

## SQL: Concetti Base -Prima Parte-

Atzeni, Ceri, Paraboschi, Torlone  
Basi Di Dati:  
Modelli e Linguaggi di Interrogazione,  
McGraw-Hill Italia  
Capitolo 4

### SQL

---

- Structured Query Language
- Contiene:
  - DDL (Data Definition Language)
  - DML (Data Manipulation Language)
  - DCL (Data Control Language).
- I sistemi commerciali spesso offrono una serie di strumenti che estendono le funzionalità definite a livello di standard.

SQL

1

### Lo standard SQL

---

- SQL Base
  - SQL-86 - costrutti base
  - SQL-89 – integrità referenziale
- SQL-2
  - SQL- 92: entry level, intermediate level, full
- SQL-3
  - SQL-99 – estensione ad oggetti, trigger, ..
  - SL-2003 – Estensione del modello a oggetti, SQL/XML, ....

SQL

2

### Livelli di Compatibilità

---

- I sistemi commerciali sono classificati in base all'aderenza allo standard 92
  - Entry Level (SQL-89)
  - Intermediate Level
  - Full SQL (SQL-92)

SQL

3

## Domini Predefiniti

- Carattere
  - permette di rappresentare singoli caratteri o stringhe. Le stringhe possono essere a lunghezza fissa oppure variabile.
  - character [varying] [(lunghezza)] [character set *famiglia*]
- Tipi numerici esatti: permette di rappresentare valori interi o valori in virgola fissa.
  - Numeric [(Precisione[,Scala])]
  - Decimal [(Precisione[,Scala])]
  - integer
  - smallint

SQL

4

## Domini Predefiniti (continua)

- I domini integer e smallint si usano quanto non è necessario avere una rappresentazione della parte frazionaria.
- L' unico vincolo implementativo è che la precisione del dominio integer sia  $\geq$  del dominio smallint.
- Il dominio numeric rappresenta numeri in base decimale;
  - Precisione indica il numero di cifre
  - Scala il numero di cifre dopo la virgola.  
numeric(6,3) [-999.999,999.999]
  - La differenza tra numeric e decimal è che la precisione per numeric è un requisito esatto, mentre per decimal è un requisito minimo.

SQL

5

## Domini Predefiniti (continua)

- Tipi numerici approssimati:
  - float [(Precisione)]
  - double precision
  - real
  - Nel dominio float si può scegliere la precisione, intesa come numero di cifre della mantissa.
  - La precisione di double è doppia rispetto a real.

SQL

6

## Domini Predefiniti (continua)

- Data e Ora:
  - date record of {year month day}
  - time record of {hour minute second}
  - timestamp date+time
- Intervalli temporali:
  - interval PrimaUnita'diTempo [to UltimaUnita'diTempo]
  - interval year to month
  - indica che gli intervalli vanno misurati in numero di anni e di mesi.

SQL

7

## Domini Elementari (continua)

- SQL-99 ha introdotto
  - Boolean
    - precedentemente realizzato tramite il tipo bit (varbit), non implementato da molti sistemi
  - Blob (binary long object)
  - Cblob (character long object)

SQL

8

## DDL

- SQL consente la definizione di uno schema di base di dati i cui elementi sono i domini, le tabelle, i vincoli, gli indici, le asserzioni, le viste e i privilegi, attraverso la seguente sintassi:  
create schema [nome]  
[[authorization] Autorizzazione]  
{ElementiDelloSchema}

SQL

9

## Creazione di Tabelle

```
create table nome
(
  NAttr Dom [ValoreDef] [vincoli]
  {,NAttr Dom [ValoreDef] [vincoli]}
  [Altri Vincoli]
)

create table Dip (
  Nome varchar(20) primary key,
  Città varchar(20)
)
```

