Introduzione al linguaggio SQL

Giorgio Bar – giorgio.bar@to.infn.it

Scuola di Specializzazione in Fisica Medica – Università di Torino

Anno accademico 2023/2024 (anno solare 2024/2025)

Modifica dei dati in SQL

Modifica dei dati

- INSERT
  - Inserimento di nuove tuple in una tabella
- DELETE
  - Cancellazione di tuple da una tabella
- UPDATE
  - Aggiornamento degli attributi di una o più tuple

Istruzione INSERT

- Consente di inserire una singola tupla all'interno di una tabella

  INSERT INTO *NomeTabella* [*ListaAttributi*] VALUES (*ListaValori*) ;

- Omettere la lista degli attributi equivale a specificare tutti gli attributi, nell'ordine di creazione nella tabella (sconsigliato)
- Se non vengono specificati i valori di tutti gli attributi della tabella, agli attributi mancanti viene assegnato il valore di default o, in assenza di questo, il valore nullo

```
INSERT INTO FORNITORI ( CodF, NomeF, Rating, Sede )
       VALUES ('F1', 'Atlante', 2, 'Torino');
```

- Esiste anche una seconda forma della INSERT che consente di inserire un insieme di tuple, estratte dalla base di dati con una istruzione SELECT.

Istruzione DELETE

- Consente di cancellare dalla tabella tutte le tuple che soddisfano la condizione indicata

```
DELETE FROM NomeTabella [ WHERE Condizione ]  ;
```

- Se si omette la clausola WHERE vengono cancellate tutte le tuple dalla tabella (ma non la tabella).
- La condizione rispetta la sintassi della SELECT, al suo interno possono comparire interrogazioni nidificate che fanno riferimento ad altre tabelle.

Istruzione UPDATE

- Consente di aggiornare uno o più attributi delle tuple della tabella che soddisfano la condizione indicata

```
UPDATE NomeTabella
SET Attributo = Espressione { , Attributo = Espressione }
[ WHERE Condizione ]  ;
```

- Se si omette la clausola WHERE la modifica viene eseguita su tutte le tuple della tabella
- L'espressione può far riferimento al valore corrente dell'attributo che verrà modificato.

```
UPDATE FORNITORI SET Rating = 2*Rating WHERE Sede = 'Torino');
```

DDL (Data Definition Language)

Gestione delle tabelle e
vincoli di integrità

Creazione di una tabella

- Istruzione SQL DDL CREATE TABLE. Permette di
  - definire gli attributi (colonne) della tabella
  - definire vincoli di integrità sui dati della tabella

```
CREATE TABLE NomeTabella (
NomeAttributo Dominio [DEFAULT Valore] [Vincoli]
{ , NomeAttributo Dominio [DEFAULT Valore] [Vincoli] }
AltriVincoli
);
```

```
CREATE TABLE FORNITORI (
       CodF   VARCHAR(4) NOT NULL,
       NomeF  VARCHAR(30) NOT NULL,
       Rating INTEGER,
       Sede   VARCHAR(30),
       PRIMARY KEY ( CodF )
);
```

Creazione di una tabella

- *Dominio*
  - Definisce il tipo di dato dell'attributo
- *Vincoli*
  - Permette di specificare vincoli di integrità sull'attributo
- DEFAULT *Valore*
  - Permette di specificare il valore di default dell'attributo
  - DEFAULT <*costante* | NULL>
- *AltriVincoli*
  - Permette di specificare vincoli di integrità di tipo generale sulla tabella

Domini elementari (ANSI SQL)

- CHARACTER | CHAR [VARYING] [(*Lunghezza*)]
  - Stringhe di caratteri, anche di lunghezza variabile.

- NUMERIC | DECIMAL [(*Precisione, Scala*)]
  - Numerici esatti, in base decimale
  - *Precisione* indica il numero totale di cifre (digits)
  - *Scala* indica il numero di cifre dopo la virgola
  - 123.45 : la precisione è 5, la scala è 2
  - Per il dominio DECIMAL la precisione costituisce un requisito minimo, mentre per NUMERIC rappresenta un valore esatto.

Domini elementari (ANSI SQL)

- INTEGER | INT | SMALLINT
  - Numerici esatti, il numero di bit dipende dalla specifica implementazione di SQL.
- FLOAT [(*precisione*)]
  - Numerici approssimati, *precisione* indica il numero di bit utilizzati per memorizzare la mantissa del numero rappresentato in notazione scientifica.
- REAL, DOUBLE PRECISION
  - Numerici approssimati, valori a singola o doppia precisione. La precisione dipende dalla specifica implementazione di SQL.

https://docs.oracle.com/database/121/SQLRF/sql_elements001.htm

Domini elementari (ANSI SQL)

- DATE
  - Memorizza i valori che specificano anno, mese e giorno
- TIME
  - Memorizza i valori che specificano ora, minuti, secondi
- TIMESTAMP [(*precisione*)] [WITH TIME ZONE]
  - Memorizza i valori che specificano anno, mese, giorno, ora, minuti e secondi ed eventualmente la frazione di secondo
  - *precisione* indica il numero di cifre decimali utilizzate per memorizzare la frazione di secondo

Domini elementari

- BLOB | BINARY LARGE OBJECT (SQL:1999)
  - Memorizza qualsiasi dato che possa essere rappresentato come una sequenza di byte.

