Laboratorio d'Informatica

Corso di Laurea Magistrale in Ingegneria per l'Ambiente e il Territorio

A.A. 2021/2022

Docente: Lorenzo Putzu

Lezione 12

Basi Di Dati

Vincolo d'integrità referenziale

- Le modifiche a una tabella **referenziata** che violerebbero il vincolo di integrità referenziale (**cancellazione** di righe o **modifica** dei valori della chiave primaria) vengono impedite da un DBMS, a meno che non si specifichi un'azione diversa nella definizione dello stesso vincolo, attraverso le clausole:
  - ON DELETE (azioni da eseguire in caso di **cancellazione**)
  - ON UPDATE (azioni da eseguire in caso di **modifica**)

Vincolo d'integrità referenziale

- Le azioni principali disponibili in SQL sono:
  - SET NULL: si assegna NULL alla chiave esterna in corrispondenza delle righe cancellate o modificate nella tabella referenziata (è possibile solo se la chiave esterna non fa parte anche della chiave primaria della tabella referenziante)
  - SET DEFAULT: si assegna il valore di *default* alla chiave esterna (se non si è specificato tale valore nel comando CREATE TABLE, si assegna NULL)
  - CASCADE: la modifica alla tabella referenziata viene **propagata** a quella referenziante: nel caso ON DELETE si eliminano le righe corrispondenti nella tabella referenziante; nel caso ON UPDATE si aggiorna il valore della chiave esterna
  - NO ACTION: la modifica viene impedita (azione predefinita, non è necessario specificarla esplicitamente)

Vincolo d'integrità referenziale

- Sintassi completa della clausola FOREIGN KEY:

```
...
FOREIGN KEY ("nome colonna", ...)
  REFERENCES "tabella referenziata" ("nome colonna", ...) ,
  [ ON DELETE CASCADE / SET NULL / SET DEFAULT /
    NO ACTION , ]
  [ ON UPDATE CASCADE / SET NULL / SET DEFAULT /
    NO ACTION , ]
...
```

Esempio

```
CREATE TABLE "Esami" (
        ...
        FOREIGN KEY ("Studente")
                        REFERENCES "Studenti" ("Matricola"),
                        ON UPDATE CASCADE  ON DELETE CASCADE,

        FOREIGN KEY ("Insegnamento")
                        REFERENCES "Insegnamenti" ("Codice")
                        ON UPDATE CASCADE  ON DELETE CASCADE ,

        ...
);
```

Esercizio (1/3)

- Si consideri una BD, di nome "Circolo nautico", che debba contenere alcuni dati sui velisti soci di un circolo nautico, sulle barche gestite dal circolo e sulle loro prenotazioni da parte dei soci.
- Una barca può essere prenotata da un solo socio alla volta, e la prenotazione dura un intero giorno.
- Ciascun socio e ciascuna barca sono identificati univocamente da un codice denominato rispettivamente "vid" e "bid" (in entrambi i casi, un numero intero).
- L'esperienza nautica di ciascun socio è rappresentata con un valore da 1 (minima) a 10 (massima).

Esercizio (2/3)

- Un esempio di istanza della BD (i vincoli d'integrità referenziale sono indicati dalle frecce)

Velisti

vid	*vnome*	*esperienza*	*età*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

Prenotazioni

vid	*bid*	*giorno*
22	101	10/10/96
58	103	11/12/96

Barche

bid	*bnome*	*colore*
101	Interlake	blu
102	Interlake	rosso
103	Clipper	verde
104	Marine	rosso

Esercizio (3/3)

- Schema logico:
  - **Velisti** (*vid*: intero, *vnome*: stringa(20), *esperienza*: int., *età*: reale)
  - **Barche** (*bid*: intero, *bnome*: stringa(25), *colore*: stringa(10))
  - **Prenotazioni** (*vid*: intero, *bid*: intero, *giorno*: data)
  - Vincoli d'integrità referenziale:
  - *vid*(Prenotazioni) → *vid*(Velisti); tutte le modifiche alla tabella Velisti devono essere propagate alla tabella Prenotazioni
  - *bid*(Prenotazioni) → *bid*(Barche); come sopra per le modifiche
  - Vincoli di dominio esteso:
  - *esperienza* ∈ {1,...,10}
  - *età* ≥ 18
  - il nome di ogni velista e di ogni barca devono sempre essere specificati

