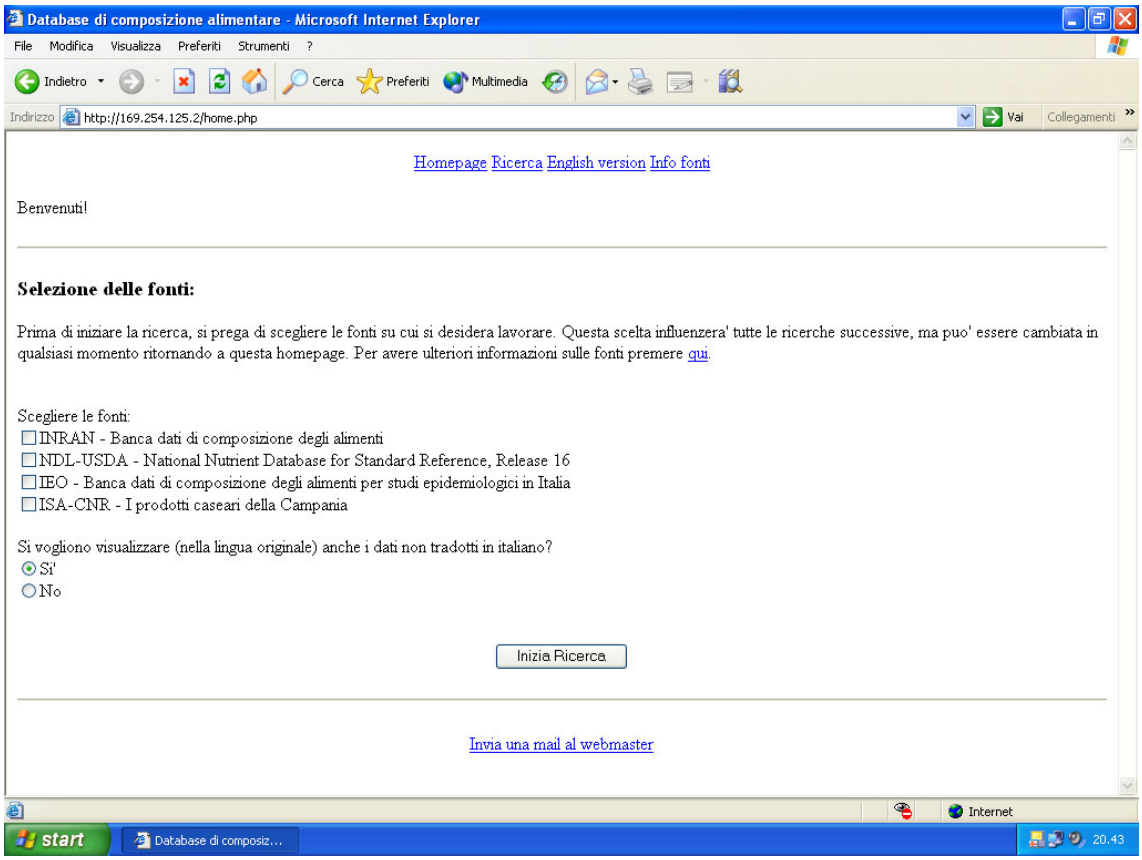
La prima scelta cui si trova di fronte l'utente qualora decidesse di voler iniziare una ricerca sui dati del database è quella di selezionare le fonti da consultare.

Nella pagina iniziale (homepage) l'utente sceglie le fonti su cui desidera lavorare: questa scelta seguirà l'utilizzatore per tutta la sua permanenza nel sito, anche se può essere modificata in qualsiasi momento ritornando all'homepage.

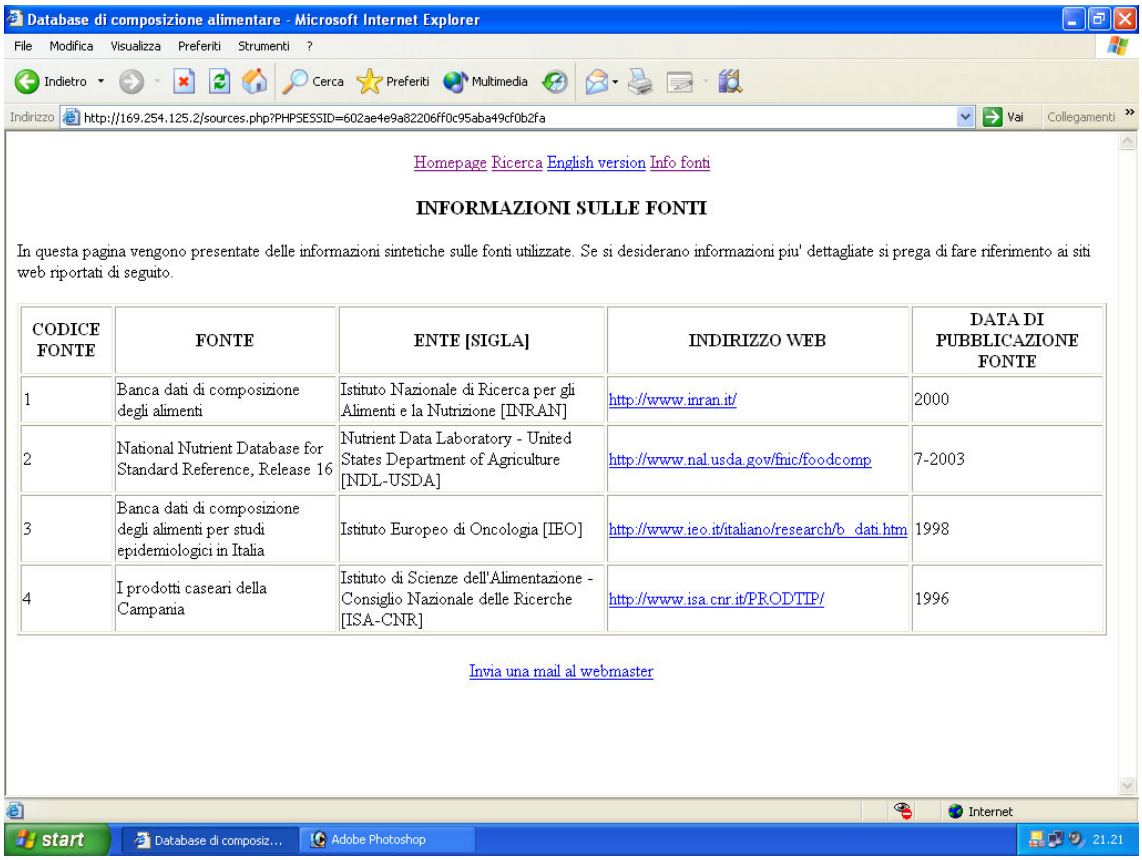
Altra decisione iniziale è quella di scegliere se si vogliono visualizzare anche le informazioni non ancora tradotte in italiano: questo si traduce, nella data warehouse, nell'utilizzo dello schema "italianopiu" in caso affermativo, dello schema "italiano" altrimenti.

Anche questa informazione fa parte della sessione utente: a meno di tornare alla pagina iniziale e cambiare questa scelta, l'utilizzatore si porterà dietro questa decisione per tutta la navigazione nel sito.

La figura seguente mostra la parte della pagina iniziale nella quale si effettuano le decisioni descritte in precedenza:



Da qualsiasi pagina è poi possibile accedere ai link dei siti delle fonti, come testimoniato dallo screenshot successivo:

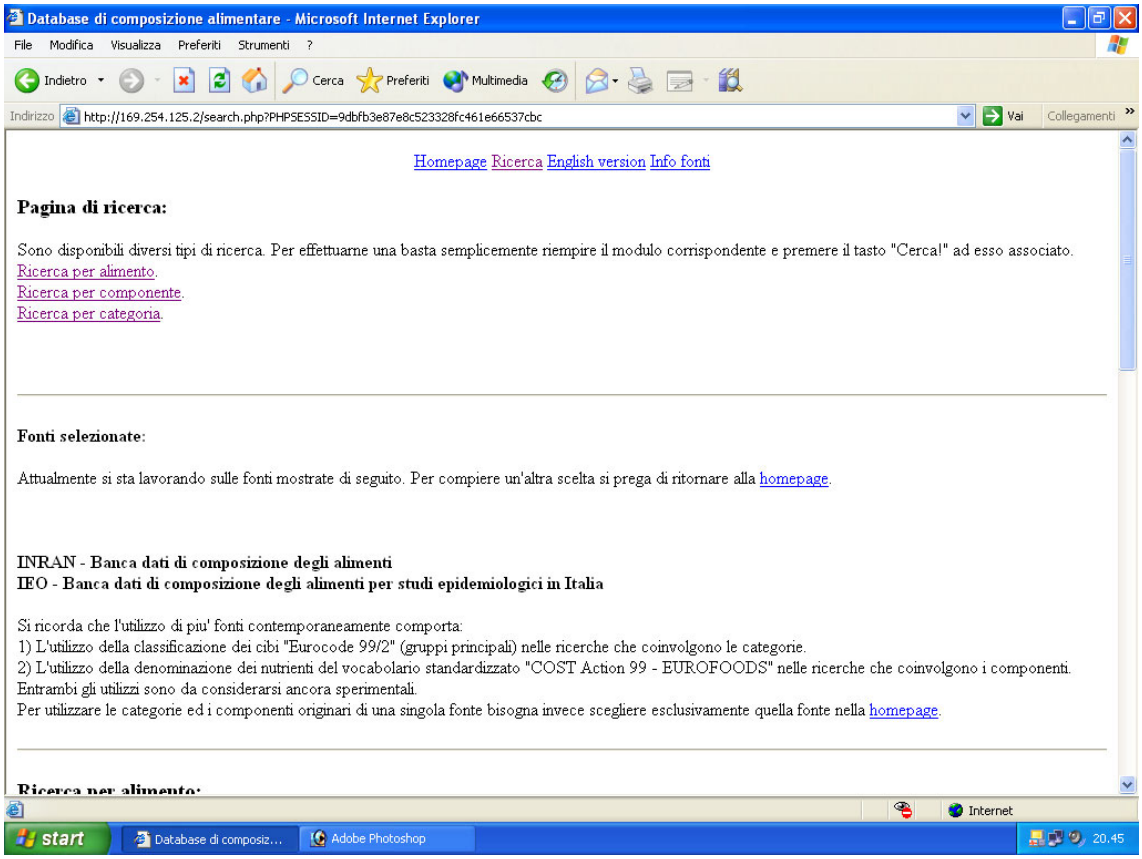


7.4 La Ricerca nel Database

Sono al momento disponibili, come testimoniato dall'immagine seguente, tre tipi di ricerca, ognuno con opzioni particolari: per alimento, per componente e per categoria.

