SQL

Esercitazioni pratiche

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

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	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

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

$$PROJ_{Nome,\ Reddito}(SEL_{Eta<30}(Persone))$$

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

Selezione, senza proiezione

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

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

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

Proiezione, senza selezione

- Nome e reddito di tutte le persone

$$PROJ_{Nome,\ Reddito}(Persone)$$

```
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
```

Proiezione, attenzione

```
select
    madre
from maternita
```

```
select distinct
    madre
from maternita
```

Condizione complessa

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

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

$$PROJ_{Padre}(paternita \; JOIN_{Figlio = Nome} \; SEL_{Reddito>20}(persone))$$

```
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

$$\text{PROJ}_{\text{Nome, Reddito, RP}}\,(\text{SEL}_{\text{Reddito>RP}}\,(\text{REN}_{\text{NP,EP,RP} \leftarrow \text{Nome,Eta,Reddito}}(\text{persone})\ \text{JOIN}_{\text{NP=Padre}}\ (\text{paternita JOIN}_{\text{Figlio =Nome}}\ \text{persone})))$$

```
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
```

Join esplicito

- Padre e madre di ogni persona

```
select paternita.figlio,padre, madre
from maternita, paternita
where paternita.figlio = maternita.figlio

select madre, paternita.figlio, padre
from maternita join paternita on
      paternita.figlio = maternita.figlio
```

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

```
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 f.nome, f.reddito, p.reddito
from (persone p join paternita on p.nome = padre)
        join persone f on figlio = f.nome
where f.reddito > p.reddito
```

Join esterno: "outer join"

- Padre e, se nota, madre di ogni persona

```
select paternita.figlio, padre, madre
from paternita left join maternita
        on paternita.figlio = maternita.figlio
```

```
select paternita.figlio, padre, madre
from paternita left outer join maternita
        on paternita.figlio = maternita.figlio
```

- outer e' opzionale

Ordinamento del risultato

- Nome e reddito delle persone con meno di trenta anni in ordine alfabetico

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

Espressioni nella target list

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

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

$$\text{SEL}_{(\text{Età} > 40)\ \text{OR}\ (\text{Età IS NULL})}\ (\text{Impiegati})$$

Unione

```
select A,  B
from R
union
select A , B
from S
```

```
select A,  B
from R
union all
select A , B
from S
```

Notazione posizionale!

```
select padre, figlio
from paternita
union
select madre, figlio
from maternita
```

	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Notazione posizionale, 2

```
select padre, figlio
from paternita
union
select figlio, madre
from maternita
```

NO!

```
select padre, figlio
from paternita
union
select madre, figlio
from maternita
```

OK

Notazione posizionale, 3

- Anche con le ridenominazioni non cambia niente:

```
select padre as genitore, figlio
from paternita
union
select figlio, madre as genitore
from maternita
```

- Corretta:

```
select padre as genitore, figlio
from paternita
union
select madre as genitore, figlio
from maternita
```

Differenza

```
select Nome
from Impiegato
except
select Cognome as Nome
from Impiegato
```

Intersezione

```
select Nome
from Impiegato
intersect
select Cognome as Nome
from Impiegato
```

Operatori aggregati: COUNT

- Il numero di figli di Franco

```
select count(*) as NumFigliDiFranco
from Paternita
where Padre = 'Franco'
```

COUNT DISTINCT

```
select count(*) from persone

select count(reddito) from persone

select count(distinct reddito) from persone
```

Altri operatori aggregati

- SUM, AVG, MAX, MIN

- Media dei redditi dei figli di Franco

```
select avg(reddito)
from persone join paternita on nome=figlio
where padre='Franco'
```

Operatori aggregati e valori nulli

```
select avg(reddito) as redditomedio
from persone
```

Operatori aggregati e target list

- un'interrogazione scorretta:

```
select nome, max(reddito)
from persone
```

- di chi sarebbe il nome? La target list deve essere omogenea

```
select min(eta), avg(reddito)
from persone
```

Operatori aggregati e raggruppamenti

- Il numero di figli di ciascun padre

```
select Padre, count(*)  AS NumFigli
from paternita
group by Padre
```

Condizioni sui gruppi

- I padri i cui figli hanno un reddito medio maggiore di 25; mostrare padre e reddito medio dei figli

```
select padre, avg(f.reddito)
from persone f join paternita on figlio = nome
group by padre
having avg(f.reddito) > 25
```

Operatori aggregati e target list

- un'interrogazione scorretta:

```
select nome, max(reddito)
from persone
```

- di chi sarebbe il nome? La target list deve essere omogenea

```
select min(eta), avg(reddito)
from persone
```