- **Scrivere i comandi SQL per la creazione della BD.**

Soluzione

```
CREATE TABLE "Velisti" (
        "vid"                   int,
        "vnome"                 varchar(20) NOT NULL,
        "esperienza"            int,
        "età"                   real,
        PRIMARY KEY ("vid"),
        CHECK ("esperienza" >= 1 AND "esperienza" <= 10),
        CHECK ("età" >= 18)
);
```

Soluzione

```
CREATE TABLE "Barche" (
    "bid"                   int,
    "bnome"                 varchar(25) NOT NULL,
    "colore"                varchar(10),
    PRIMARY KEY ("bid")
);
```

Soluzione

```
CREATE TABLE "Prenotazioni" (
    "bid"                   int,
    "vid"                   int,
    "data"                  date,
    PRIMARY KEY ("bid", "vid", "data"),
    FOREIGN KEY ("bid") REFERENCES "Barche"("bid")
        ON UPDATE CASCADE  ON DELETE CASCADE,
    FOREIGN KEY ("vid") REFERENCES "Velisti"("vid")
        ON UPDATE CASCADE ON DELETE CASCADE
);
```

Interrogazioni

- Operazioni che:
  - **modificano l'istanza** di una BD
    - **inserimento** di una nuova riga in una tabella
    - **cancellazione** di una riga da una tabella
    - **modifica** dei valori di una o più colonne di una riga di una tabella
  - **estraggono** dati dall'istanza di una BD

- I comandi SQL che le realizzano fanno parte del *data manipulation language* (DML).

Interrogazioni

- In una BD relazionale il risultato di un'interrogazione è definito come:
  - una **nuova istanza** della BD, per operazioni di modifica (che non violino nessun vincolo d'integrità)
  - **una nuova tabella** (fittizia, non memorizzata nella BD), per operazioni di estrazione di dati

Esempi

- Esempi di interrogazioni per l'estrazione di dati dalla BD "Circolo Nautico":
  - Trovare i nomi dei velisti che hanno prenotato la barca numero 103
  - Trovare i nomi e le età di tutti i velisti
  - Trovare i nomi dei velisti che hanno prenotato almeno una barca
  - Trovare i codici dei velisti che hanno prenotato una barca rossa, oppure una barca verde

Interrogazioni: il comando SELECT

- Forma elementare:
  - SELECT "nome_colonna", ..., "nome_colonna"
    FROM "nome_tabella" ;
    estrae da tutte le righe di una tabella i valori delle colonne indicate
  - SELECT * FROM "nome_tabella" ;
    restituisce l'intera tabella
- Esempi:
  - Trovare il codice e il nome di tutti i velisti:
    `SELECT "vid","nome" FROM "Velisti";`
  - Trovare i dati su tutte le barche:
    `SELECT * FROM "Barche";`

SELECT: la clausola WHERE

- Per estrarre i valori delle colonne indicate, dalle sole righe di una tabella che soddisfano una data **condizione**:

- SELECT "nome_colonna", ..., "nome_colonna"
  FROM "nome_tabella"
  WHERE <*qualificazione* (condizione)> ;

SELECT: la clausola WHERE

- Le condizioni (*qualificazioni*) più semplici sono:
  - confronto tra il valore di una colonna e un valore costante dello stesso dominio
  - confronto tra i valori di due colonne aventi lo stesso dominio
- Si possono usare i seguenti operatori di confronto:
  `<` `<=` `=` `>` `>=` `<>`
- Si possono inoltre definire condizioni composte combinando altre condizioni (semplici o a loro volta composte) con gli operatori logici AND, OR, NOT.

SELECT: la clausola WHERE

- Nota:
  - i nomi delle colonne devono essere racchiusi tra doppi apici, per es.: `"Cognome"`
  - i valori costanti devono essere racchiusi tra apici singoli, per es.:
    ```
    "Cognome" = 'Rossi'
    "Data di nascita" >= '1990-01-01'
    "età" > '25'
    ```
  - i valori numerici (`integer` e `real`) possono anche essere scritti senza racchiuderli tra apici singoli, per es.: `"età" > 25`

Esempi

- Trovare i nomi dei velisti con più di 40 anni:

```
SELECT "vnome"
FROM  "Velisti"
WHERE "età" > 40;
```

- Trovare i nomi dei velisti con più di 40 anni, oppure con esperienza maggiore di 7:

```
SELECT "vnome"
FROM  "Velisti"
WHERE "età" > 40  OR  "esperienza" > 7;
```

