

SQL

Esercitazioni pratiche

- Per SQL è possibile (e fondamentale) svolgere esercitazioni pratiche
- Verranno anche richieste come condizione per svolgere le prove parziali
- Soprattutto sono utilissime
- Si può utilizzare qualunque DBMS
 - IBM DB2, Microsoft SQL Server, Oracle, PostgreSQL, ...
- A lezione utilizziamo PostgreSQL

CREATE TABLE, esempi

```
CREATE TABLE corsi(  
  codice numeric NOT NULL PRIMARY KEY,  
  titolo character(20) NOT NULL,  
  cfu numeric NOT NULL)
```

```
CREATE TABLE esami(  
  corso numeric REFERENCES corsi (codice),  
  studente numeric REFERENCES studenti (matricola),  
  data date NOT NULL,  
  voto numeric NOT NULL,  
  PRIMARY KEY (corso, studente))
```

La chiave primaria viene definita come NOT NULL anche se non lo specifichiamo (in Postgres)

DDL, in pratica

- In molti sistemi si utilizzano strumenti diversi dal codice SQL per definire lo schema della base di dati
- Vediamo (per un esempio su cui lavoreremo)

SQL, operazioni sui dati

- interrogazione:
 - **SELECT**
- modifica:
 - **INSERT, DELETE, UPDATE**

Inserimento

(necessario per gli esercizi)

```
INSERT INTO Tabella [ ( Attributi ) ]  
VALUES( Valori )
```

oppure

```
INSERT INTO Tabella [ ( Attributi ) ]  
SELECT ...  
(vedremo più avanti)
```

```
INSERT INTO Persone VALUES ('Mario',25,52)
```

```
INSERT INTO Persone(Nome, Reddito, Eta)  
VALUES('Pino',52,23)
```

```
INSERT INTO Persone(Nome, Reddito)  
VALUES('Lino',55)
```

Maternità

Madre	<u>Figlio</u>
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Paternità

Padre	<u>Figlio</u>
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Persone

<u>Nome</u>	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Esercizi

- Definire la base di dati per gli esercizi
 - installare un sistema
 - creare lo schema
 - creare le relazioni (CREATE TABLE)
 - inserire i dati
- eseguire le interrogazioni
 - suggerimento, usare schemi diversi
set search_path to <nome schema>

```
create table persone (  
  nome char (10) not null primary key,  
  eta numeric not null,  
  reddito numeric not null);  
create table paternita (  
  padre char (10) not null ,  
  figlio char (10) not null primary key);  
...  
insert into Persone values('Andrea',27,21);  
...  
insert into Paternita values('Sergio','Franco');  
...
```

Istruzione **SELECT** (versione base)

SELECT ListaAttributi
FROM ListaTabelle
[**WHERE** Condizione]

- "target list"
- clausola **FROM**
- clausola **WHERE**

Intuitivamente

```
SELECT ListaAttributi  
FROM ListaTabelle  
[ WHERE Condizione ]
```

- Prodotto cartesiano di ListaTabelle
- Selezione su Condizione
- Proiezione su ListaAttributi

Selezione e proiezione

- Nome e reddito delle persone con meno di trenta anni

$\text{PROJ}_{\text{Nome, Reddito}}(\text{SEL}_{\text{Eta} < 30}(\text{Persone}))$

```
select nome, reddito  
from persone  
where eta < 30
```

Selezione, senza proiezione

- Nome, età e reddito delle persone con meno di trenta anni

$SEL_{\text{Eta} < 30}(\text{Persone})$

```
select *  
from persone  
where eta < 30
```

Proiezione, senza selezione

- Nome e reddito di tutte le persone

$\text{PROJ}_{\text{Nome, Reddito}}(\text{Persone})$

```
select nome, reddito  
from persone
```

Proiezione, con ridenominazione

- Nome e reddito di tutte le persone

$REN_{Anni} \leftarrow_{Eta} (PROJ_{Nome, Eta}(Persone))$

```
select nome, eta as anni  
from persone
```


Espressioni nella target list

```
select Nome, Reddito/12 as redditoMensile  
from Persone
```

Condizione complessa

```
select *  
from persone  
where reddito > 25  
and (eta < 30 or eta > 60)
```

Condizione “LIKE”

- Le persone che hanno un nome che inizia per 'A' e ha una 'd' come terza lettera

```
select *  
from persone  
where nome like 'A_d%'
```

Gestione dei valori nulli

Impiegati

Matricola	Cognome	Filiale	Età
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

- Gli impiegati la cui età è o potrebbe essere maggiore di 40

SEL (Età > 40) OR (Età IS NULL) (Impiegati)

Proiezione, attenzione

```
select  
  madre  
from maternita
```

```
select distinct  
  madre  
from maternita
```

Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Selezione, proiezione e join

- I padri di persone che guadagnano più di 20

```
PROJPadre(paternita  
  JOINFiglio = Nome  
  SELReddito > 20(persone))
```

```
select distinct padre  
from persone, paternita  
where figlio = nome and reddito > 20
```

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

```

PROJNome, Reddito, RP (SELReddito>RP
(RENNP,EP,RP ← Nome, Eta, Reddito (persone)
      JOINNP=Padre
(paternita JOINFiglio =Nome persone)))

```

```

select f.nome, f.reddito, p.reddito
from persone p, paternita, persone f
where p.nome = padre and
      figlio = f.nome and
      f.reddito > p.reddito

```


SELECT, con ridenominazione del risultato

```
select figlio, f.reddito as reddito,  
       p.reddito as redditoPadre  
from persone p, paternita, persone f  
where p.nome = padre and figlio = f.nome  
and f.reddito > p.reddito
```