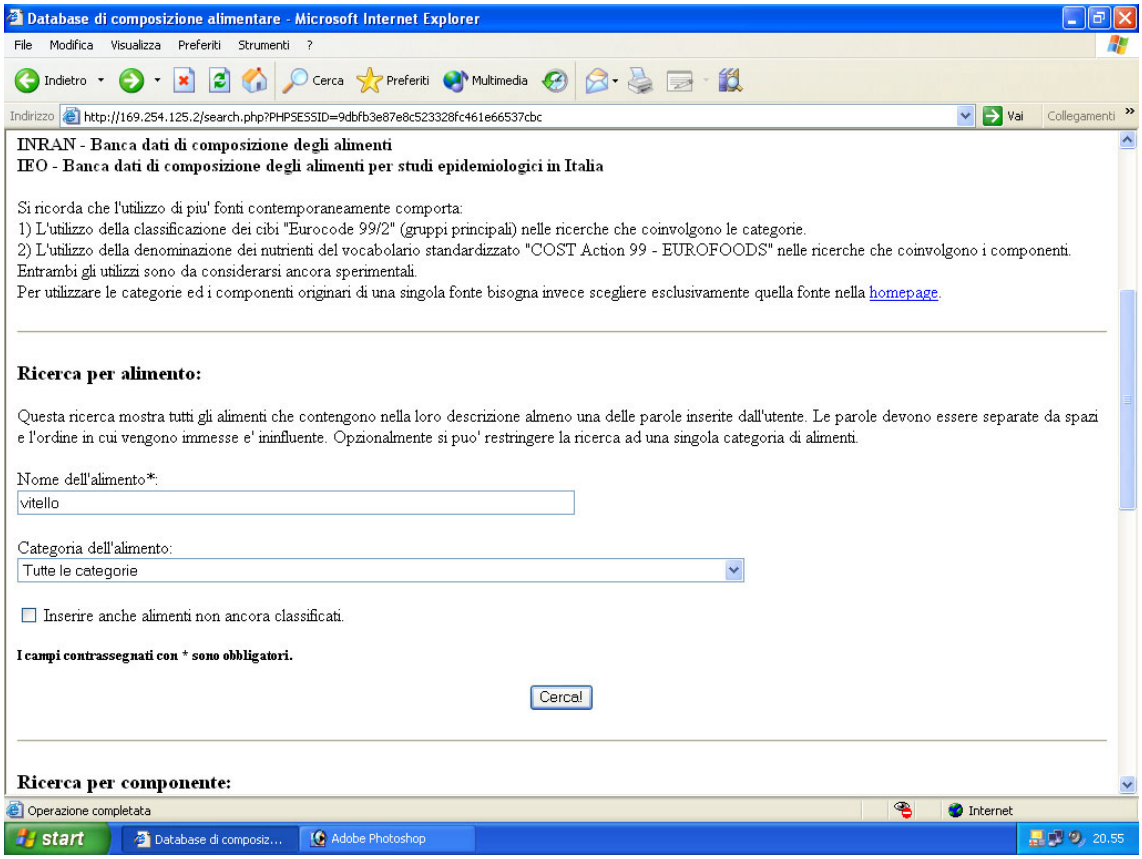
Nella pagina di ricerca viene ricordato all'utente su quali fonti sta lavorando. Si avvisa sempre inoltre che adoperare più fonti comporta, per alcune ricerche, anche l'utilizzo di certe caratteristiche sperimentali: vengono impiegati infatti i componenti e le categorie "standard", di cui si è già abbondantemente discusso nel capitolo cinque.

Se si seleziona invece una sola fonte alla volta, si rendono automaticamente disponibili le categorie ed i componenti originari di quella particolare fonte.

