



STRUCTURED QUERY LANGUAGE (SQL) E INTERROGAZIONI

Patrizio Dazzi
a.a. 2017 - 2018

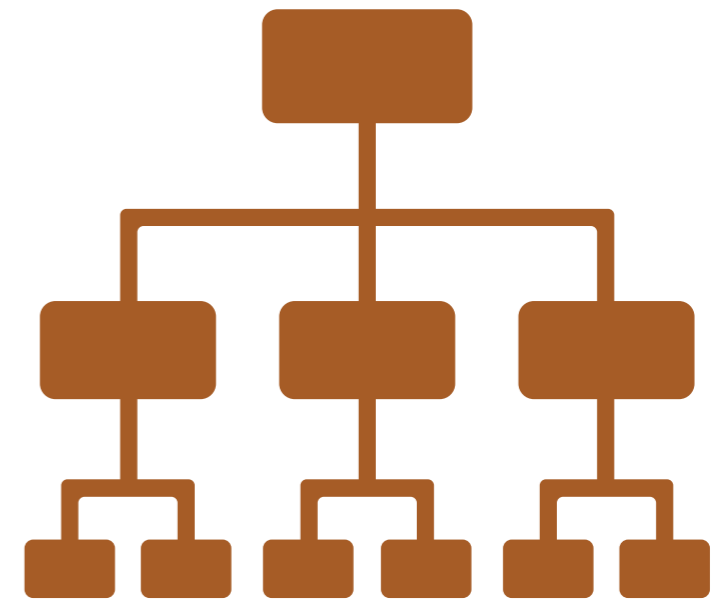
COMUNICAZIONI

- Oggi andiamo di corsa... take care.
- Prove in itinere
 - Primo compitino il 30 o il 31 ottobre (disponibilità aule)
 - Secondo compitino il 19 dicembre
- Homeworks
 - Domani il secondo Homework (algebra relazionale)
 - Consegna del primo a breve
 - voti nell'intervallo $[0,10]$
 - valido per chi ha preso più di 5
 - si sommano i voti dei vari homework
 - bonus sul voto finale



PICCOLO RIASSUNTO DELLA PUNTATA PRECEDENTE

- Giunzione
 - Operatore diadico
 - Due tabelle
 - Attributo su cui si giungono le tabelle
- Esercitazioni
 - Libri e Autori
 - Formula 1
 - Studi Cinematografici



**SQL! CHI ERA
COSTUI ?**



SQL: CONCETTI FONDAMENTALI

- Operazioni di diverso tipo
 - Creazione ed eliminazione di basi di dati
 - Creazione ed eliminazione di tabelle
 - Inserimenti e cancellazione di ennuple