SQL

10

## Creazione di Tabelle (continua)

- Gli attributi possono appartenere ai domini predefiniti o a domini creati con la sintassi  
create domain ndom as TipodiDato [ValoreDiDefault] [Vincolo]
  - Nella definizione delle tabelle e di domini esistono i valori di Default che rappresentano il valore assunto dall' attributo quando viene inserita una riga nella tabella senza specificare il valore per l' attributo stesso.  
default {valore | user | null}
- Figli smallint default 0

SQL

11

## Vincoli Intrarelazionali

- Not Null: indica che l'attributo deve essere specificato, a meno che ad esso non sia associato un valore di default diverso da null.
- Unique: si applica ad un attributo per dire che nella tabella non possono esistere due righe con valori identici su tale attributo. Matricola char(6) unique
  - Può anche essere applicato a più attributi:  
cognome varchar(20),  
nome varchar(20),  
unique(cognome,nome)  
per indicare che non devono esistere righe che abbiano uguali sia il nome sia il cognome.

SQL

12

## Vincoli Intrarelazionali (continua)

- Primary key: può essere indicato una sola volta per tabella.
- Mediante esso si specifica l'attributo (o l'insieme di attributi) che rappresentano la chiave primaria della tabella.
- E' usato per sottolineare l'attributo (o l'insieme di attributi) più usato per accedere alle righe di una tabella.

```
Create table impiegato (  
  Nome      varchar(20),  
  Cognome   varchar(20),  
  Dipart    varchar(20),  
  primary key (Cognome, Nome)  
)
```

SQL

13

## Vincoli Interrelazionali

- Tratteremo i vincoli di integrità referenziale (detti anche vincoli di riferimento).
- In SQL esiste il vincolo foreign key. Esso crea un legame tra l'attributo della tabella corrente (interna) e un attributo di un'altra tabella (esterna) (tipicamente alla chiave di questa).
- L'unico requisito è che l'attributo della tabella esterna sia definito unique.
- Tale vincolo può essere specificato in due modi:
  - references
  - foreign key

SQL

14

## Vincoli Interrelazionali (continua)

- references: è usato quando si vuole creare il legame con riferimento ad un singolo attributo:  
create table Impiegato (  
 Nome varchar(20),  
 Cognome varchar(20),  
 Dipart varchar(20) references Dipartimento(NomeDip)  
)
- foreign key: è usato quando ci si riferisce ad insieme di attributi:  
create table Impiegato (  
 Nome varchar(20),  
 Cognome varchar(20),  
 Dipart varchar(20) references Dipartimento(NomeDip),  
 foreign key (Nome, Cognome) references Anagrafica (nonan,cognan)  
)

SQL

15

## Vincoli Interrelazionali (continua)

- Violazioni di un vincolo foreign key.

```
create table Impiegato (  
    Nome varchar(20),  
    Cognome varchar(20),  
    Dipart varchar(20) references Dipartimento(NomeDip)  
)
```

- Impiegato è la tabella interna.
- Dipartimento è la tabella interna.
- La violazione può essere a livello di tabella
  - interna (Impiegato)
  - esterna (Dipartimento)

SQL

16

## Vincoli Interrelazionali (continua)

- Violazione sulla tabella interna.
  - Il vincolo può essere violato usando (quando si inserisce una nuova riga o quando si modifica una riga esistente) per il campo Dipart un valore non presente nella colonna NomeDip della tabella Dipartimento.
  - L'operazione viene semplicemente impedita.

SQL

17

## Vincoli Interrelazionali (continua)

- Consideriamo la tabella esterna.  
Le operazioni che possono dare problemi sono:

- Modifiche dell'attributo riferito
- Cancellazione di righe

- Politiche possibili

- cascade
- set null
- set default
- no action

```
create table Impiegato (  
    Nome char(20),  
    Cognome char(20),  
    Dipart char(20),  
    foreign key (Dipart)  
    references  
    Dipartimento(NomeDip)  
    on delete set null  
    on update cascade  
)
```

SQL

18

## Modifica degli Schemi

- alter.

```
alter domain nome <set default ..  
    drop default  
    add constraint drop constraint >  
alter table nome <  
    alter column <set default ..  
    drop default>  
    add constraint  
    drop constraint  
    add column  
    drop column>  
alter table Dipartimento add column NroUffici integer(4)
```