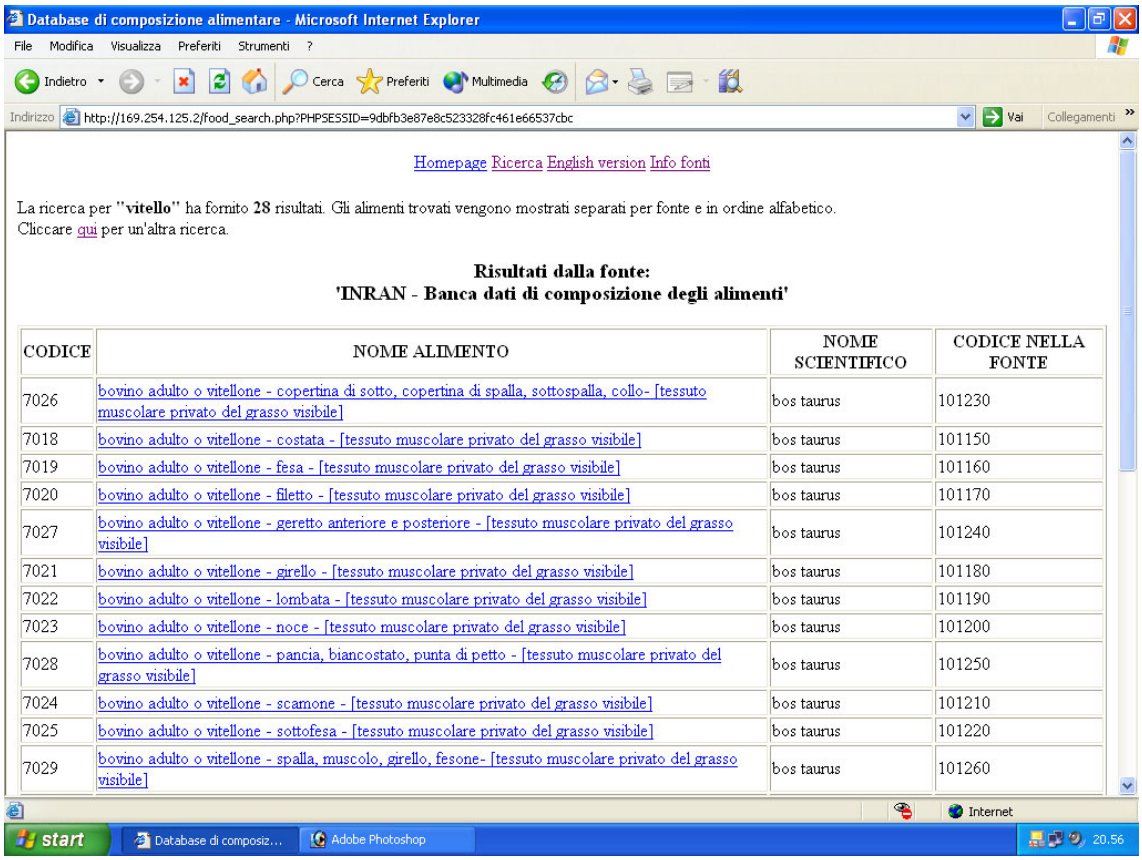


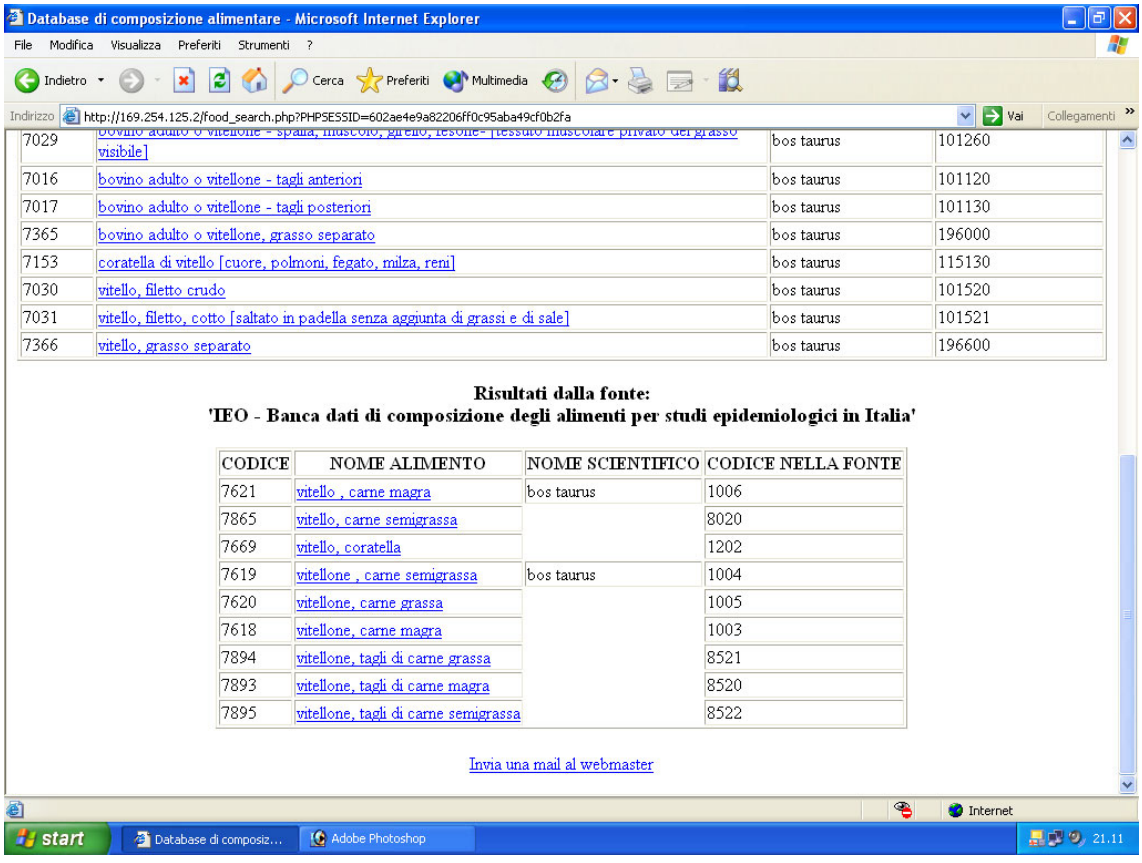
7.4.1 La Ricerca per Alimento

Come brevemente descritto nella pagina web stessa, questa ricerca dà la possibilità di inserire una o più parole descriventi l'alimento (le parole vengono poi utilizzate come se fossero legate dal connettivo logico "or"), come in un qualsiasi motore di ricerca. E' poi possibile restringere la ricerca ad una particolare categoria, includendo o meno anche gli alimenti che non appartengono ad alcuna categoria. Quest'ultima opzione appare solo se si utilizzano più fonti contemporaneamente, in quanto al momento esistono alcuni alimenti (della sola fonte USDA) che non sono stati ancora classificati con le categorie "standard".



Come conseguenza della pressione del bottone "Cerca!", appare la lista degli alimenti che rispetta le condizioni immesse dall'utente. Un elenco di esempio (dove è stata cercata la parola "vitello") è mostrato nelle due figure seguenti, dove è da notare che in questo caso gli alimenti sono divisi per fonte.


Database di composizione alimentare - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Cerca Preferiti Multimedia
Indirizzo http://169.254.125.2/food_search.php?PHPSESSID=9dbfb3e87e8c523328fc461e66537cbc
Vai Collegamenti
Homepage Ricerca English version Info fonti
La ricerca per "vitello" ha fornito 28 risultati. Gli alimenti trovati vengono mostrati separati per fonte e in ordine alfabetico.
Cliccare qui per un'altra ricerca.
Risultati dalla fonte:
'INRAN - Banca dati di composizione degli alimenti'
CODICE | NOME ALIMENTO | NOME SCIENTIFICO | CODICE NELLA FONTE
7026 | bovino adulto o vitellone - copertina di sotto, copertina di spalla, sottospalla, collo- [tessuto muscolare privato del grasso visibile] | bos taurus | 101230
7018 | bovino adulto o vitellone - costata - [tessuto muscolare privato del grasso visibile] | bos taurus | 101150
7019 | bovino adulto o vitellone - fesa - [tessuto muscolare privato del grasso visibile] | bos taurus | 101160
7020 | bovino adulto o vitellone - filetto - [tessuto muscolare privato del grasso visibile] | bos taurus | 101170
7027 | bovino adulto o vitellone - geretto anteriore e posteriore - [tessuto muscolare privato del grasso visibile] | bos taurus | 101240
7021 | bovino adulto o vitellone - girello - [tessuto muscolare privato del grasso visibile] | bos taurus | 101180
7022 | bovino adulto o vitellone - lombata - [tessuto muscolare privato del grasso visibile] | bos taurus | 101190
7023 | bovino adulto o vitellone - noce - [tessuto muscolare privato del grasso visibile] | bos taurus | 101200
7028 | bovino adulto o vitellone - pancia, biancostato, punta di petto - [tessuto muscolare privato del grasso visibile] | bos taurus | 101250
7024 | bovino adulto o vitellone - scamone - [tessuto muscolare privato del grasso visibile] | bos taurus | 101210
7025 | bovino adulto o vitellone - sottofesa - [tessuto muscolare privato del grasso visibile] | bos taurus | 101220
7029 | bovino adulto o vitellone - spalla, muscolo, girello, fesone- [tessuto muscolare privato del grasso visibile] | bos taurus | 101260
Internet
start
Database di composiz...
Adobe Photoshop
20.56

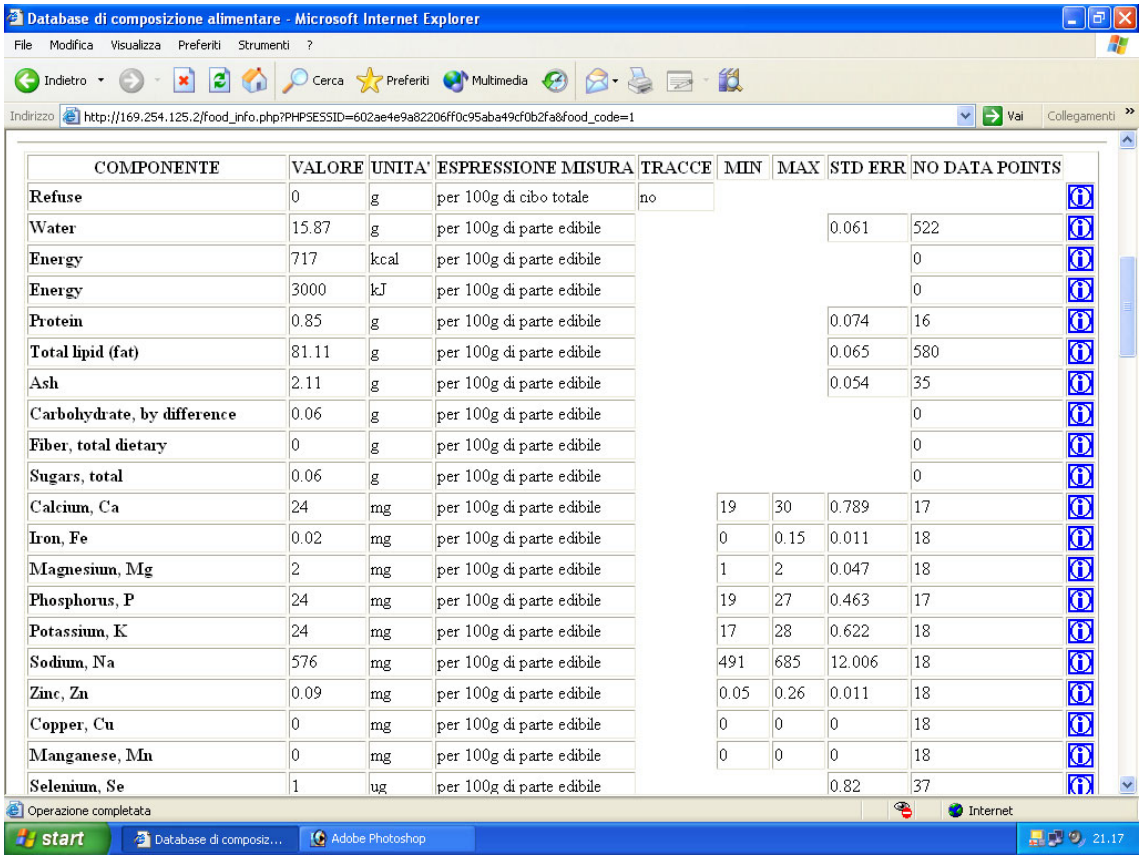

Database di composizione alimentare - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Cerca Preferiti Multimedia
Indirizzo http://169.254.125.2/food_search.php?PHPSESSID=602ae4e9a82206ff0c95aba49cf0b2fa
Vai Collegamenti
7029 | bovino adulto o vitellone - spalla, muscolo, girello, fesone- [tessuto muscolare privato del grasso visibile] | bos taurus | 101260
7016 | bovino adulto o vitellone - tagli anteriori | bos taurus | 101120
7017 | bovino adulto o vitellone - tagli posteriori | bos taurus | 101130
7365 | bovino adulto o vitellone, grasso separato | bos taurus | 196000
7153 | coratella di vitello [cuore, polmoni, fegato, milza, reni] | bos taurus | 115130
7030 | vitello, filetto crudo | bos taurus | 101520
7031 | vitello, filetto, cotto [saltato in padella senza aggiunta di grassi e di sale] | bos taurus | 101521
7366 | vitello, grasso separato | bos taurus | 196600
Risultati dalla fonte:
'IEO - Banca dati di composizione degli alimenti per studi epidemiologici in Italia'
CODICE | NOME ALIMENTO | NOME SCIENTIFICO | CODICE NELLA FONTE
7621 | vitello , carne magra | bos taurus | 1006
7865 | vitello, carne semigrassa | | 8020
7669 | vitello, coratella | | 1202
7619 | vitellone , carne semigrassa | bos taurus | 1004
7620 | vitellone, carne grassa | | 1005
7618 | vitellone, carne magra | | 1003
7894 | vitellone, tagli di carne grassa | | 8521
7893 | vitellone, tagli di carne magra | | 8520
7895 | vitellone, tagli di carne semigrassa | | 8522
Invia una mail al webmaster
Internet
start
Database di composiz...
Adobe Photoshop
21.11

Cliccando poi sul nome di un singolo alimento se ne ottiene la relativa scheda, che mostra delle informazioni generiche e la composizione del cibo stesso. Nell'esempio vediamo la scheda dell'alimento "butter, salted", preso dalla fonte USDA (che è quella in cui sono presenti più informazioni aggiuntive). La prima immagine mostra le caratteristiche generiche dell'alimento:

La seconda parte della scheda invece illustra la composizione vera e propria, arricchita dalle più importanti informazioni statistiche.

