

INTERROGAZIONI INSIEMISTICHE:

OPERATORI INSIEMISTICI per UNIRE Risultato di più OPERAZIONI

Si Applicano al livello più Esterno della QUERY.

■ **Union** → Aggiunge il RISULTATO di Q2 a Q1 (Eliminando duplicati)

■ **INTERSECT** → Tuple Comuni in Q1 ed in Q2

■ **EXCEPT** → Tuple che ci Sono Solo in Q1 (DIFERENZA)

Tipo di uso ⇒ <QUERY_1> [*] <QUERY_2>

Vengono Eseguite le 2 Query Singolarmente, creando così 2 Insiemi di TUPLE e poi Si uniscono Q1 e Q2 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 50char e 40char sono compatibili

A parità di Posizione, il TIPO deve Essere Compatibile

SQL <> ALGEBRA RELAZIONALE

lavora con INSIEMI DI TUPLE e non ci sono MAI duplicati

→ Serve DISTINCT... ma Se usa gli OPERATORI

Insiemistici vengono ELIMINATI i duplicati se non si Specifica le parole chiave ALL.

Query che Visualizzi i Nomi degli Insegnamenti e dei CORSI di Laurea che non Iniziano con le Lettere 'A'

SELECT NomeIns

From Insegn

①

WHERE NomeIns NOT LIKE 'A%'

SELECT Nome

From CorsoStudi

②

WHERE Nome NOT LIKE 'A%'

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 NameIns FROM Insegn ①

SELECT Name FROM CorsoStudi ②

① INTERSECT <ALL> ②

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

SELECT NameIns FROM Insegn ①

SELECT Name 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 Avvere delle DIPENDENZE CIRCOLARI

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

Se VOGLIAMO Sovrascrivere
Quelli delle QUERY

Quando mi Scrollo 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 (nameIns, codiceIns) Presi dalle Tabelle INSEGN

Query Sarebbe

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

Vista Corrispondente:

CREATE VIEW InserogatoCompleto AS

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

Su PGAdmin4 c'è una Sezione per le Viste

Da TERMINALE:

1D → Turtle lc 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 ^a CASCATA

NumIns ≠ in un Corso Studi:

Group By CS. Name

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 Aurei 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 SOTTRAZIONE.

```
SELECT DISTINCT P1.id, P1.Nome, P1.Cognome
FROM Persona P1 JOIN Docenza D1 JOIN insErogato IE1
    JOIN Inseg I1
WHERE IE1.annoacc = '2010/2011' AND P1.ID NOT IN (
    SELECT D2.id_Persona
    FROM Docenza 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 ILIKE '%...%')
```

QUERY 5: Trovare Nome, Cognome, Telefono dei DOCENTI che hanno Tenuto nel 2009/2020 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 Annidata)

```
SELECT DISTINCT P.id, P.nome, P.Cognome, P.Telef.
  From Persona JOIN Docenza ON (...)
                JOIN InsErogato ON (...)
 WHERE IE.ae = '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 possibile)

SECONDO CASO (Operatori Insiemistici)

Query di Prima:

SELECT DISTINCT P.id, P.nome, P.Cognome, P.Telcf.
FROM Persona JOIN Docenza ON (...)
JOIN InsErogato ON (...)

1

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

2

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

1 EXCEPT 2

Q6:

```
SELECT F.Nome, COUNT(NomeUnita) AS Num_Unita, SUM(Crediti)
FROM Insegnato IE JOIN Facoltà 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