

# INTERROGAZIONI INSIEMISTICHE:

**OPERATORI INSIEMISTICI** per **UNIRE** Risultato di più OPERAZIONI

Si Applicano al livello più Esterno della QUERY.

**UNION** → Aggiunge il RISULTATO di  $Q_2$  a  $Q_1$  (Eliminando duplicati)

**INTERSECT** → Tuple Comuni in  $Q_1$  ed in  $Q_2$

**EXCEPT** → Tuple che ci Sono Solo in  $Q_1$  (DIFFERENZA)

**Tipo di uso** ⇒  $\langle \text{QUERY}_1 \rangle [*] \langle \text{QUERY}_2 \rangle$

Vengono Eseguite le 2 Query Singolarmente, creando così 2 Insiemi di TUPLE e poi Si uniscono  $Q_1$  e  $Q_2$  a Seconda della Semantica dell' Operatore

## VINCOLI:

Le 2 Query devono produrre Risultati con lo Stesso Numero di Colonne e TIPI compatibili Tra loro

Stringhe 30 char e 40 char sono compatibili

A parità di Posizione, il Tipo deve Essere Compatibile

SQL <> ALGEBRA RELAZIONALE

↳ lavoro con INSIEMI DI TUPLE e NON ci sono MAI duplicati

→ Serve DISTINCT... ma Se usa gli OPERATORI  
Insiemeistici vengono ELIMINATI i duplicati se non si Specifica  
la parola chiave ALL.

Query che Visualizzi i Nomi degli Insegnamenti e dei Corsi  
di Laurea che non Iniziano con la Lettera 'A'

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

①

```
SELECT Nome  
FROM CorsoStudi  
WHERE Nome NOT LIKE 'A%'
```

②

① UNION ②

C'è un UNICA Colonna con Tipi COMPATIBILI [Varchar(...)]

Può ESSERE <> ma Sono Stringhe ←

① UNION ALL ② Nel Caso in cui Volessi Tenere i  
duplicati presenti

Per RIMUOVERE i DUPLICATI devo fare ORDINAMENTO...  
Quindi l'operazione costosa è l'ORDINAMENTO

~~Visualizzare~~ i nomi degli INSEGNAMENTI che Sono  
anche nomi di Corsi di Laurea

```
SELECT NomeIns FROM Insegn (1)
```

```
SELECT Nome FROM CorsoStudi (2)
```

(1) INTERSECT <ALL> (2)

~~Visualizzare~~ i nomi degli Insegnamenti che NON Sono  
anche nomi di Corsi di Laurea

```
SELECT NomeIns FROM Insegn (1)
```

```
SELECT Nome FROM CorsoStudi (2)
```

(1) EXCEPT <ALL> (2)

QUERY POSSONO diventare COMPLESSE, con QUERY ANNIDATE e  
posso Evitare di RISCRIVERE Tramite le **VISTE**.

Tabelle VIRTUALI, il cui CONTENUTO dipende dal CONTENUTO  
di ALTRE TABELLE.


In SQL le VISTE Sono OTTENUTE Associando un Nome ed una lista di Attributi al RISULTATO dell'Esecuzione di un Interrogazione (UN SALVATAGGIO)

Ogni Volta che Si USA il Nome di una Vista in una QUERY viene Eseguita la QUERY Corrispondente

Non posso Avere delle DIPENDENZE CIRCOLARI 

CREATE [TEMP] VIEW <Nome-Vista> [<Nome-Colonne>] AS ..

Se VOGLIAMO Sovrascrivere  
Quelli delle QUERY 

 Quando mi Scollego dal DB, Viene Eliminata la VISTA al Termine della SESSIONE

Anche le TABELLE Possono ESSERE TEMPORANEE (ad Esempio come TEST)

Definire una Vista che CONTIENE Gli Insegnamenti Erogati (nomeIns, codiceIns) Presi dalle Tabelle INSEGN

Query Sarebbe

```
SELECT NameIns, CodiceIns, Insegnato *  
FROM Insegn JOIN InsErogato ON  
I.id ON IE.id-ins
```

Vista Corrispondente:

```
CREATE VIEW Inserogato Completo AS
```

```
SELECT NameIns, CodiceIns, Inserogato *  
FROM Insegn JOIN InsErogato ON  
I.id ON IE.id-ins
```

Su PGAdmin4 c'è una Sezione per Le Viste

Da TERMINALE:

\D → Tutte le Tab

\D + NomeTab → definizione Tab + Cod. SQL che le definisce

\DV → Elenco liste

Viste Sono NECESSARIE per RISPONDERE ad una QUERY, ad  
Esempio:

~~///~~ Corsi Studio con il Massimo Numero di Nomi di Insegn.  
diversi

Ci Richiede di Applicare 2 Operatori di Aggregazione<sup>a</sup> CASCATA

NumIns ≠ in un Corso Studi:

```
SELECT Cs.Nome, COUNT(DISTINCT I.NomeIns)  
FROM CorsoStudi cs JOIN InsErogato IE  
JOIN Insegnamento I  
GROUP BY Cs.Nome
```

Creo una Vista:

```
CREATE VIEW InsCorsoStudi (Nome, NumIns) AS
  SELECT Cs.Nome, COUNT(DISTINCT I.NomeIns)
  FROM CorsoStudi cs JOIN InsErogato IE
                        JOIN Insegnamento I
  GROUP BY Cs.Nome
```

Ora

```
SELECT I.nome, I.numIns
FROM InsCorsoStudi I
```

```
WHERE I.numIns = ANY (
  SELECT MAX (NumIns)
  FROM InsCorsoStudi)
```

La Vista

SELEZIONO IL MASSIMO #Insegnamenti IN UN CORSO DI STUDI

Senza Vista Avrei dovuto fare 2 Query Annidate nel From.

~~Max(count(...))~~ Non Andava Bene, deve Essere un Campanello d'Allarme Quando si Tratta:

- Min
- MAX

## SOLUZIONI QUERY LAB: (Es. 05)

Trovare l'ID, Nome, Cognome che Nell' Anno Acc 2010/2011 hanno Tenuto un Ins. che **NON** hanno Tenuto Nell' Anno Accademico Precedente

→ Come il MAI mi Porta ad una sottrazio.

```
SELECT DISTINCT P1.id, P1.Nome, P1.Cognome
FROM Persona P1 JOIN Doenza D1 JOIN insErogato IE1
      JOIN Inseg I1
WHERE IE1.annoacc = '2010/2011' AND P1.ID NOT IN (

  SELECT D2.id-Persona
  FROM Doenza D2 JOIN insErogato JOIN Insegn.
  WHERE IE2.anno = '2009/2010' AND
    I2.nome = I1.nome )
DATA BINDING QUI

ORDER BY P1.id
```

**QUERY 4:** Nomi Corsi Studi che NON hanno MAI erogato Insegnamenti che CONTENGONO la Stringa 'Matemat'

```
SELECT Nome
FROM CorsoStudi
WHERE CorsoStudi NOT IN (SELECT CorsoStudi.ID
                           FROM Insegn
                           JOIN InsErogato
                           JOIN CorsoStudi
                           WHERE nomeIns LIKE '...')
```

**QUERY 5:** Trovare Nome, Cognome, Telefono dei DOCENTI che hanno Tenuto Nel 2009/2010 un Occorrenza di un Ins. che NON Sia un UNITÀ LOGISTICA del Corso di Studi con ID=4 ma che non hanno MAI Tenuto un Modulo dell'Insegnamento di Programmazione Nel medesimo Corso di Studi

**Primo Caso (Query Annidate)**

```
SELECT DISTINCT P.id, P.nome, P.Cognome, P.Telef.
FROM Persona JOIN Docenza ON (...)
              JOIN InsErogato ON (...)
WHERE IE.aa = '2009/2010' AND
      IE.id_corsostudi = 4 AND
      IE.modulo > 0 AND
      P.Id NOT IN (SELECT id FROM Docenza
```



JOIN InsErogato

JOIN Insegn

WHERE I.nome = Programmazione AND  
IE.idCorsoStudi = 4 AND  
IE.modulo > 0

NB! Non c'è DATA BINDING Quindi i nomi possono essere diversi (altrimenti ~~non~~ sarebbe possibili)

## SECONDO CASO (Operatori Insiemistici)

Query di Prima:

SELECT DISTINCT P.id, P.nome, P.Cognome, P.Telef.  
FROM Persona JOIN Docenza ON (...)  
JOIN InsErogato ON (...)  
WHERE IE.aa = '2009/2010' AND  
IE.idCorsoStudi = 4 AND  
IE.modulo > 0 AND

P.id NOT IN (SELECT id FROM Docenza  
JOIN InsErogato  
JOIN Insegn  
WHERE I.nome = Programmazione AND  
IE.idCorsoStudi = 4 AND  
IE.modulo > 0

① EXCEPT ②

Q6:

```
SELECT f.Nome, COUNT(NameUnita) AS Num_Unita, SUM(crediti)
FROM InsErogato IE JOIN facolta f
WHERE IE.aa = '2010/2011' AND
      IE.modulo < 0
GROUP BY f.Nome
```

Prima Eseguito WHERE e poi GROUP BY (HAVING Seleziona i Gruppi)

COUNT → Conta le Righe

SUM → Somma Valori