Da notare che in queste schede compositive vengono sempre utilizzate le denominazioni dei componenti della fonte (possibilmente tradotte nella lingua in uso) e non i nomi del vocabolario "standard".

L'ordine con cui vengono mostrati i componenti inoltre è lo stesso dei report creati dalle fonti primarie nei rispettivi applicativi e/o libri.



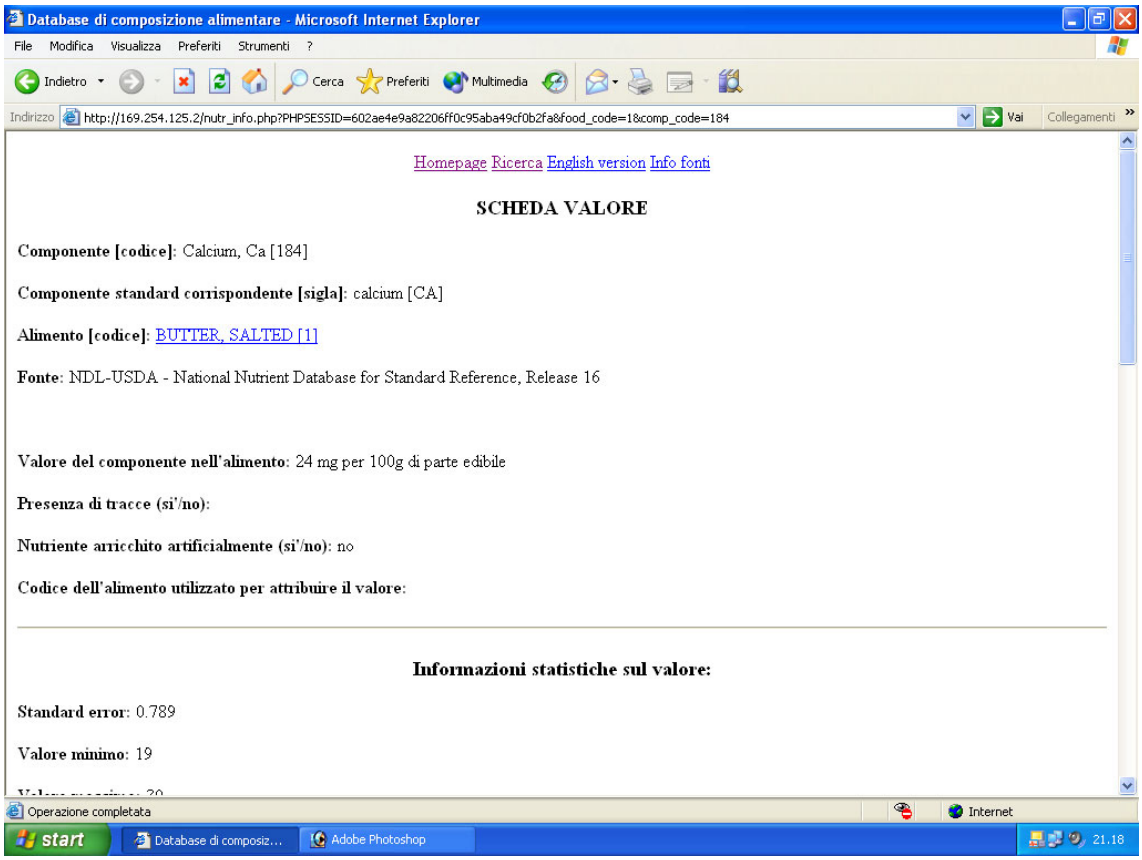
COMPONENTE	VALORE	UNITA'	ESPRESSIONE MISURA	TRACCE	MIN	MAX	STD ERR	NO DATA POINTS	
Refuse	0	g	per 100g di cibo totale	no					
Water	15.87	g	per 100g di parte edibile				0.061	522	
Energy	717	kcal	per 100g di parte edibile					0	
Energy	3000	kJ	per 100g di parte edibile					0	
Protein	0.85	g	per 100g di parte edibile				0.074	16	
Total lipid (fat)	81.11	g	per 100g di parte edibile				0.065	580	
Ash	2.11	g	per 100g di parte edibile				0.054	35	
Carbohydrate, by difference	0.06	g	per 100g di parte edibile					0	
Fiber, total dietary	0	g	per 100g di parte edibile					0	
Sugars, total	0.06	g	per 100g di parte edibile					0	
Calcium, Ca	24	mg	per 100g di parte edibile		19	30	0.789	17	
Iron, Fe	0.02	mg	per 100g di parte edibile		0	0.15	0.011	18	
Magnesium, Mg	2	mg	per 100g di parte edibile		1	2	0.047	18	
Phosphorus, P	24	mg	per 100g di parte edibile		19	27	0.463	17	
Potassium, K	24	mg	per 100g di parte edibile		17	28	0.622	18	
Sodium, Na	576	mg	per 100g di parte edibile		491	685	12.006	18	
Zinc, Zn	0.09	mg	per 100g di parte edibile		0.05	0.26	0.011	18	
Copper, Cu	0	mg	per 100g di parte edibile		0	0	0	18	
Manganese, Mn	0	mg	per 100g di parte edibile		0	0	0	18	
Selenium, Se	1	ug	per 100g di parte edibile				0.82	37	

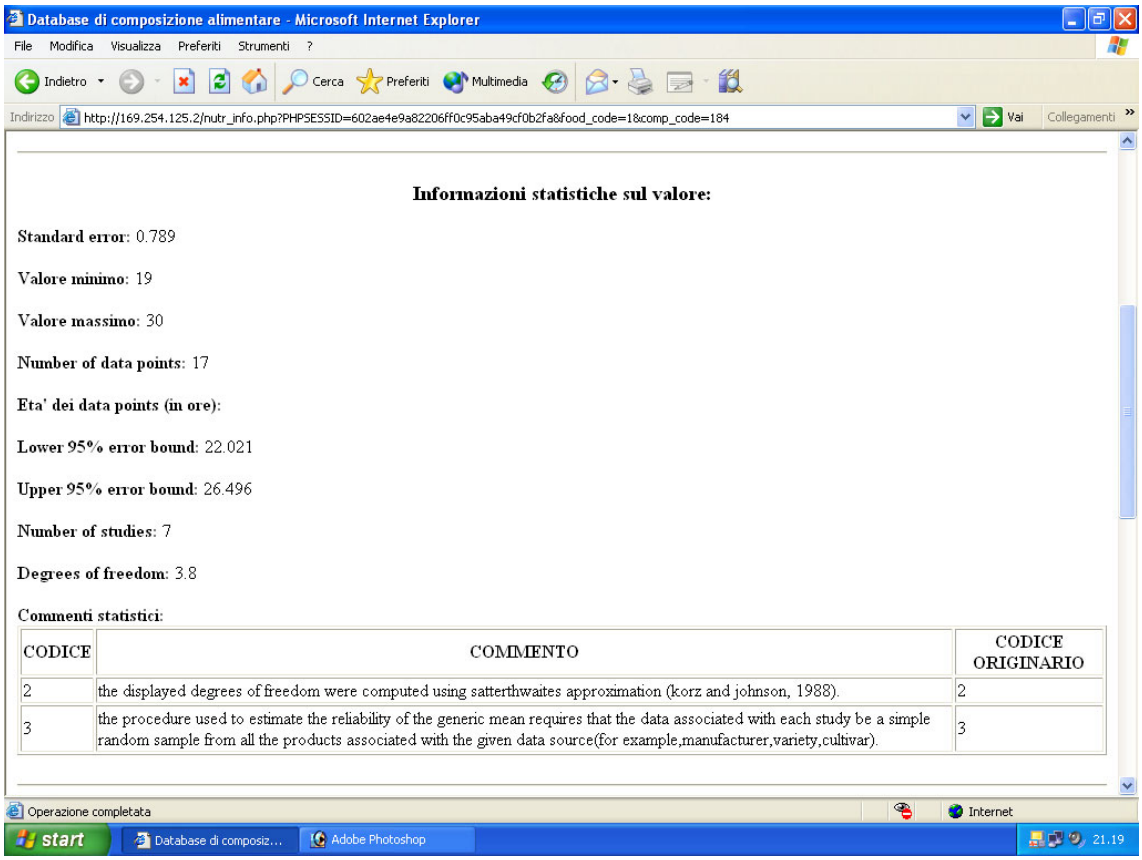
Ulteriori informazioni sul singolo valore sono accessibili nell'apposita scheda, che si può visualizzare cliccando sul simbolo blu d'informazione presente in ogni riga della tabella precedente.