SQL

19

## Modifica degli Schemi (continua)

- drop.  
Permette di rimuovere schemi, tabelle etc.  
  
drop <schema | domain | table | view | assertion | >  
[restrict | cascade]
- restrict: specifica che il comando deve essere eseguito solo in presenza di oggetti vuoti.
- cascade: tutti gli oggetti specificati sono rimossi, innestando una reazione a catena.  
Data la pericolosità del comando, molti sistemi permettono di analizzare il risultato di un drop cascade prima di eseguirlo effettivamente.

SQL

20

## Indici

- Gli indici rappresentano uno strumento prezioso per velocizzare l'accesso ai dati.  
  
create [unique] index NomeIdx on NomeTable (ListaAttributi)
- unique indica che nella tabella non sono possibili righe che assumono lo stesso valore su ListaAttributi (che quindi è una chiave eventualmente non minimale)
- L'ordine in cui sono dati gli attributi è importante.  
create unique index ind onto Anagrafica (Cognome, Nome)  
drop index NomeIdx

SQL

21

## Cataloghi Relazionali

- Si supponga di avere creato uno schema e di aver creato tutta una serie di oggetti. Come faccio a vedere tutti gli oggetti definiti?
- Ogni DBMS mantiene alcune tabelle in cui sono riportate le descrizioni degli oggetti creati (ad es. le tabelle create).
- Esiste un Definition\_Schema ed una serie di viste su di esso (TABLES, COLUMNS, DOMAINS, .....)

SQL

22

## Interrogazioni

- SQL esprime le interrogazioni in modo dichiarativo attraverso il comando select.
- L'interrogazione è poi automaticamente tradotta in termini procedurali e poi eseguita.
- L'alternativa è costituita dall'uso di linguaggi tradizionali, in cui SQL è immerso.
- Esistono vari modi per formulare una stessa query.
- L'utente, tipicamente, non ha bisogno di preoccuparsi dell'efficienza della query formulata quanto della sua leggibilità e modificabilità.

SQL

23

## Interrogazioni (continua)

- Esempio:  
Impiegato (Nome, Cognome, Stipendio, Dipart, Ufficio, Stipendio)  
Dipartimento (Nome, Indirizzo, Città)  

```
select Stipendio as Salario  
from Impiegato  
where Cognome='Rossi'
```
- Ritorna una tabella con un unico attributo (Salario) i cui valori sono ottenuti dalle righe di Impiegato aventi Cognome = 'Rossi'.  

```
select *  
from Impiegato  
where Cognome='Rossi'
```

SQL

24

## Interrogazioni (continua)

- ```
select Stipendio/12 as StipMens  
from Impiegato  
where cognome='Rossi'
```
- ```
select Impiegato.nome, Dipartimento.nome, Citta'  
from Impiegato, Dipartimento  
where Impiegato.Dipart = Dipartimento.Nome
```
- Le righe vengono estratte dal prodotto cartesiano.
  - In maniera più compatta:  

```
select I.nome, D.nome, D.Citta'  
from Impiegato as I, Dipartimento as D  
where I.Dipart = D.Nome
```

SQL

25

## Interrogazioni (continua)

- Nella clausola where e' possibile usare connettivi logici (and, or, not) ed espressioni di confronto.  

```
select nome, cognome  
from Impiegato  
where ufficio=20 and dipartimento='DIS'
```

```
select nome, cognome  
from Impiegato  
where ufficio=20 or dipartimento='DIS'
```

```
select nome  
from Impiegato  
where cognome = 'Rossi' and (ufficio=20 or dipartimento='DIS')
```

SQL

26

## Operatore like

- E' usato nel confronto di stringhe e si appoggia su due caratteri speciali:
    - \_ (indica un qualsiasi carattere)
    - % (indica una stringa, anche vuota)
- ```
select nome  
from Impiegato  
where cognome like '_o%i'
```
- Ritornerà tutti nomi di Impiegato il cui cognome ha una "o" in seconda posizione e termina per "i" (Rossi, Doncelli, Loi).