- Realizza gli operatori dell'algebra relazionale
 - vediamo il mapping

COLTELLINO SVIZZERO

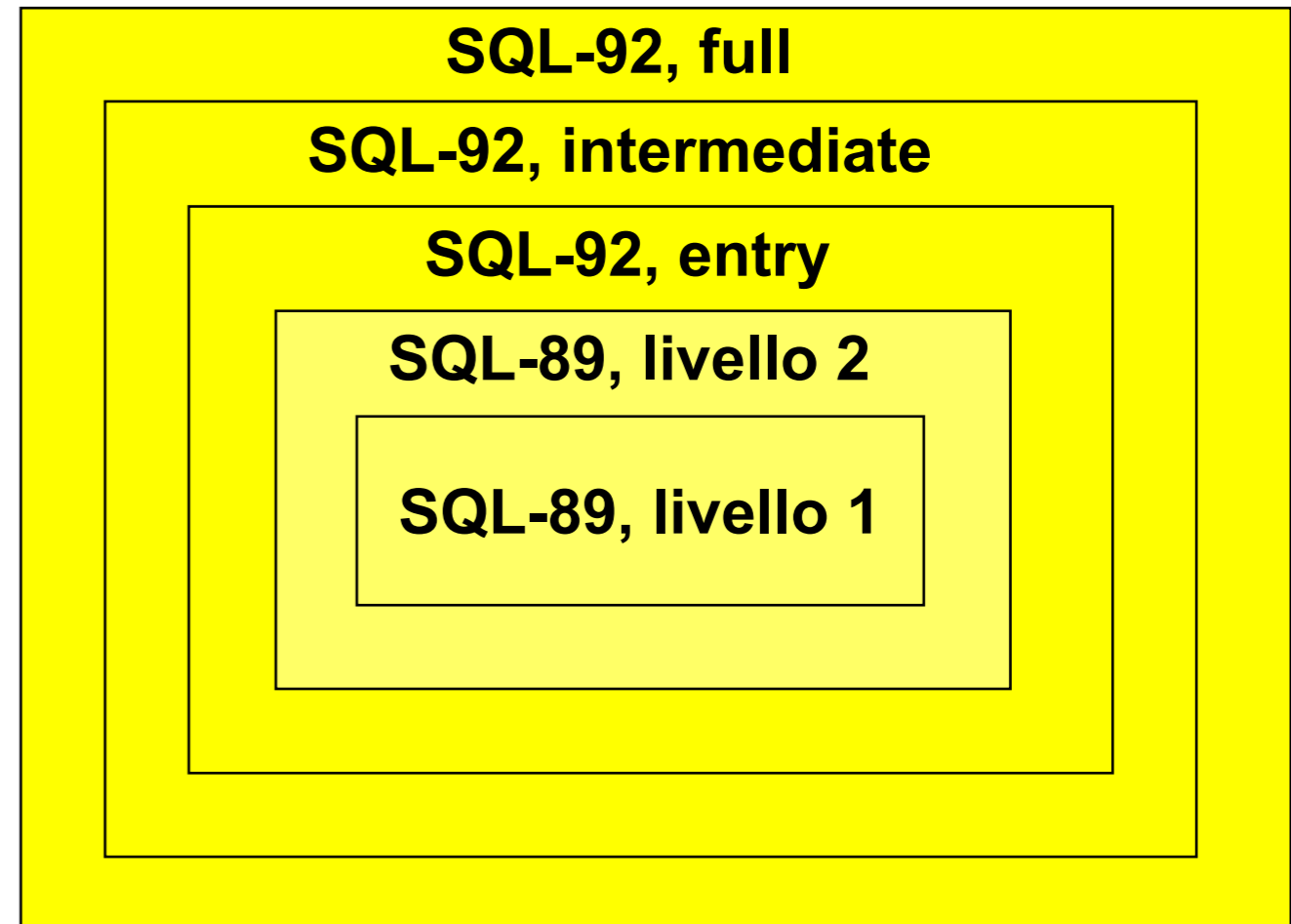
- **SQL (“Structured Query Language”)**
 - linguaggio per l’interazione con il DBMS
 - tutte le operazioni vengono specificate in SQL
- **DDL (“Data Definition Language”)**
 - creazione degli oggetti dello schema
- **DCL (“Data Control Language”)**
 - controllo degli utenti e delle autorizzazioni
- **DML (“Data Manipulation Language”)**
 - manipolazione dell’istanza della base di dati (interrogazioni e aggiornamenti)

STORIA DELLO STANDARD

- Prime implementazioni
 - IBM System/R 1979 (SEQUEL)
- Primi prodotti commerciali
 - IBM SQL/DS, Oracle 1981
- SQL-86
 - prima versione dello standard, basata sul dialetto IBM

STORIA DELLO STANDARD

- SQL-89 (SQL-1)
 - vincoli di integrità
 - livello1 e livello2
- SQL-92 (SQL-2)
 - entry
 - intermediate
 - full