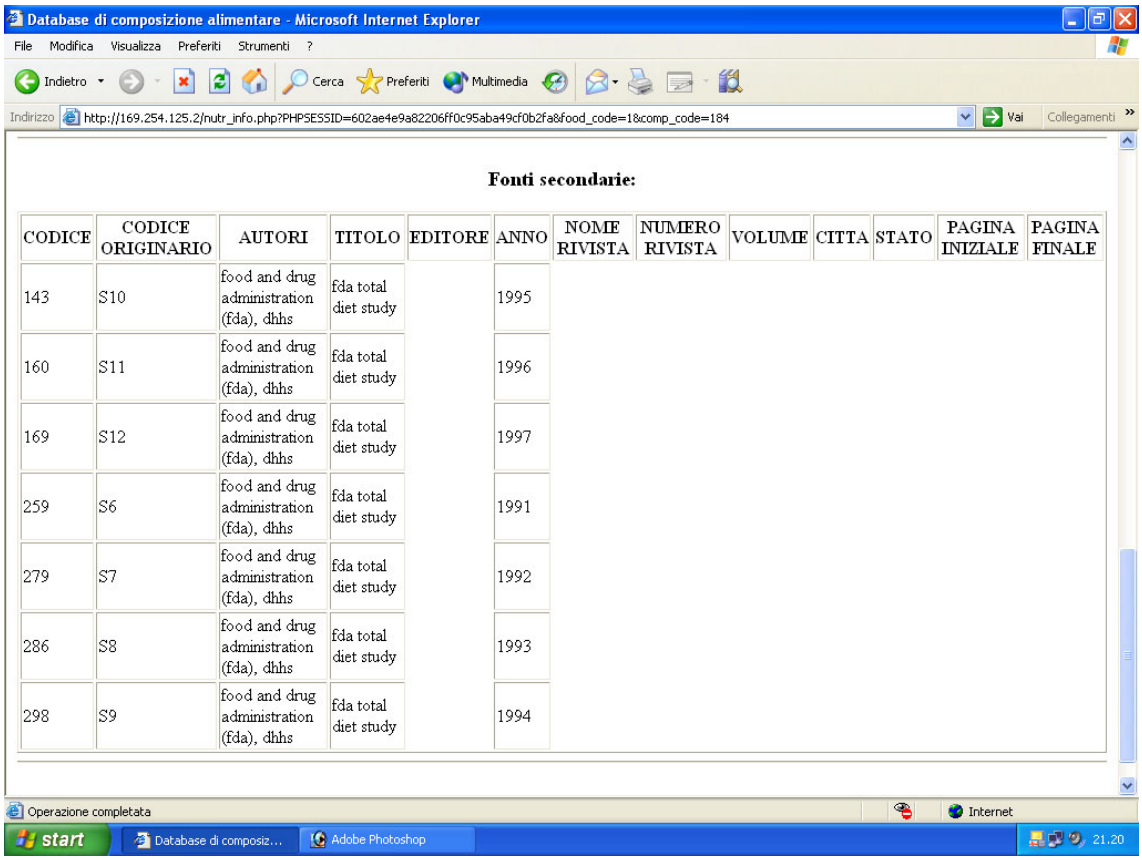
Anche in questo caso abbiamo innanzitutto dei dati generici, che vengono mostrati nella prima delle immagini seguenti. E' inoltre presente anche un link alla scheda dell'alimento (basta ciccare sul nome di quest'ultimo), utile quando si accede alla scheda del valore in altri modi (si veda a tal proposito la ricerca per componente)

Sempre nella scheda del valore, come illustrato dalla seconda figura, si possono ottenere informazioni statistiche più approfondite.

Infine ogni scheda si chiude con l'elenco delle eventuali fonti secondarie, cosa mostrata nell'ultimo dei tre screenshot.


Database di composizione alimentare - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Cerca Preferiti Multimedia
Indirizzo http://169.254.125.2/nutr_info.php?PHPSESSID=602ae4e9a82206ff0c95aba49cf0b2fa&food_code=18&comp_code=184
Vai Collegamenti
Homepage Ricerca English version Info fonti
SCHEDA VALORE
Componente [codice]: Calcium, Ca [184]
Componente standard corrispondente [sigla]: calcium [CA]
Alimento [codice]: BUTTER, SALTED [1]
Fonte: NDL-USDA - National Nutrient Database for Standard Reference, Release 16
Valore del componente nell'alimento: 24 mg per 100g di parte edibile
Presenza di tracce (si'/no):
Nutriente arricchito artificialmente (si'/no): no
Codice dell'alimento utilizzato per attribuire il valore:
Informazioni statistiche sul valore:
Standard error: 0.789
Valore minimo: 19
Operazione completata
Internet
start
Database di composiz...
Adobe Photoshop
21.18


Database di composizione alimentare - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Cerca Preferiti Multimedia
Indirizzo http://169.254.125.2/nutr_info.php?PHPSESSID=602ae4e9a82206ff0c95aba49cf0b2fa&food_code=18&comp_code=184
Vai Collegamenti
Informazioni statistiche sul valore:
Standard error: 0.789
Valore minimo: 19
Valore massimo: 30
Number of data points: 17
Eta' dei data points (in ore):
Lower 95% error bound: 22.021
Upper 95% error bound: 26.496
Number of studies: 7
Degrees of freedom: 3.8
Commenti statistici:
CODICE
COMMENTO
CODICE ORIGINARIO
2
the displayed degrees of freedom were computed using satterthwaites approximation (korz and johnson, 1988).
2
3
the procedure used to estimate the reliability of the generic mean requires that the data associated with each study be a simple random sample from all the products associated with the given data source(for example,manufacturer,variety,cultivar).
3
Operazione completata
Internet
start
Database di composiz...
Adobe Photoshop
21.19

