

---

Basi di Dati  
Modulo Laboratorio

## Lezione 6: Interrogazioni di tipo insiemistico

DR. SARA MIGLIORINI

# Interrogazioni di tipo insiemistico

- SQL mette a disposizione anche degli **operatori insiemistici** per manipolare i risultati di più query.
- Gli operatori insiemistici si possono utilizzare solo al livello più esterno di una query, operando sul risultato di due o più clausole SELECT.
- Gli operatori insiemistici sono: **UNION**, **INTERSECT** e **EXCEPT**.
- Si possono avere sequenze di UNION/INTERSECT/EXCEPT

Sintassi

```
query_1 { UNION | INTERSECT | EXCEPT } [ALL] query_2
```

# Interrogazioni di tipo insiemistico

- Gli operatori si possono applicare solo quando `query1` e `query2` producono risultati con lo stesso numero di colonne e di tipo compatibile fra loro.
- Tutti gli operatori eliminano i duplicati dal risultato a meno che `ALL` non sia stato specificato.
- `UNION` aggiunge il risultato di `query2` a quello di `query1`.
- `INTERSECT` restituisce le righe che sono presenti sia nel risultato di `query1` sia in quello di `query2`.
- `EXCEPT` restituisce le righe di `query1` che non sono presenti nel risultato di `query2`. In pratica esegue la differenza insiemistica. (

# Interrogazioni di tipo insiemistico

Esempio: UNION ALL

Visualizzare i nomi degli insegnamenti e i nomi dei corsi di laurea che non iniziano per 'A' mantenendo i duplicati.

```
SELECT nomeins  
FROM Insegn  
WHERE NOT nomeins LIKE 'A%'
```

UNION ALL

```
SELECT nome  
FROM CorsoStudi  
WHERE NOT nome LIKE 'A%';
```

(8253 righe)

# Interrogazioni di tipo insiemistico

Esempio: INTERSECT ALL

Visualizzare i nomi degli insegnamenti che sono anche nomi di corsi di laurea.

```
SELECT nomeins  
FROM Insegn
```

```
INTERSECT ALL
```

```
SELECT nome  
FROM CorsoStudi;
```

(8253 righe)

# Interrogazioni di tipo insiemistico

## Esempio

Visualizzare i nomi degli insegnamenti che NON sono anche nomi di corsi di laurea.

```
SELECT nomeins  
FROM Insegn
```

```
EXCEPT
```

```
SELECT nome  
FROM CorsoStudi ;
```

(7709 righe)

# Le VISTE

---

- Le viste sono **tabelle «virtuali»** il cui contenuto dipende dal contenuto delle altre tabelle della base di dati.
- In SQL le viste vengono definite associando un nome ed una lista di attributi al risultato dell'esecuzione di un'interrogazione.
- Ogni volta che si usa una vista, si esegue la query che la definisce.
- Nell'interrogazione che definisce la vista possono comparire anche altre viste.
- SQL **non ammette** però:
  - dipendenze immediate (definire una vista in termini di se stessa) o ricorsive (definire una interrogazione di base e una interrogazione ricorsiva);
  - dipendenze transitive circolari (V1 definita usando V2, V2 usando V3, . . . , Vn usando V1).

# Le VISTE

## Sintassi

```
CREATE [ TEMP ] VIEW nome [( col_name [, ...]) ] AS  
query
```

- **TEMP**: la vista è temporanea. Quando ci si sconnette, la vista viene distrutta. È un'estensione di PostgreSQL. Nella base di dati did2014 si possono fare solo viste temporanee.
- **column\_name**: nomi delle colonne che compongono la vista. Se non si specificano, si ereditano dalla query.
- **query** deve restituire un insieme di attributi pari e nel medesimo ordine a quello specificato con ( **column\_name** [, ...] ) se presente.
- SQL permette che una vista sia aggiornabile solo quando ciascuna tupla della vista corrisponde a una sola tupla di ciascuna tabella di base (argomento avanzato).



# Le VISTE

## Esempio

Definire la vista che contiene gli insegnamenti erogati completi di nomeins e codiceins presi dalla tabella Insegn.

```
CREATE TEMP VIEW InsErogatiCompleti AS
SELECT I. nomeins , I. codiceins , IE .*
FROM InsErogato IE
JOIN Insegn I ON IE. id_insegn = I.id;
```

Risultato:

`\dv InsErogatiCompleti`

Lista delle relazioni

Schema	Nome	Tipo	Proprietario
--------	------	------	--------------

-----+-----+-----+-----
-------------------------

pg_temp_2	inserogaticompleti	vista	smigliorini
-----------	--------------------	-------	-------------

`\d+ InsErogatiCompleti`

<restituisce tutte le colonne e la definizione della query>

# Le VISTE

Esempio di vista necessaria per una query

Si vuole determinare quali sono i corsi di studi con il massimo numero di nome di insegnamenti diversi.

Prima si crea una vista:

```
CREATE TEMP VIEW InsCorsoStudi (Nome , NumIns) AS
  SELECT CS.nome , COUNT ( DISTINCT I. nomeins )
  FROM CorsoStudi CS
  JOIN InsErogato IE ON CS.id=IE. id_corsostudi
  JOIN Insegn I ON IE. id_insegn = I.id
  GROUP BY CS. nome;
```

# Le VISTE

Esempio di vista necessaria per una query (cont.)

Poi la si usa:

```
SELECT Nome , NumIns  
FROM InsCorsoStudi  
WHERE NumIns = ANY (  
    SELECT MAX ( NumIns ) FROM InsCorsoStudi  
);
```

Laurea specialistica in Medicina e Chirurgia: 1231