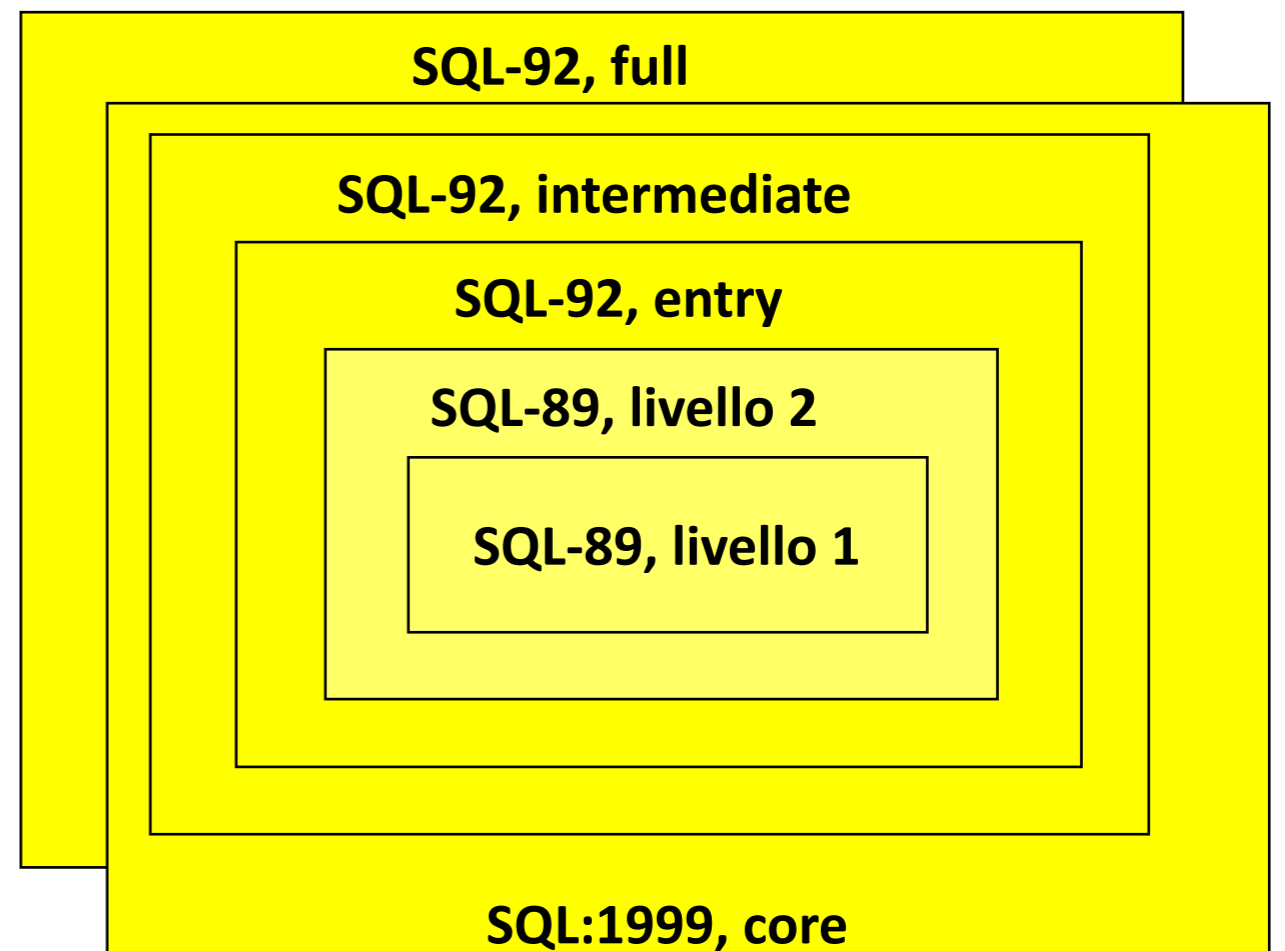


STORIA DELLO STANDARD

- Standard collegati
- SQL/CLI
 - “Call Level Interface” (ODBC), 1995
- SQL/PSM
 - “Persistent Storage Modules”, 1997
- SQL/OLB
 - “Object Language Bindings”, 1998