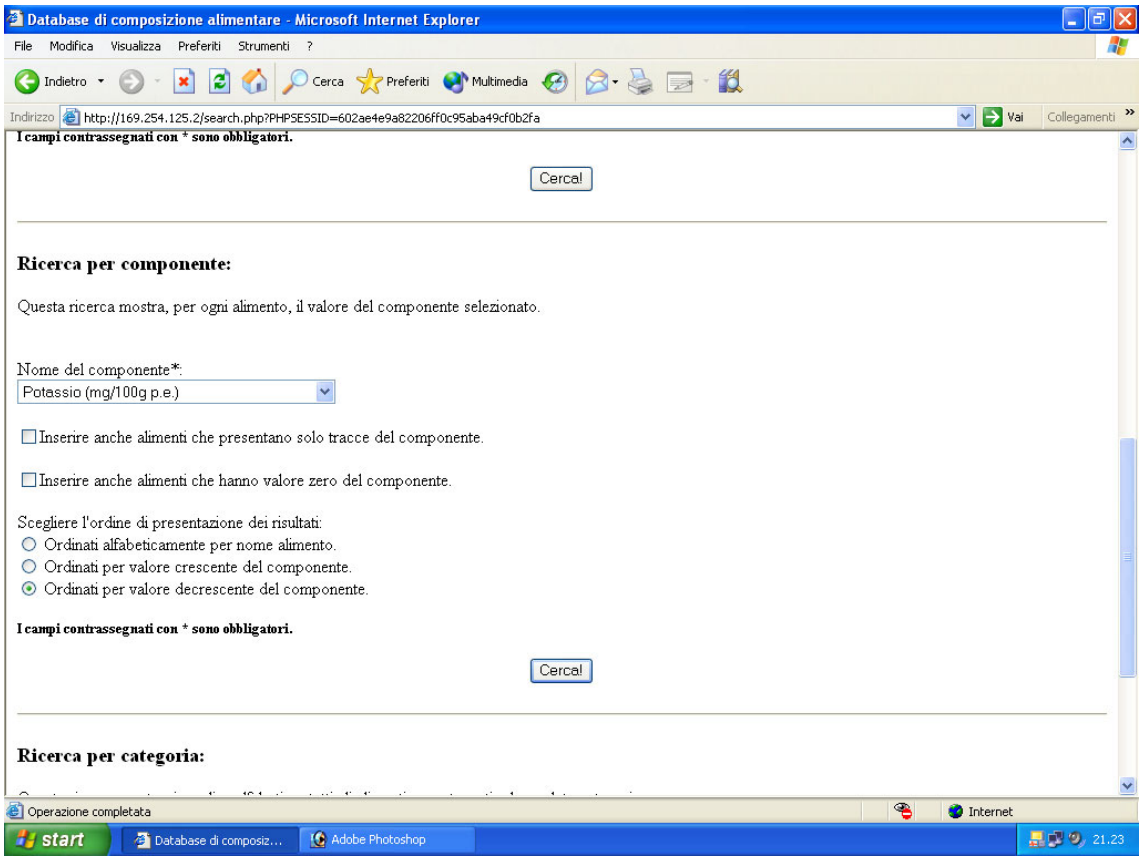


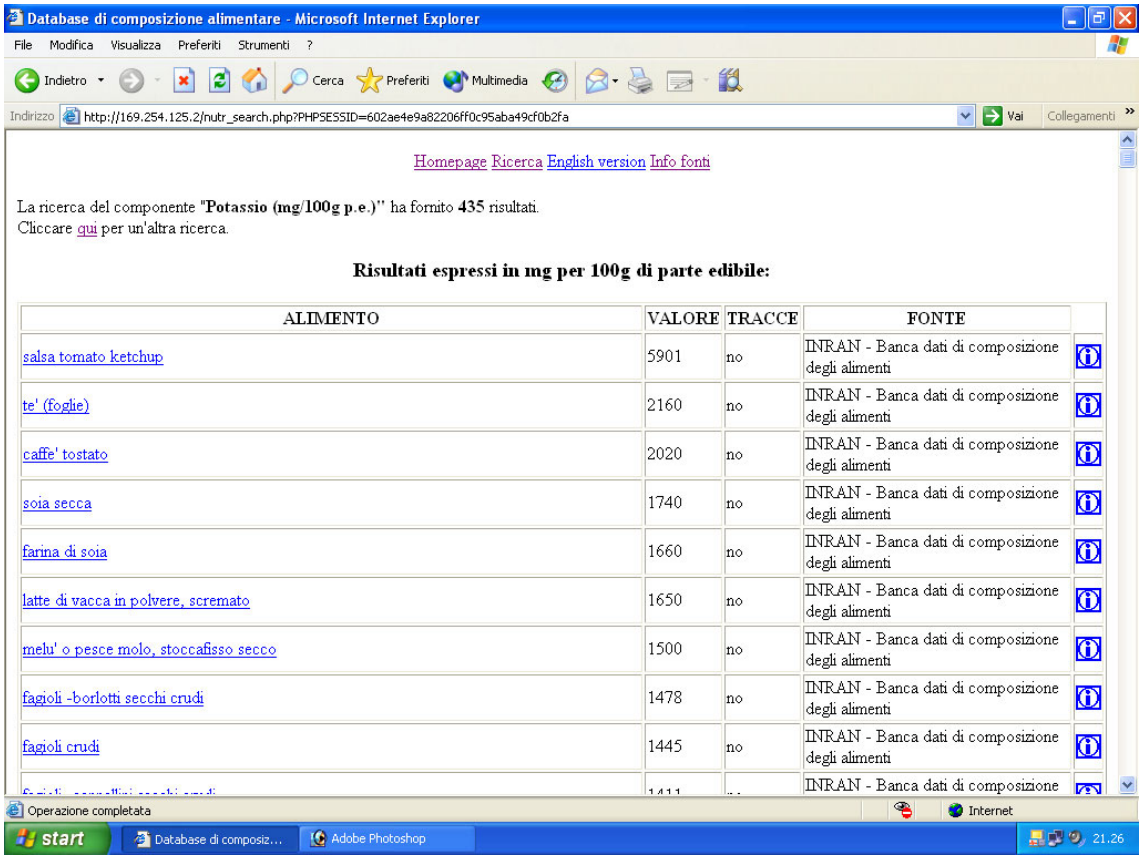
7.4.2 La Ricerca per Componente

Un ulteriore utilissimo strumento di ricerca e confronto è quello della ricerca per componente. Tramite esso è possibile ottenere una lista di tutti gli alimenti delle fonti selezionate che contengono quel dato componente.

Tale lista può essere ordinata in vari modi e può includere o meno gli alimenti per cui il valore del componente è zero o non quantificabile (il nutriente si dice in tal caso presente “in tracce”). Si consiglia comunque, per motivi di leggibilità della lista e per velocizzare la query, di evitare di inserire gli alimenti con contenuto nullo o quasi nullo del componente. Questa ricerca infatti, essendo presenti più di ottomila e ottocento alimenti nel database, è intrinsecamente lenta e dà performance abbastanza buone solo quando il numero di fonti selezionate è esiguo.

Nella figura immediatamente successiva mostriamo il modulo (o form) da riempire per effettuare questa ricerca, mentre in quella seguente si vede un esempio di elenco creato dal sistema.


Database di composizione alimentare - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Cerca Preferiti Multimedia
Indirizzo http://169.254.125.2/search.php?PHPSESSID=602ae4e9a82206ff0c95aba49cf0b2fa
Vai Collegamenti
I campi contrassegnati con * sono obbligatori.
Cerca!
Ricerca per componente:
Questa ricerca mostra, per ogni alimento, il valore del componente selezionato.
Nome del componente*:
Potassio (mg/100g p.e.)
Inserire anche alimenti che presentano solo tracce del componente.
Inserire anche alimenti che hanno valore zero del componente.
Scegliere l'ordine di presentazione dei risultati:
Ordinati alfabeticamente per nome alimento.
Ordinati per valore crescente del componente.
Ordinati per valore decrescente del componente.
I campi contrassegnati con * sono obbligatori.
Cerca!
Ricerca per categoria:
Operazione completata
Internet
start
Database di composiz...
Adobe Photoshop
21.23

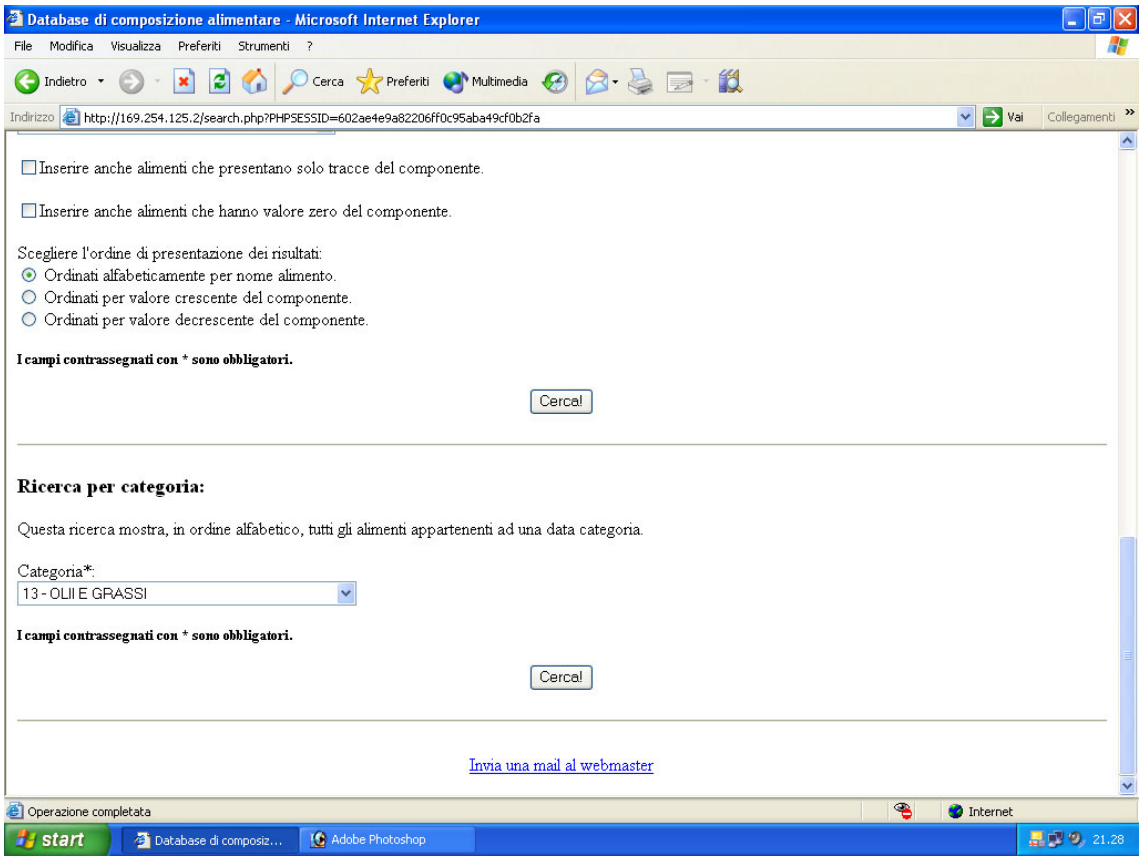

Database di composizione alimentare - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro Cerca Preferiti Multimedia
Indirizzo http://169.254.125.2/nutr_search.php?PHPSESSID=602ae4e9a82206ff0c95aba49cf0b2fa
Vai Collegamenti
Homepage Ricerca English version Info fonti
La ricerca del componente "Potassio (mg/100g p.e.)" ha fornito 435 risultati.
Cliccare qui per un'altra ricerca.
Risultati espressi in mg per 100g di parte edibile:
ALIMENTO VALORE TRACCE FONTE
salsa tomato ketchup 5901 no INRAN - Banca dati di composizione degli alimenti
te' (foglie) 2160 no INRAN - Banca dati di composizione degli alimenti
caffe' tostato 2020 no INRAN - Banca dati di composizione degli alimenti
soia secca 1740 no INRAN - Banca dati di composizione degli alimenti
farina di soia 1660 no INRAN - Banca dati di composizione degli alimenti
latte di vacca in polvere, scremato 1650 no INRAN - Banca dati di composizione degli alimenti
melu' o pesce molo, stoccafisso secco 1500 no INRAN - Banca dati di composizione degli alimenti
fagioli -borlotti secchi crudi 1478 no INRAN - Banca dati di composizione degli alimenti
fagioli crudi 1445 no INRAN - Banca dati di composizione degli alimenti
Operazione completata
Internet
start
Database di composiz...
Adobe Photoshop
21.26

Si noti come, a partire da quest'ultimo elenco, si abbia immediato accesso ad altre importanti informazioni: cliccando sul nome dell'alimento si ottiene la scheda ad esso relativa, mentre premendo sull'icona blu di informazione si ottengono le informazioni aggiuntive sul valore.