- CLOB | CHARACTER LARGE OBJECT (SQL:1999)
  - Memorizza sequenze di caratteri di elevate dimensioni (single-byte o multibyte character sets)

https://nils85.github.io/sql-compat-table/datatype.html

Creazione di una tabella

- Creazione della tabella fornitori

CodF	NomeF	Rating	Sede

```
CREATE TABLE FORNITORI (
    CodF   VARCHAR(4) NOT NULL,
    NomeF  VARCHAR(30) NOT NULL,
    Rating INTEGER,
    Sede   VARCHAR(30)
);
```

- Manca la definizione dei vincoli di integrità

Modifica della struttura

- Istruzione ALTER TABLE
  - Aggiunta di una nuova colonna

```
ALTER TABLE PRODOTTI ADD COLUMN TipoP INTEGER DEFAULT 42 NOT NULL;
```

  - Eliminazione di una colonna (attributo) esistente

```
ALTER TABLE PRODOTTI DROP COLUMN TipoP;
```

  - Definizione di un nuovo valore di default per una colonna (attributo)

```
ALTER TABLE PRODOTTI ALTER COLUMN Peso SET DEFAULT 0;
```

  - Definizione di un nuovo vincolo di integrità
  - Eliminazione di un vincolo di integrità

Cancellazione di una tabella

```
DROP TABLE NomeTabella [ RESTRICT | CASCADE ] ;
```

- Tutte le righe della tabella vengono eliminate insieme alla tabella
- RESTRICT (opzione di default)
  - La tabella non viene rimossa se è presente in qualche definizione di vincolo o vista
- CASCADE
  - Se la tabella compare in qualche definizione di vista, anche questa viene rimossa

https://dev.mysql.com/doc/refman/8.0/en/drop-table.html The RESTRICT and CASCADE keywords do nothing. They are permitted to make porting easier from other database systems.

Dizionario dei dati

- Il dizionario dei dati contiene i metadati (informazioni sui dati) di una base di dati relazionale:
  - Descrizione di tutte le strutture (tabelle, indici, viste) della base di dati
  - Stored procedure SQL
  - Privilegi degli utenti
  - Statistiche su tabelle, indici, viste e sulla crescita della base di dati
- Gestito direttamente dal DBMS relazionale, le informazioni sono memorizzate in tabelle della base dati (diverse per ogni DBMS).
- Può essere interrogato mediante istruzioni SQL

Integrità dei dati

- I dati all'interno di una base di dati sono corretti se soddisfano un insieme di regole dette vincoli di integrità
- Le operazioni di modifica dei dati definiscono un nuovo stato della base dati, non necessariamente corretto
- La verifica della correttezza della stato di una base di dati può essere effettuata
  - dalle procedure applicative, che effettuano tutte le verifiche necessarie
  - mediante la definizione di trigger
  - mediante la definizione di vincoli di integrità sulle tabelle

Procedure applicative

- Tutte le verifiche di correttezza necessarie sono previste all'interno di ogni applicazione
- Vantaggi
  - approccio efficiente
- Svantaggi
  - è possibile "aggirare" le verifiche interagendo direttamente con il DBMS
  - un errore di codifica può avere un effetto significativo sulla base di dati
  - la conoscenza delle regole di correttezza è "nascosta" nelle applicazioni

Trigger

- Procedure memorizzate nel dizionario dati del sistema, quando si verifica un evento di modifica dei dati sotto il controllo del trigger, la procedura viene eseguita automaticamente
- Vantaggi
  - permettono di definire vincoli di integrità di tipo complesso, vengono normalmente utilizzati insieme alla definizione di vincoli sulle tabelle
  - unico punto centralizzato di verifica
  - impossibilità di "aggirare" la verifica dei vincoli
- Svantaggi
  - applicativamente complessi
  - possono rallentare l'esecuzione delle applicazioni

Vincoli di integrità sulle tabelle

- Definiti nelle istruzioni CREATE o ALTER TABLE e memorizzati nel dizionario dati di sistema. Durante l'esecuzione di una qualsiasi operazione di modifica dei dati il DBMS verifica in modo automatico che i vincoli siano rispettati
- Vantaggi
  - definizione dichiarativa dei vincoli, la cui verifica è affidata al sistema
  - unico punto centralizzato di verifica
  - impossibilità di "aggirare" la verifica dei vincoli
- Svantaggi
  - possono rallentare l'esecuzione delle applicazioni
  - non è possibile definire tipologie arbitrarie di vincoli, ad esempio dei vincoli su dati aggregati

Vincoli di integrità

I vincoli di integrità possono essere specificati in modo dichiarativo, affidando al sistema la verifica della loro consistenza.

