

TUTORATO DI BASI DI DATI

Mattia Fazzi: mattia.fazzi@student.unife.it

14/05/2019

SQL

- Linguaggio utilizzato per definire i dati nel database.
- Definire i vincoli.
- Possibilità di creare, modificare e eliminare le tabelle.

CREATE TABLE

- Comando utilizzato per creare una tabella dato il nome.
- Ogni attributo ha un tipo ben definito:
 - INTEGER,
 - FLOAT,
 - CHAR(n),
 - ...
- Possibilità di aggiungere vincoli su ogni attributo:
 - NOT NULL specifica che il valore non può essere vuoto

CREATE TABLE

- Possiamo aggiungere vincoli di chiave primaria, vincoli di integrità referenziale e vincoli di unicità.

ES:

```
CREATE TABLE STUDENT (  
    NAME VARCHAR(20) NOT NULL,  
    SURNAME VARCHAR(20) NOT NULL,  
    EMAIL VARCHAR(30) NOT NULL,  
    SSN VARCHAR(20),  
    PRIMARY KEY(EMAIL),  
    SSN UNIQUE );
```

DROP TABLE

- Comando per rimuovere una relazione dal database.
- La tabella non potrà più essere utilizzata in qualsiasi altra query.

ES:

```
DROP TABLE STUDENT;
```

ALTER TABLE

- Comando utilizzato per modificare una relazione.
- L'attributo aggiunto avrà valore NULL in tutte le tuple create precedentemente la modifica e per questo motivo il NOT NULL non è permesso.
- In ogni caso, le tuple create precedentemente possono essere modificate con il comando UPDATE.

ES:

```
ALTER TABLE STUDENT ADD SEX VARCHAR(1);
```

CREATE SCHEMA

- Specifica il nuovo schema del database.
- Nello schema del database vengono memorizzate tutte le informazioni riguardanti il database:
 - Tabelle
 - Attributi delle tabelle
 - Vincoli
 - ...

OPZIONI DI INTEGRITA' REFERENZIALE

- Nel momento in cui una tabella viene cancellata o modificata, si possono definire dei comportamenti che assumeranno i valori che puntano ad attributi di quella tabella per evitare situazioni impreviste.
- Keywords:
 - RESTRICT: impedisce la modifica nella tabella a cui punta
 - CASCADE: aggiorna il valore se l'attributo viene modificato nella tabella puntata
 - SET NULL: aggiorna il valore a NULL
 - SET DEFAULT: aggiorna il valore a un valore definito di default.

OPZIONI DI INTEGRITA' REFERENZIALE

ES:

```
CREATE TABLE STUDENT (
```

```
...
```

```
    CORSO_DI_LAUREA DEFAULT 0,
```

```
    CORSO_DI_LAUREA INTEGER REFERENCES ID_CORSO ON  
    DELETE SET DEFAULT ON UPDATE CASCADE);
```

TIPI ADDIZIONALI

- DATE:
 - Formato utilizzato per le date: yyyy-mm-dd
- TIME:
 - Formato utilizzato per un orario: hh:mm:ss
- TIME(i):
 - Formato utilizzato per un orario in cui possiamo specificare i millisecondi
- TIMESTAMP:
 - Formato che contiene sia data sia orario.

OPERAZIONI DI MODIFICA:INSERT

- Comando utilizzato per inserire uno o più tuple in una tabella.
- Gli attributi inseriti possono essere nello stesso ordine in cui sono specificati nella CREATE TABLE oppure si può specificare quali attributi verranno definiti in fase di inserimento.

ES:

```
INSERT INTO STUDENTS (NAME, SURNAME, EMAIL) VALUES  
(‘MATTIA’, ‘FAZZI’, ‘mattia.fazzi@student.unife.it’)
```

OPERAZIONI DI MODIFICA:DELETE

- Rimuove tutte le tuple da una tabella che soddisfano la condizione specificata nella clausola WHERE.
- Se non specificata la clausola WHERE, verranno eliminate tutte le tuple.
- Bisogna prestare particolare attenzione ai vincoli delle altre tabelle che puntano agli attributi della tabella da cancella perché potrebbero bloccare la cancellazione.

