

# SQL / 23-03

Create Table permette di schematizzare la relazione  
Si interogisce con il DQL (Data Definition Language)  
Include anche il linguaggio di gestione dei dati DML.  
Le tabelle SQL non sono relazioni in generale

Create Table crea una tabella con una sola colonna:

Create Table R (A char)

(senza chiave)

→ non ha relazione

A
'a'
'a'

} doppie

Questo perché evita il controllo di tuple già presenti uguali

Se pongo una chiave allora sarà il DB a effettuare il controllo in maniera efficiente

$R(A, B, C) = R(A, \underline{B}, C)$  in schema relazionale

↓  
≠ in SQL

posso mettere valori duplicati

Di SQL si hanno dei "dialetti", e dipende dal sistema che lo usa,

→ usiamo lo standard

→ PostgreSQL può avere peculiarità di diverso

Studiamo lo standard x relazionali (altri sistemi: Oracle, MySQL etc...)

SQL non è case sensitive a meno delle stringhe

("A" ≠ "a")

per ogni colonna ALTRE nome - dominio

CREATE TABLE R

( A char, → 1 carattere

- 33,33 e +33,33

B numeric(4,2), → (tot cifre, di cui 2 decimali)

C date,

D char(20) → stringa di 20 caratteri max

uso varchar x stringhe grandi ma posso avere

stringhe piccole (dim allocata al momento)

) → separatore

(non serve se terminatore)

(x i testi vlob)

In SQL se non specifico un valore allora si intende attributo (default) opzionale. Posso invece esprimere i valori che non devono essere nulli quindi

R(A, B, C, D) A, B e D not null

↓  
è chiave, implica not null

CREATE TABLE R

( A char PRIMARY KEY NOT NULL,

implicito

B numeric(4,2) UNIQUE NOT NULL,

C date,

D char(20)

NOT NULL REFERENCES R'

(FOREIGN KEY D)  
NON NECESSARIO SU  
FACIL RIF A D

/esterna  
una chiave alternativa  
di cosa può avere  
valori null

oppure

oppure

PRIMARY KEY (A) UNIQUE (B) \*);

↳ chiave esterna  
a un tabella R'

↳ è uguale a •, ma obbligatorio se la più colonne come chiave

Dopo la creazione delle tabelle posso porre vincoli per modificare  
SUCCESSIVAMENTE: add Constraint

Di chiavi primarie in SQL ve n'è una

Di chiavi alternative (**UNIQUE**) infinite

SQL capisce da solo la chiave da prendere con una chiave esterna  
da  $R'$  •, se questa non ha chiave primaria viene segnalato un  
errore. Oppure se la chiave esterna è fatta di più attributi ( $R'$  ha  
come primaria più attributi). Scrivo in fondo \* :

FOREIGN KEY (D) REFERENCES  $R'$

Non è necessario specificare le colonne di  $R'$   
ma necessario foreign key D.

Con la parola chiave CHECK per porre dei vincoli ulteriori:  
posso confrontarli di una sola colonna o di più colonne  
(es. età  $\leq 18$  oppure  $A < B$ )

Se non rispettiamo i vincoli SQL lo segnala (e non esegue l'operazione)

Quando lego una chiave esterna a  $R$  da  $R'$  può darsi che io non  
abbia controllo di  $R'$  potrei violare il vincolo di integrità referenziale:  
se riferisco  $D$  di  $R'$  ma questa viene rimossa ho un problema  
anche in  $R$ . Avendo dichiarato  $D$  esterna IMPEDISCE la rimozione  
di tale colonna da  $R'$ .

È un vincolo forte, dunque spesso si gestisce l'operazione con altre

→ FOREIGN KEY (D) REFERENCES  $R'$  ON DELETE CASCADE  
(modifica della chiave, es. aggiunta) ← / ON UPDATE SET NULL  
in base al tipo  
ps zero  
→ : numeri

se qualunque info su Studente viene cancellata da  $R'$  allora anche  
in  $R$  posso togliere e cancellare ogni info su Studente, non più di mio interesse