- Vincoli di tabella: restrizioni sui dati permessi nelle colonne di una tabella
- Vincoli di integrità referenziale: gestione dei riferimenti tra tabelle diverse
  - basati sul concetto di chiave esterna

https://www.postgresql.org/docs/9.3/ddl-constraints.html

Vincoli di tabella

- Sono definiti su una o più colonne di una tabella
- Sono verificati dopo ogni istruzione SQL che opera sulla tabella soggetta al vincolo (inserimento di nuovi dati o modifica del valore di colonne soggette al vincolo)
- Se il vincolo è violato, l'istruzione SQL che ha causato la violazione genera un errore di esecuzione
- Tipologie di vincolo
  - Chiave primaria
  - Ammissibilità del valore nullo
  - Unicità
  - Vincoli generali di tupla

Chiave primaria

- Insieme di attributi che identifica in modo univoco le righe di una tabella. Può essere specificata una sola chiave primaria per una tabella
- Definizione della chiave primaria

```
CREATE TABLE FORNITORI (
    CodF   VARCHAR(4) PRIMARY KEY,
    NomeF  VARCHAR(30),
    Rating INTEGER,
    Sede   VARCHAR(30)
);
CREATE TABLE ORDINI (
    CodF   VARCHAR(4),
    CodP   VARCHAR(6),
    Qta    INTEGER,
    PRIMARY KEY ( CodF, CodP )
);
```

Se composta da un solo attributo

Se composta da uno o più attributi

Ammissibilità del valore nullo

- Il valore NULL indica l'assenza di informazioni
- Il vincolo NOT NULL indica che è obbligatorio specificare sempre un valore per l'attributo

NomeAttributo Dominio NOT NULL

```
CREATE TABLE FORNITORI (
    CodF   VARCHAR(4) NOT NULL,
    NomeF  VARCHAR(30) NOT NULL,
    Rating INTEGER,
    Sede   VARCHAR(30),
    PRIMARY KEY ( CodF )
);
```

Unicità

- Vincolo UNIQUE. Un attributo o un insieme di attributi non può assumere lo stesso valore in righe diverse della tabella
- Ma è ammessa la ripetizione del valore NULL (considerato sempre diverso)
- Per un solo attributo

```
NomeAttributo Dominio UNIQUE
```

- Per uno o più attributi

```
UNIQUE ( ElencoAttributi )
```

Vincoli generali di tupla

- Permettono di esprimere condizioni di tipo generale su ogni tupla

  NomeAttributo Dominio CHECK *(Condizione)*

- Possono essere indicati come condizione i predicati specificabili nella clausola WHERE, la base dati è corretta se la condizione è vera

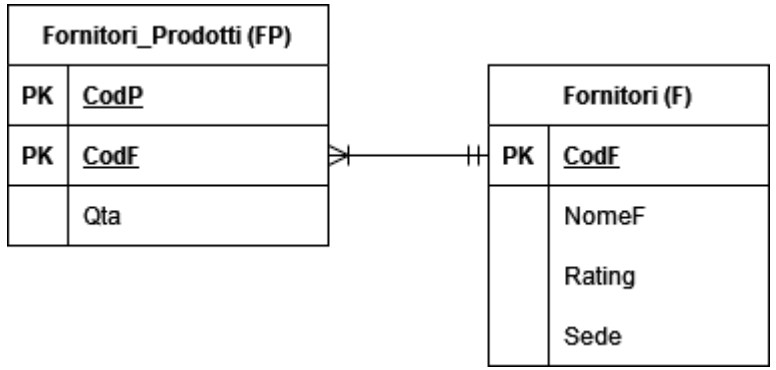
```
CREATE TABLE FORNITORI (
    CodF    VARCHAR(4) NOT NULL,
    NomeF   VARCHAR(30) NOT NULL UNIQUE,
    Rating  INTEGER CHECK ( Rating > 0 ),
    Sede    VARCHAR(30),
    PRIMARY KEY ( CodF )
);
```

Vincoli di integrità referenziale

- Permettono di gestire il legame tra tabelle mediante il valore degli attributi

FORNITORI (codF, NomeF, Rating, Sede)

ORDINI (codF, codP, Qta)