ES:

```
DELETE FROM STUDENTS WHERE NAME = 'MATTIA'
```

OPERAZIONI DI MODIFICA:UPDATE

- Comando utilizzato per modificare gli attributi di una o più tuple.
- Se inserita la clausola WHERE, verranno modificate solo le tuple che soddisfano la clausola.
- In questo caso deve essere presente la clausola SET per specificare con quale valore si modificherà l'attributo.

ES:

```
UPDATE STUDENTS SET ID_CORSO = 3 WHERE ID_CORSO = 2
```

QUERY SQL

- Meccanismo per ottenere i dati dal database.
- Formato:

```
SELECT <attributi>
```

```
FROM <tabelle>
```

```
WHERE <condizioni>
```

ES:

```
SELECT NAME, SURNAME
```

```
FROM STUDENTS
```

```
WHERE ID_CORSO = 1
```

ALIASES

- E' possibile che due tabelle utilizzino lo stesso nome per un attributo e questo renderebbe difficile per il DBMS capire a quale attributo fare riferimento.
- Per questo motivo si utilizzano gli alias per rendere tutto più chiaro e univoco.

ES:

```
SELECT S.NAME, S.SURNAME, E.NAME, E.SURNAME  
FROM STUDENTS AS S, STUDENTS AS E  
WHERE E.SUPERSSN = S.SSN
```

ASTERISCO (*)

- L'asterisco si utilizza per ottenere tutti gli attributi delle relazioni specificate nella clausola FROM.

ES:

```
SELECT *
```

```
FROM STUDENTS
```

```
WHERE ID_STUDENTE = 1
```

Seleziono tutti i valori dello studente con id = 1

DISTINCT

- La keyword distinct serve per eliminare tutti i valori duplicati di una query.

ES:

```
SELECT DISTINCT NAME,  
FROM STUDENTS
```

In questo caso non ho la clausola WHERE quindi selezionerò tutti i nomi degli studenti ed eliminerò i duplicati.

OPERAZIONI

- E' possibile eseguire delle operazioni insiemistiche tra le query.
- Operazioni:
 - Unione: unione di due risultati di due differenti query (**UNION**)
 - Differenza: eliminare dai risultati della prima query i risultati presenti anche nella seconda query (**MINUS**)
 - INTERSEZIONE: selezionare le tuple di una query che compaiono anche nella seconda (**INTERSECT**)
- In ogni caso, le tuple duplicate verranno eliminate.
- E' importante che le tuple delle due query abbiano lo stesso set di attributi.

QUERY INNESTATE

- Le query innestate vengono utilizzate quando è complicato definire determinati valori di attributi nella clausola WHERE e risulterebbe molto più comodo prima selezionarli con una select e successivamente utilizzare i valori trovati per selezionare solamente alcune tuple di una determinata tabella.
- Possiamo avere più livelli di query innestate.

EXISTS e IN

- EXISTS viene utilizzato nella clausola WHERE quando si vogliono tutte quelle tuple che ESISTONO anche in un'altra query.

ES:

```
SELECT NAME FROM STUDENTS WHERE EXISTS (  
    SELECT * FROM PROFESSOR)
```

- In questo caso, si vogliono trovare tutti i nomi che hanno sono in comune tra la relazione studente e la relazione professore.

EXISTS e IN

- La keyword IN viene utilizzata insieme a un attributo a cui fare riferimento.

ES:

```
SELECT *
```

```
FROM STUDENTS
```

```
WHERE NAME IN (SELECT NAME FROM PROFESSOR)
```

JOIN

- Specificata in una clausola FROM, JOIN permette di correlare due tabelle. L'attributo su cui viene fatto la join è la foreign key di una tabella e chiave nell'altra.
- La join può essere fatta su più tabelle, sempre rispettando i valori dell'attributo da mettere in correlazione.