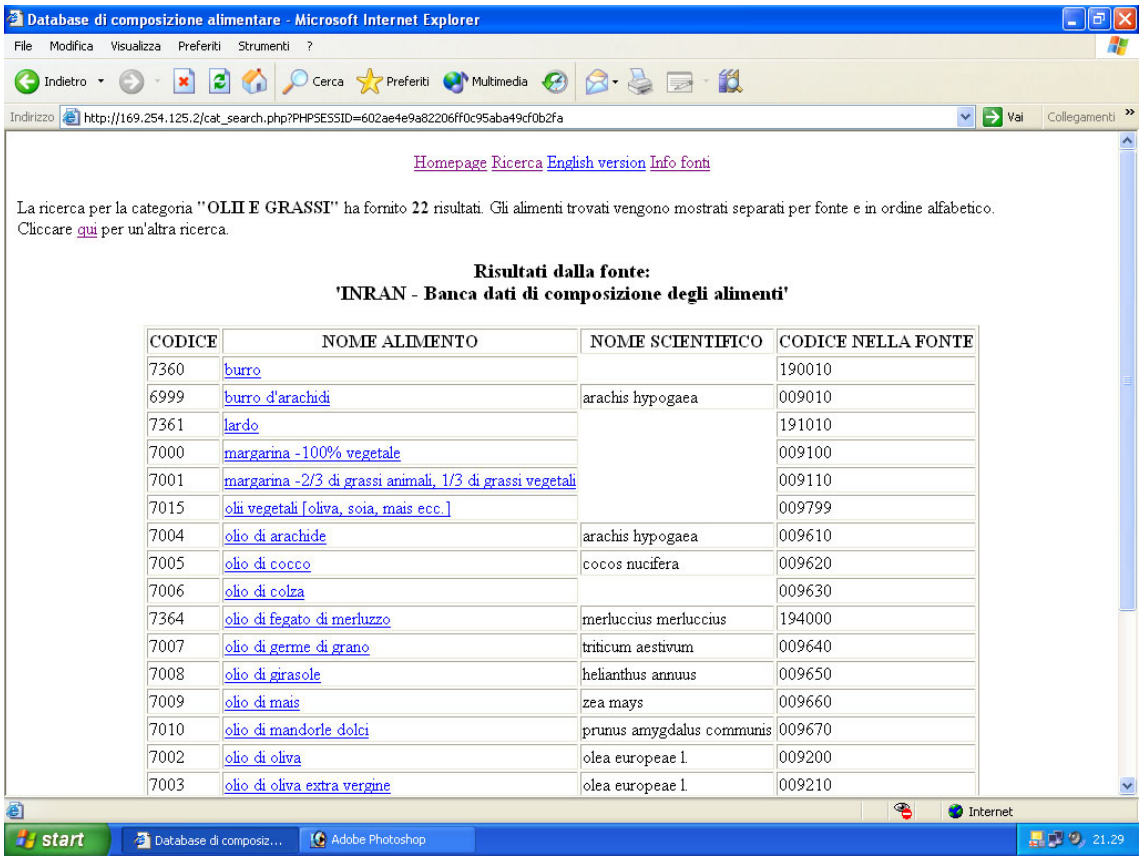
7.4.3 La Ricerca per Categoria

L'ultima, semplice ricerca prevista è quella per categoria. Scegliendo il nome della categoria da un elenco a scomparsa, l'utente ottiene, in ordine alfabetico e divisi per fonte, gli alimenti ad essa appartenenti.

In questa prima immagine vediamo il form da riempire per questo tipo di ricerca.



Premendo sul bottone “Cerca!” otteniamo un elenco molto simile a quello risultante dalla ricerca per alimento, come illustrato da questa ulteriore figura.



Anche da questa tabella è possibile accedere, come di consueto, alla scheda del singolo alimento: basta ciccare sul nome del cibo di cui si desidera ricevere maggiori informazioni.

CAPITOLO OTTO

CONCLUSIONI E SVILUPPI FUTURI

Raramente un progetto, in ambito informatico, viene dichiarato definitivamente concluso: questo capitolo è dedicato alle possibili strade che si potrebbero seguire per rendere ancora più valido il lavoro fin qui illustrato.

8.1 Osservazioni sul Lavoro Svolto

Leggendo quanto scritto finora ci si rende facilmente conto che tutti gli obiettivi iniziali del progetto, illustrati nel primo capitolo, sono stati raggiunti.

In particolare è stato migliorato il cosiddetto stato dell'arte: anche se il sito sviluppato è stato presentato come un prototipo, è sicuramente al momento tra i migliori che permettano di visualizzare informazioni in italiano, di lavorare contemporaneamente su più fonti, di visualizzare dati dettagliati sui singoli valori.

Naturalmente l'ultima parola spetta agli utilizzatori del software, da cui ci si aspetta un feedback adeguato, così da migliorare sempre il servizio.

Resta comunque ancora molto lavoro da svolgere, ma gran parte di questo lavoro può (e spesso deve) essere svolto anche da personale esperto nel settore nutrizionale e non in quello informatico.

8.2 Possibili Aggiunte ai Dati

Quando si pensa alle possibili espansioni del database, viene subito in mente la possibilità di inserire dati presi da ulteriori fonti: in questo caso non c'è che da seguire i passi illustrati nei capitoli da uno a sei (ricerca, selezione, reverse engineering, sviluppo dei moduli di feeding).

Inoltre anche le fonti già inserite forniscono spesso informazioni aggiuntive che, come è stato fatto notare di volta in volta, sono facilmente integrabili con i dati già presenti: l'unico difetto che hanno queste informazioni è quello di non poter essere quasi mai inseribili in maniera automatica.

E' comunque molto interessante anche la possibilità di tradurre dall'inglese all'italiano (e viceversa) tutto il materiale già presente nel database.

8.3 Possibili Aggiunte al Sito

Come già detto, il sito è un prototipo, cui innanzitutto occorre aggiungere la versione inglese e un'interfaccia più accattivante.

Si tenga presente inoltre che il software al momento fa una semplice consultazione dei dati presenti nel database. Non è difficile immaginare applicazioni più complicate che agiscono sui dati: si possono compiere calcoli statistici su più alimenti, calcolare il contenuto di nutrienti di un'intera dieta, riscontrare in modo semiautomatico discrepanze tra i dati delle varie fonti, etc.

8.4 Costruzione di Ulteriore Software

Basandosi sugli stessi dati del sito, si può poi ovviamente costruire altro software: anche per questo il progetto è fortemente incentrato sui dati e non sull'applicazione.

Un primo esempio utile può essere quello di implementare un applicativo di consultazione (e non solo) per sistemi operativi Windows o per palmari: nel primo caso non dovrebbe per nulla essere difficile sfruttare il codice di feeding già scritto per riempire un semplice database Access, da distribuire insieme al programma. Si potrebbe addirittura sfruttare direttamente l'eseguibile sotto Linux per alimentare, tramite il software "ODBC-ODBC Bridge" di cui si è discusso nel capitolo sei, il già menzionato db risiedente su macchina Windows.

Anche in questo caso ci si aspetta che sia l'utenza, come accade quasi sempre per il software di successo, ad indicare la via da seguire.

APPENDICE A

MANUALE DEI MODULI DI FEEDING

Questa appendice è rivolta agli utilizzatori del software di feeding, quindi al Data Base Administrator. Più che di un manuale si tratta sostanzialmente di un breve tutorial, che ripercorre i passi svolti nella creazione e alimentazione del database.

A.1 Generazione del Database

Il primo passo da seguire, dopo aver installato il software freeware allegato, è quello di creare la struttura del database.

Per fare ciò si può partire dallo schema logico del capitolo cinque e far generare al CASE uno script di creazione della base di dati per il particolare DBMS.

Il software di feeding è stato pensato in modo da essere indipendente dal DBMS, anche se è stato testato con il solo PostgreSQL: non ci dovrebbero comunque essere particolari problemi qualora si decidesse di utilizzare un sistema di gestione dei dati diverso, in ambiente Linux o Windows.

A.2 Inserimento dei Dati di Inizializzazione

Esistono ovviamente dei primi dati che non sono presenti nelle fonti, ma che devono essere inseriti obbligatoriamente dal DBA prima di qualsiasi operazione automatica, pena il fallimento della stessa.

Esempi di queste informazioni sono quelle riguardanti le tabelle “Fonte primaria”, “Componente standard”, “Categoria standard”, etc.. Per inserire questi dati sono stati creati degli script che, contenendo dei semplici “insert”, dovrebbero poter essere utilizzati per qualsiasi DBMS.

A.3 Preparazione dei Dati delle Fonti

Per quanto riguarda le fonti selezionate, esistono alcuni imperfezioni nei dati iniziali, che è bene correggere prima di inserire le stesse nel database: dopo aver effettuato tali correzioni, conviene tenere da parte la versione modificata della fonte per eventuali reinserimenti.

Per quanto riguarda la fonte INRAN, bisogna solo modificare i dati in accordo all'errata corrige presente sul rispettivo sito. Queste modifiche possono naturalmente essere fatte anche dopo il feeding.