Esercizi

- Si consideri la BD degli esempi mostrati in precedenza, composta dalle tabelle "Studenti", Insegnamenti" ed "Esami" (denominata da qui in poi "Segreteria Studenti"):

1)Trovare nomi e cognomi degli studenti nati il 1 gennaio 1991

2)Trovare nomi e cognomi degli studenti nati tra il 1 e il 31 gennaio 1991

3)Trovare i nomi degli insegnamenti da 6 crediti

4)Trovare le matricole degli studenti che hanno conseguito la votazione di 30 in qualche esame

Soluzioni

1)
```
SELECT "Nome", "Cognome"
FROM "Studenti"
WHERE "Data di nascita" = '1991-01-01';
```

2)
```
SELECT "Nome", "Cognome"
FROM "Studenti"
WHERE "Data di nascita" >= '1991-01-01'
            AND "Data di nascita" <= '1991-01-31';
```

3)
```
SELECT "Nome"
FROM "Insegnamenti"
WHERE "Crediti" = 6;
```

4)
```
SELECT "Studente"
FROM "Esami"
WHERE "Voto" = 30;
```

La clausola DISTINCT

- Contrariamente alle tabelle definite con CREATE TABLE, la tabella risultato di un'interrogazione **può** contenere righe identiche.
- Per es., nella BD "Circolo Nautico":
- `SELECT "colore" FROM "Barche";`
- restituisce:

colore
blu
rosso
verde
rosso

La clausola DISTINCT

- Per eliminare le eventuali righe duplicate (se queste non sono utili):
- SELECT DISTINCT "nome_colonna", ...
  FROM "nome_tabella"
  WHERE <*qualificazione* (condizione)> ;
- Esempio:
- `SELECT DISTINCT "colore" FROM "Barche";`
- restituisce:

colore
blu
rosso
verde

Interrogazioni su più tabelle

- Esempio: tovare i nomi dei velisti che hanno prenotato la barca numero 103.
- I dati richiesti si trovano in tabelle **diverse**

Velisti

vid	*vnome*	*esperienza*	*età*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

Prenotazioni

vid	*bid*	*giorno*
22	101	10/10/96
58	103	11/12/96

Barche

bid	*bnome*	*colore*
101	Interlake	blu
102	Interlake	rosso
103	Clipper	verde
104	Marine	rosso

Interrogazioni su più tabelle

- Intuitivamente, si può seguire questa procedura:
- 1) Schema della tabella risultato: una colonna (*vnome*):
  SELECT "Velisti"."vnome" ...
- 2) In quali tabelle si trovano le informazioni richieste? **Velisti** e **Prenotazioni**:
  SELECT "Velisti"."vnome" FROM "Velisti", "Prenotazioni" ...
- 3) Quali condizioni devono essere rispettate? Analizzare **tutte le possibili combinazioni** di righe delle tabelle in esame, ed estrarre *vnome* da una riga di "Velisti", se il *vid* della stessa riga coincide con il *vid* della riga di "Prenotazioni", e il *bid* di quest'ultima è 103:
- ```
  SELECT "Velisti"."vnome"
  FROM "Velisti", "Prenotazioni"
  WHERE "Velisti"."vid" = "Prenotazioni"."vid"
              AND "Prenotazioni"."bid"= 103;
  ```
  ```

Interrogazioni su più tabelle

- Nota: per distinguere le colonne di tabelle diverse, il nome di ogni colonna deve essere preceduto da quello della tabella, e i due nomi devono essere separati da un punto.
- Se il nome di una colonna compare in una sola tabella, e quindi non genera ambiguità, può essere scritto senza il nome della tabella corrispondente. Tuttavia questo rende l'interrogazione meno leggibile.

Variabili di *range*

- Per semplificare un'interrogazione su più tabelle, i nomi di queste ultime possono essere sostituiti da *variabili di range*, cioè nomi simbolici scelti dall'utente e indicati nella clausola FROM subito dopo il nome di ciascuna tabella.
- L'esempio precedente può essere riscritto come segue, associando alle tabelle "Velisti" e "Prenotazioni" le variabili di *range* V e P:
- ```
  SELECT V."vnome"
  FROM "Velisti" V, "Prenotazioni" P
  WHERE  V."vid" = P."vid"  AND  P."bid"= 103;
  ```
  ```