SQL

27

## Valori Nulli

- SQL-89
- Esempio  
Stipendio > 40  
è soddisfatta dalle righe in cui Stipendio e' noto e maggiore di 40.
- Per i valori nulli si usa Stipendio is null

SQL

28

## Valori Nulli (continua)

- In SQL-2 ci si è orientati verso una logica a 3 valori.
- L' unico predicato che ritorna solo valore vero o falso è *is [not] null*
- Non ci sono particolari stravolgimenti, se non in interrogazioni molto complesse

| Not |   |
|-----|---|
| V   | F |
| F   | V |
| U   | U |

| O r | V | U | F |
|-----|---|---|---|
| V   | V | V | V |
| U   | V | U | U |
| F   | V | U | F |

| a n d | V | U | F |
|-------|---|---|---|
| V     | V | U | F |
| U     | U | U | F |
| F     | F | F | F |

SQL

29

## SQL vs Relazione come Insieme

- Abbiamo detto che il risultato del select è una relazione.
- Questo non è del tutto vero, in quanto tale risultato potrebbe avere righe uguali.
- Ciò è dovuto a ragioni di efficienza, perché eliminare i duplicati può essere costoso in termini di tempo e deve essere espressamente richiesto.

```
select distinct Città  
from ...
```

SQL

30

## Join

- Quando si formula una interrogazione su 2 o più tabelle, si agisce sul prodotto cartesiano.
- E' possibile specificare nella condizione where gli attributi leganti.
- La sintassi SQL-2 ha però introdotto un'altra possibilità per specificare questo operatore con diverse varianti

```
from t1 tipojoin join t2 on condizione  
con tipojoin
```

- inner, right (outer), left (outer), full (outer)

SQL

31

## Inner join

- Con il join interno le righe coinvolte nel risultato sono in genere un sottoinsieme delle righe di ciascuna tabella, perché una riga della tabella R1 potrebbe non avere corrispondenze nella tabella R2 e viceversa.
- In alcune applicazioni potrebbe servire mantenere tutte le righe di qualcuna delle (tutte le) tabelle coinvolte, mettendo dei valori Null per indicare l'assenza di informazioni provenienti dall'altra tabella.

SQL

32

## Outer join

- Outer join
  - left
  - right
  - full
- Il join esterno mantiene tutte le righe della tabella di sinistra (left), della tabella di destra (right) o di entrambe (full).

SQL

33

## Variabili (Alias)

- Esempio

```
select I.nome, D.nome, D.Citta'
from Impiegato as I, Dipartimento as D
where I.Dipart = D.Nome
```
- In questo caso la variabile è da interpretarsi semplicemente come un ridenominazione.
- Utilizzando gli Alias è però possibile anche far riferimento a più esemplari della stessa tabella.
- In questo caso l'interpretazione è che, per ogni alias, si introduce una variabile di tipo tabella che viene inizializzata con la tabella in questione.

SQL

34

## Esempio

- Esempio:

|               |      |      |     |       |
|---------------|------|------|-----|-------|
| Impiegati:    | Matr | Nome | Eta | Stip. |
| Supervisione: | Capo | Imp  |     |       |
- Trovare il nome e lo stipendio dei capi degli impiegati che guadagnano più di 40.

```
select I1.Nome, I1.Stip
from Impiegati (as) I1, Supervisione S, Impiegati I2
where I1.Matr=S.Capo and
I2.matr=S.Imp and
I2.Stp > 40
```

SQL

35

## Ordinamento

---

- Il risultato di una interrogazione è una relazione e come tale non è ordinata.
- Nelle applicazioni è comunque utile avere i risultati ordinati secondo una certa strategia.  
`select *  
from automobili  
order by marca desc, modello`
- L'ordinamento introduce, in molti casi, un overhead significativo.