La fonte USDA ha invece un problema più grave che, se non risolto, non permette l'inserimento automatico della stessa. Infatti i dati iniziali contengono in sé una violazione di un vincolo di integrità referenziale: tale errore della fonte viene prontamente segnalato in fase di inserimento automatico, producendo il blocco dell'operazione. La violazione riguarda la presenza, nella tabella "data_src_link", di codici assenti in realtà in "data_src": i codici incriminati sono "S995", "S1108" e "S1762", e vanno ovviamente eliminati prima dell'inserimento.

Un problema di minor rilievo è presente nella fonte IEO, dove semplicemente è presente un nome scientifico che, a differenza di tutti gli altri, non è presente tra parentesi quadre: l'alimento in questione è quello di codice 401 e ovviamente tale problema non blocca l'inserimento, ma semplicemente non fa riconoscere al programma il nome scientifico come tale.

Una ulteriore modifica da fare ai dati riguarda i file Access. Se si è intenzionati ad utilizzare i driver ODBC forniti da MDB Tools (magari in combinazione con l'applicazione Data Transfer) bisogna ricordare che esistono molte limitazioni sugli identificatori: non possono essere presenti nomi di tabelle o di colonne più lunghi di 15 caratteri e non possono essere utilizzati molti caratteri, tra cui "_", ":" e "-".

Sono state create comunque delle versioni delle fonti con tutte le modifiche necessarie già apportate: tali versioni sono come sempre allegate alla tesi.

A.4 Utilizzo di Data Transfer

Questa applicazione può essere utilizzata in sostituzione all'utilizzo dell'ODBC-ODBC Bridge. Il software semplicemente trasferisce i dati delle fonti INRAN e USDA da un DBMS all'altro, utilizzando ODBC. Bisogna innanzitutto creare i due database (sorgente e destinazione) per ogni fonte, magari utilizzando gli script generati da PowerDesigner, partendo dagli schemi non ristrutturati.

Si può poi richiamare l'eseguibile (es. "./datatransfer"), che chiederà al DBA innanzitutto quale schema si vuole replicare (INRAN o USDA) e in seguito si farà fornire DSN, User Name e Password delle fonti di dati ODBC corrispondenti a sorgente e destinazione. L'utente può decidere di inserire il carattere "n" (maiuscolo o

minuscolo) qualora non volesse inserire User Name e Password, che possono essere state già inserite nella definizione della fonte di dati ODBC.

L'applicazione dà comunque un errore di connessione fallita se i dati inseriti non sono corretti.

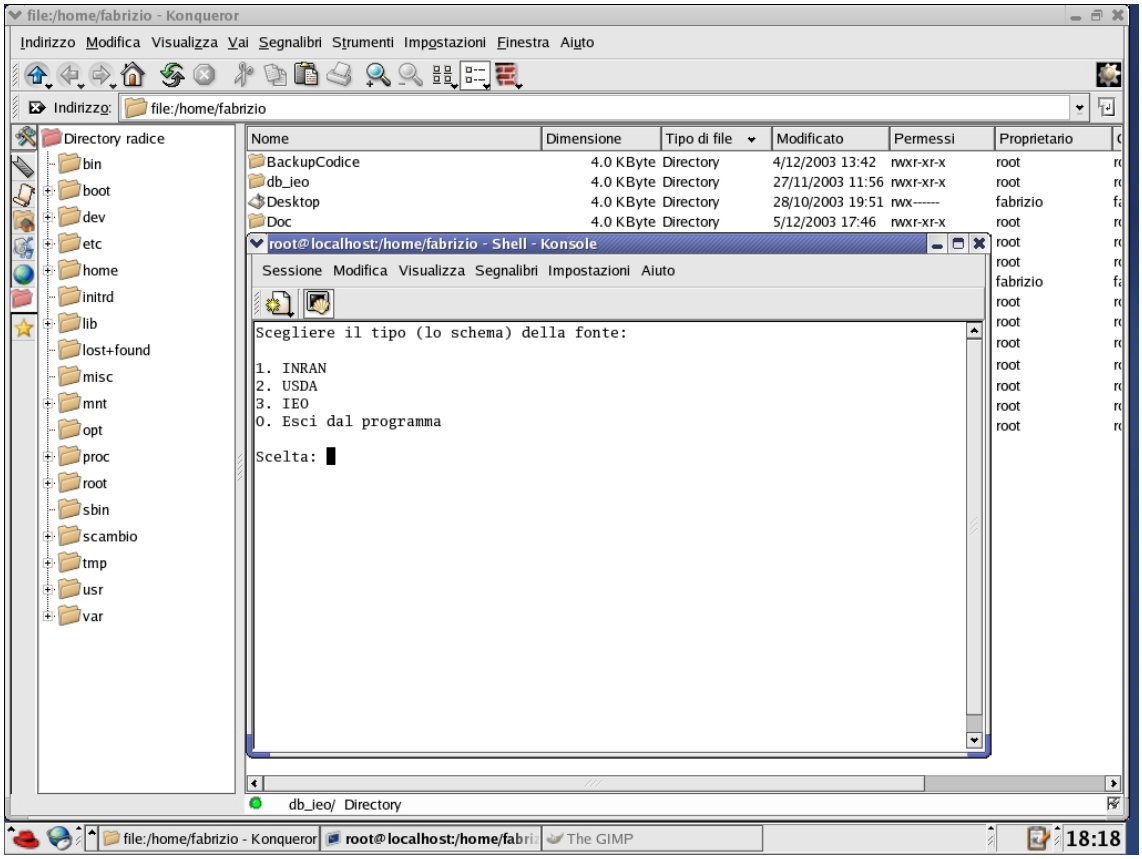
Si consiglia ovviamente di aggiornare le statistiche del planner dopo aver riempito i nuovi database.

A.5 Utilizzo di Feeder

Questo software esegue operazioni automatiche sul database di integrazione. Il programma permette inserimento, aggiornamento o eliminazione dei dati di una singola fonte. Prima di qualsiasi operazione è bene comunque creare un backup della base di dati (in PostgreSQL si può fare con le utility pg_dump e pg_restore). In particolar modo il backup è consigliato prima delle cancellazioni, in quanto queste portano all'eliminazione anche di tutti i dati relativi alla fonte selezionata inseriti manualmente in un secondo momento. A proposito di tali dati, si consiglia sempre l'inserimento tramite script, così da rendere sempre ripetibili le operazioni effettuate.

Anche questo applicativo richiede innanzitutto lo schema (INRAN, USDA o IEO) della fonte. Viene poi inserita dal DBA l'operazione da effettuare e, dopo una conferma in caso di eliminazione, vengono richiesti i dati di accesso: i soliti DSN, User Name e Password per le fonti ODBC, il percorso nel caso di file ASCII.

La figura seguente mostra la prima schermata del programma, così da dare un'idea dell'interfaccia testuale realizzata.



A.6 Inserimento dei Dati di Collegamento

Dopo aver inserito i dati delle varie fonti, vanno immesse anche le informazioni di collegamento tra le fonti e le entità “standard”. Per far ciò sono stati creati degli ulteriori script, che stavolta si devono eseguire esclusivamente dopo gli inserimenti automatici, magari in un’unica transazione e dopo il solito backup. Anche questi script vengono forniti il allegato.

A.7 Creazione della Data Warehouse

Se si desidera poi utilizzare i dati per la consultazione tramite il sito web descritto nel capitolo sette, devono venir generati gli schemi e le viste documentati nello stesso

capitolo. Al momento le viste non includono dei trigger che le aggiornino automaticamente: la datawarehouse va dunque ricreata periodicamente per mantenerla aggiornata con i dati nel database di integrazione, operazione questa che comunque non richiede eccessivo dispendio di tempo. Come sempre, si faccia riferimento allo script allegato per l'effettuazione di questa procedura.

A.8 Manutenzione del Database

Sia il database che la datawarehouse necessitano di poche, sporadiche operazioni di mantenimento da parte del Data Base Administrator.

Vanno innanzitutto aggiornate periodicamente le statistiche del planner, cosa possibile in PostgreSQL col comando "VACUUM ANALYZE" (vedi [13]). Questa procedura è importante anche perché libera lo spazio di eventuali tuple eliminate e perché evita il transaction ID wraparound.

Altra operazione che potrebbe rendersi necessaria per migliorare le prestazioni del sistema è la ricostruzione degli indici, possibile in PostgreSQL tramite "REINDEX nome_tabella". Questa operazione è da eseguirsi soprattutto sulle tabelle in cui siano state effettuate molte cancellazioni e inserimenti.

Tutte le operazioni descritte vanno comunque effettuate ad intervalli molto ampi di tempo e, naturalmente, l'aggiornamento delle statistiche va attuato almeno una volta dopo ogni modifica sostanziale ai dati.

RINGRAZIAMENTI

Quando penso a chi ringraziare per questa piccola fatica che avete tra le mani, il primo pensiero va ovviamente al prof. d'Acierno: un amico. Caro professore, rimpiango solo di aver avuto poco tempo per discorrere con lei di cose diverse dal "lavoro", perché è raro, è lei lo sa, imbattersi in persone del suo calibro: non cambiate mai, almeno non in peggio!

Ringrazio inoltre ovviamente i miei genitori, perché mi hanno dato la possibilità, sicuramente non da poco, di laurearmi (e soprattutto perché hanno preso poche precauzioni in quella notte di luglio di venticinque anni fa!).

Un ultimo sentito "grazie" lo voglio dire, anche a costo di scadere nella retorica, a tutte le persone che mi hanno dato affetto e amicizia sincera: senza di loro neanche una parola scritta in questa tesi avrebbe senso.