ES:

```
SELECT S.NAME, S.SURNAME FROM STUDENTS AS S JOIN  
CORSO_DI_LAUREA AS C ON S.ID_CORSO = C.ID  
WHERE C.NAME = 'INFORMATICA'
```

FUNZIONI DI AGGREGAZIONE

- COUNT(*): conta tutti le tuple presenti nel risultato di una query.
- SUM(<attributo>): Somma tutti i valori presenti nella colonna risultato della query
- MIN(<attributo>): Trova il minimo valore dell'attributo presente nel risultato della query.
- MAX(<attributo>): Trova il massimo valore dell'attributo presente nel risultato della query.
- AVG(<attributo>): calcola la media dei valori dell'attributo presente nel risultato della query.

AGENZIA DI VIAGGI

Un'agenzia di viaggi vi commissiona il database per la loro piattaforma online su cui i clienti (nome, cognome, sesso, codfiscale, data nascita, data iscrizione) possono contattare direttamente un agente di viaggio (nome, cognome, sesso, data nascita, data assunzione, codfiscale) tramite messaggi (testo, data invio) per chiedere informazioni riguardo a offerte del sito. Le offerte (costo, data creazione, durata di validità, durata della vacanza) sono pacchetti composti da: un pernottamento in un hotel (nome, città, indirizzo, #stelle, email hotel) o in B&B(nome, città, indirizzo, email proprietario, nome proprietario) e modalità di viaggio se previsto (mezzo di trasporto, città partenza, città arrivo, durata del viaggio, costo). L'agente di viaggio deve completare la prenotazione(data prenotazione, data partenza, data ritorno, costo complessivo) dell'offerta da parte di un cliente.

SQL

- Selezionare le offerte che costano almeno 200 euro.
- Selezionare il nome e indirizzo dei soggiorni che si trovano a firenze.
- Selezionare la data partenza e data ritorno delle prenotazioni che hanno costo complessivo minore di 350 euro e che sono state prenotate dopo il 2016.
- Selezionare il nome degli agenti di viaggio che hanno effettuato una prenotazione con costo complessivo maggiore di 400 euro.
- Selezionare le offerte che comprendono soggiorni che si trovano a milano.

PALESTRA

Il CUS vi incarica di progettare il nuovo database della centro sportivo. Ogni utente (nome, cognome, data nascita, sesso, codice fiscale) sarà un cliente (data iscrizione) o un personal trainer (data assunzione). Ogni cliente può iscriversi a determinati corsi (nome, tipologia, descrizione) che saranno tenuti da un personal trainer. Ogni cliente può prenotare una seduta di allenamento che sarà supervisionata da un personal trainer. La prenotazione avrà data, orario e durata della seduta. Ogni utente può essere iscritto a un determinato club sportivo (tipologia e nome allenatore) che si terrà in una specifica palestra (nome, città, indirizzo). Ogni utente può decidere di comprare un abbonamento (identificativo, durata, data inizio, data fine) o di comprare un numero generico di entrate (numero entrate: 10,20 o 30, data acquisto, data validità).

SQL

Selezionare gli utenti che si sono iscritti dopo settembre 2017.

Selezionare il nome e il cognome dei personal trainer femmine.

Selezionare le sedute di allenamento con durata superiore a 60 minuti.

Selezionare il nome e il cognome dei personal trainer che hanno tenuto una seduta di allenamento prima delle ore 15.

Selezionare l'indirizzo delle palestre che sono sede del club allenato da «Ciccio Graziani».

Selezionare il nome, cognome e sesso dei personal trainer tengono un corso a cui è iscritto almeno un cliente che ha un abbonamento che finisce dopo gennaio 2019.

RAGGRUPPAMENTO

Contare quanti maschi hanno sottoscritto un abbonamento da 10 ingressi.

Trovare il nome e cognome del personal trainer più giovane.

Calcolare la media dell'età delle persone iscritta al corso di tipologia «Crossfit»