Esempio

- Trovare i cognomi degli studenti che hanno sostenuto l'esame avente codice `AMI01`:
- ```
  SELECT S."Cognome"
  FROM "Studenti" S, "Esami" E
  WHERE          E."Esame" = 'AMI01'  AND
                        S."Matricola" = E."Studente";
  ```
  ```

Strategia di valutazione concettuale

- Algoritmo per eseguire interrogazioni su più tabelle, descritto in precedenza in modo informale:

1. eseguire il **prodotto cartesiano** delle tabelle (insiemi di righe) indicate nella clausola FROM, ottenendo una nuova tabella P (ogni riga di P corrisponde a una combinazione di righe delle tabelle originali)
2. **selezionare** le righe di P che soddisfano la qualificazione (la condizione nella clausola WHERE)
3. **proiettare** le righe selezionate sulle colonne indicate nella clausola SELECT (cioè eliminare tutte le altre colonne)
4. se è presente la clausola DISTINCT, eliminare le eventuali righe duplicate

Esempio

```
SELECT V."vnome"
FROM "Velisti" V, "Prenotazioni" P
WHERE V."vid" = P."vid" AND P."bid" = 103;
```

Velisti

vid	*vnome*	*esperienza*	*età*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

Prenotazioni

vid	*bid*	*giorno*
22	101	10/10/96
58	103	11/12/96

Esempio

```
SELECT V."vnome"
FROM "Velisti" V, "Prenotazioni" P
WHERE V."vid" = P."vid" AND P."bid" = 103;
```

Velisti

vid	*vnome*	*esperienza*	*età*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

Prenotazioni

vid	*bid*	*giorno*
22	101	10/10/96
58	103	11/12/96

1. Prodotto cartesiano delle tabelle nella clausola FROM:

V.vid	*V.vnome*	*V.esperienza*	*V.età*	*P.vid*	*P.bid*	*P.giorno*
22	Dustin	7	45.0	22	101	10/10/96
22	Dustin	7	45.0	58	103	11/12/96
31	Lubber	8	55.5	22	101	10/10/96
31	Lubber	8	55.5	58	103	11/12/96
58	Rusty	10	35.0	22	101	10/10/96
58	Rusty	10	35.0	58	103	11/12/96

Esempio

```
SELECT V."vnome"
FROM "Velisti" V, "Prenotazioni" P
WHERE V."vid" = P."vid" AND P."bid" = 103;
```

Velisti			
vid	*vnome*	*esperienza*	*età*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

Prenotazioni		
vid	*bid*	*giorno*
22	101	10/10/96
58	103	11/12/96

2. Selezione delle righe che soddisfano la qualificazione:

V.vid	*V.vnome*	*V.esperienza*	*V.età*	*P.vid*	*P.bid*	*P.giorno*
58	Rusty	10	35.0	58	103	11/12/96

Esempio

```
SELECT V."vnome"
FROM "Velisti" V, "Prenotazioni" P
WHERE V."vid" = P."vid" AND P."bid" = 103;
```

Velisti			
vid	*vnome*	*esperienza*	*età*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

Prenotazioni		
vid	*bid*	*giorno*
22	101	10/10/96
58	103	11/12/96

3. Proiezione sulle colonne indicate nella clausola SELECT:

vnome
Rusty

Esercizi

1)Trovare i nomi dei velisti che abbiano prenotato almeno una barca

2)Trovare i colori delle barche prenotate da qualche velista di nome Lubber

3)Trovare i cognomi degli studenti che abbiano sostenuto almeno un esame

4)Trovare il nome degli insegnamenti dei quali qualche studentessa di nome Maria Bianchi abbia sostenuto l'esame

Soluzioni

1)
```
SELECT V."vnome"
FROM "Velisti" V, "Prenotazioni" P
WHERE V."vid" = P."vid";
```

1)
```
SELECT B."colore"
FROM "Velisti" V, "Prenotazioni" P,
            "Barche" B
WHERE V."vid" = P."vid" AND
                  P."bid" = B."bid" AND
                  V."vnome" = 'Lubber';
```

Soluzioni

3)
```
SELECT  S."Cognome"
FROM  "Studenti" S, "Esami" E
WHERE S."Matricola" = E."Studente";
```

4)
```
SELECT  I."Nome"
FROM "Insegnamenti" I, "Esami" E,
            "Studenti" S
WHERE I."Codice" = E."Esame" AND
              E."Studente" = S."Matricola" AND
              S."Nome" = 'Maria' AND
              S."Cognome" = 'Bianchi' ;
```