- La colonna CodF di ORDINI può assumere solo valori presenti nella colonna CodF di FORNITORI
  - CodF in ORDINI: colonna referenziante (o chiave esterna)
  - CodF in FORNITORI: colonna referenziata (tipicamente una chiave primaria)

https://www.postgresql.org/docs/9.3/ddl-constraints.html

Definizione della chiave esterna

- La chiave esterna è definita nell'istruzione CREATE TABLE della tabella referenziante

```
FOREIGN KEY ( ElencoAttributiReferenzianti )
REFERENCES NomeTabella [ ( ElencoAttributiReferenziati ) ]
```

- Se gli attributi referenziati hanno lo stesso nome di quelli referenzianti, non è obbligatorio specificarli

```
CREATE TABLE ORDINI (
    CodF   VARCHAR(4),
    CodP   VARCHAR(6),
    Qta    INTEGER,
    PRIMARY KEY ( CodF, CodP ),
    FOREIGN KEY (CodF) REFERENCES FORNITORI(CodF),
    FOREIGN KEY (CodP) REFERENCES PRODOTTI(CodP)
);
```

Politiche di gestione dei vincoli

- I vincoli di integrità sono verificati dopo ogni istruzione SQL che potrebbe causarne la violazione
- Non sono ammesse operazioni di inserimento e modifica sulla tabella referenziante che violino il vincolo, vale a dire attributi referenzianti con valori non presenti nella tabella referenziata

Politiche di gestione dei vincoli

```
FOREIGN KEY ( ElencoAttributiReferenzianti )
REFERENCES NomeTabella [ ( ElencoAttributiReferenziati ) ]
[ ON UPDATE < CASCADE | SET DEFAULT | SET NULL | NO ACTION > ]
[ ON DELETE < CASCADE | SET DEFAULT | SET NULL | NO ACTION > ]
```

- Operazioni di modifica o cancellazione sulla tabella referenziata causano sulla tabella referenziante:
  - CASCADE: propagazione dell'operazione di aggiornamento o cancellazione
  - SET NULL / SET DEFAULT: null o valore di default negli attributi referenzianti delle tuple che hanno valori non più presenti nella tabella referenziata
  - NO ACTION: non si esegue l'azione invalidante (opzione di default)

Politiche di gestione dei vincoli

```
CREATE TABLE ORDINI (
    CodF   VARCHAR(4),
    CodP   VARCHAR(6),
    Qta    INTEGER CHECK (Qta IS NOT NULL AND Qta >0),
    PRIMARY KEY ( CodF, CodP ),
    FOREIGN KEY (CodF) REFERENCES FORNITORI(CodF)
            ON DELETE NO ACTION
            ON UPDATE CASCADE,
    FOREIGN KEY (CodP) REFERENCES PRODOTTI(CodP)
            ON DELETE NO ACTION
            ON UPDATE CASCADE
);
```

Per approfondire

- Paolo Atzeni, Stefano Ceri, Piero Fraternali
  Basi di dati. Modelli e linguaggi di interrogazione
  McGraw-Hill Education
  ISBN: 8838668000
- Database SQL Language Reference
  https://docs.oracle.com/database/121/SQLRF/
- PostgreSQL 13.2 Documentation
  https://www.postgresql.org/docs/13/index.html

Estensioni SQL per interrogazioni OLAP

- Nuove funzioni aggregate caratterizzate da:
  - finestra di calcolo, all'interno della quale specificare funzioni aggregate per la definizione di totali parziali e cumulativi e il calcolo della media mobile
  - possibilità di ricavare la posizione nell'ordinamento (ranking)
- Operatori per il calcolo di più raggruppamenti (GROUP BY) diversi nello stesso momento

https://www.oracle.com/database/technologies/olap.html

Estensioni SQL per interrogazioni OLAP

Finestra di calcolo caratterizzata da:

- partizionamento: divide le righe in gruppi, ma senza collassarle (a differenza della GROUP BY)
- ordinamento delle righe, separatamente all'interno di ogni partizione
- finestra di aggregazione: definisce il gruppo di righe su cui calcolare l'aggregato, per ciascuna riga della partizione.

```
SELECT Nazione, Mese, Importo,
       AVG(Importo) OVER (PARTITION BY Nazione
                          ORDER BY Mese
                          ROWS 2 PRECEDING) AS MediaMobile
FROM Vendite;
```

Basi di dati spaziali

- Oracle's spatial database
  https://www.oracle.com/database/spatial/
- PostGIS spatial database extender for PostgreSQL
  https://postgis.net/

```
WITH city AS (
  SELECT 'Gotham' AS name,
         ST_Buffer(ST_Point(0,0), 10) AS geom
)
 , superhero(name,geom) AS (
    VALUES
      ('Bat Boy', ST_Point(0.1,0))
    , ('Bat Girl', ST_Point(1,1) )
)
SELECT superhero.name
FROM city INNER JOIN superhero
        ON ST_Contains(city.geom, superhero.geom);
```