STORIA DELLO STANDARD

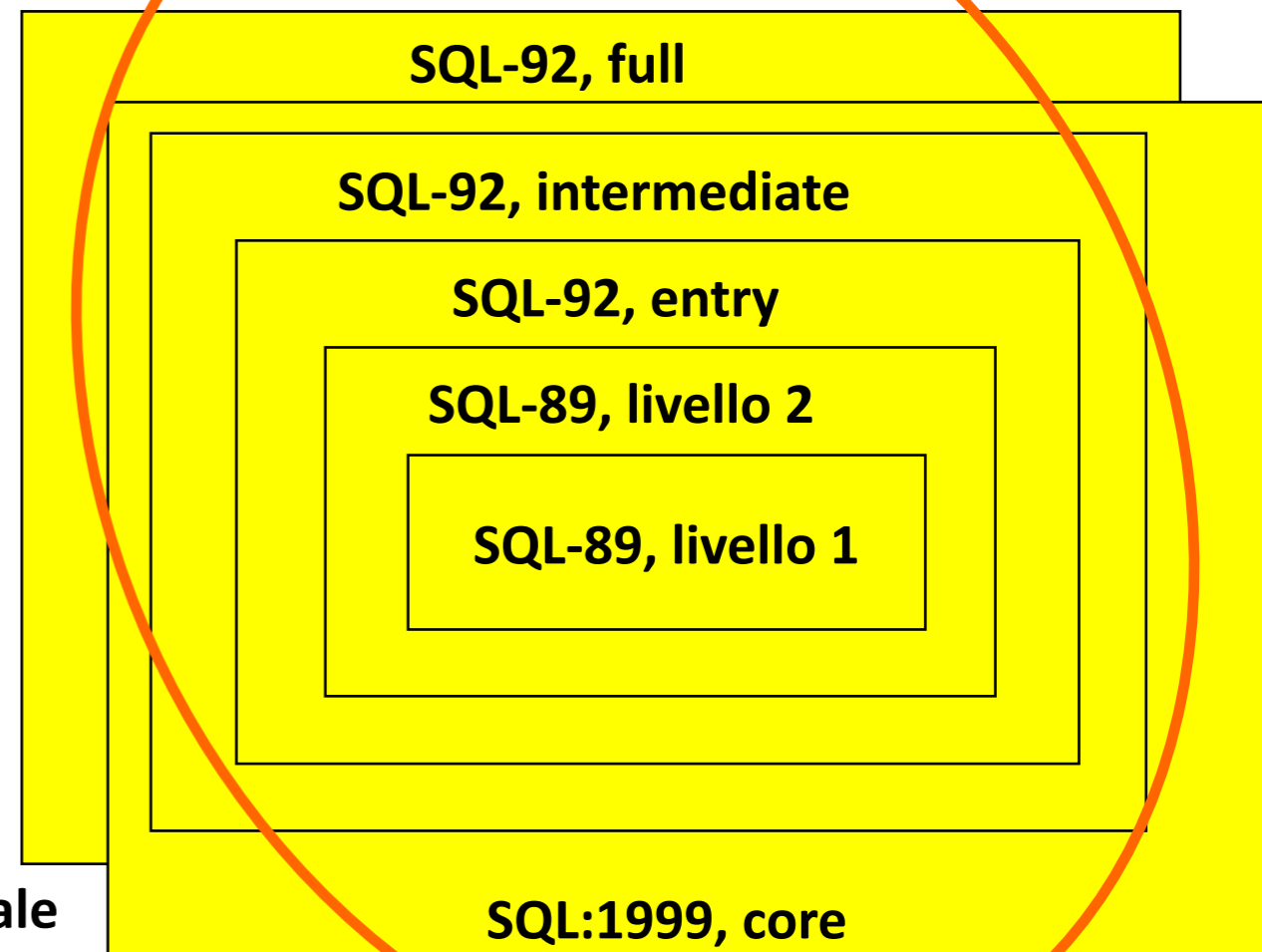
- SQL:1999 (SQL-3)
 - estensioni “object-relational”
 - core: tutto SQL-92 entry, (quasi) tutto SQL-92 intermediate, parte di SQL-92 full
 - packages
- Attualmente:
 - lavori su SQL:200x



STORIA DELLO STANDARD

- Attualmente
 - SQL-92 intermediate, con funzionalità di full
 - nessuna implementazione completa di SQL-92 full
 - parte di SQL:1999
- Ci concentriamo su
 - SQL-92, intermediate

tipico DBMS commerciale



ALGEBRA RELAZIONALE E SQL



ATTENTI A QUEI DUE...

- Bella l'algebra relazionale... vero ?
 - Come la “diamo in pasto” ad un calcolatore ?
 - Usiamo SQL!!!
- SQL come Trait d'union tra algebra relazionale e calcolatore
 - Vediamo come esprimere i concetti visti sino ad oggi in SQL
 - Struttura, sintassi, differenze

INTERROGAZIONI

- Istruzione del DML

- SELECT

- una o più sottointerrogazioni correlate da operatori insiemistici

- Filosofia

- si specificano gli operatori da applicare, non l'ordine in cui devono essere applicati

- l'ottimizzatore sceglie la strategia ottima

- Sottointerrogazione:

- selezioni

- proiezioni (con funzioni aggregative)

- eliminazione di duplicati (DISTINCT)

- ridenominazioni

- ordinamenti finali (ORDER BY)

**SELECT
FROM
WHERE**



INTERROGAZIONI

- Interrogazioni SQL

- una o più sottointerrogazioni correlate da operatori insiemistici

- Nucleo della SELECT

- **SELECT**: proiezioni, ridenominazioni, distinct

- **FROM**: prodotti cartesiani o join, alias

- [**WHERE**]: selezioni

- Clausole aggiuntive

- [**ORDER BY**]: ordinamenti

INTERROGAZIONE

SELECT Lista Attributi

FROM Lista Tabelle

[**WHERE** Condizione]

Intuitivamente:

- Seleziona tra le righe delle tabelle elencate nel **FROM** quelle che soddisfano le condizioni espresse nel **WHERE** e estrae solo gli attributi indicati nel **SELECT**

CLAUSOLA FROM

► Funzione

- Permette di indicare la tabella o l'insieme delle tabelle su cui eseguire l'interrogazione.

Situazione Semplice: una tabella

- **FROM** Studenti

oppure

- **FROM** Studenti **AS** S

SELEZIONE: CLAUSOLA WHERE

- Funzione

- **Seleziona solo le ennuple che soddisfano la condizione**

- WHERE <condizione>

- <condizione>

- condizioni di selezione, connettivi booleani

Esempio

WHERE S.Cognome='Rossi' **AND** S.Anno>1

SELEZIONE: CLAUSOLA WHERE

➤ Condizioni di selezione

- condizioni sui valori degli attributi
- operatori di confronto =, >, <, >=, <=, <>
- espressioni con operatori e funzioni
- connettivi booleani AND, OR, NOT

➤ Operatori speciali

- IS NULL, IS NOT NULL
- LIKE

ESEMPIO DI SELEZIONE

- Estrarre le informazioni degli studenti che si chiamano Rossi

SELECT *

FROM Studenti **AS** S

WHERE S.Cognome='Rossi'

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
587614	Rossi	Luca	laurea sp.	2	FT
276545	Rossi	Maria	laurea tr.	1	null

ESEMPIO DI SELEZIONE

➤ Estrarre il cognome degli studenti

SELECT *

FROM Studenti **AS** S

WHERE S.Cognome='Rossi' **AND** S.Anno>1

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
587614	Rossi	Luca	laurea sp.	2	FT

CONDIZIONE COMPLESSA

```
SELECT *  
FROM Studenti  
WHERE ciclo = 'laurea tr'  
AND(anno = 1 OR anno = 3)
```

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
200768	Verdi	Fabio	laurea tr.	3	FT

CONDIZIONE "LIKE"

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

```
SELECT *  
FROM Studenti  
WHERE nome LIKE 'M_r%'
```

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
937653	Bruni	Mario	laurea sp.	1	CV

OPERATORE LIKE

- Operatore LIKE
 - corrispondenza “parziale” tra valori testuali
- “Pattern”
 - sequenza di caratteri e simboli speciali
 - %: una sequenza di 0 o più caratteri
 - _ : un carattere qualsiasi
 - corrisponde ad un insieme di stringhe

OPERATORE LIKE

➤ Esempi:

- 'B%i': {'Bianchi', 'Belli', 'Brutti', 'Bi', ...}
- 'p___a': {'palla', 'pasta', 'pista', ...}
- 'A_t%': {'Antonio', 'Artrite', ...}

➤ Condizioni

- <attributo di tipo testo> LIKE <pattern>
- vera se il valore dell'attributo appartiene all'insieme di stringhe corrispondenti

GESTIONE VALORI NULL

- Studenti che non hanno un relatore

```
SELECT *  
FROM Studenti  
WHERE relatore is NULL
```

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null

SELEZIONE E VALORI NULLI

- Attenzione ai valori nulli
 - le condizioni sono sempre false in presenza di valori nulli
 - es: `facolta <> 'Ingegneria'` solo le ennuple per cui la facoltà è non nulla e diversa da ing.
 - condizioni speciali: **IS NULL, IS NOT NULL**
 - es: `facolta <> 'Ingegneria' OR facolta IS NULL` tutte le ennuple in cui il valore non è ing.

ESEMPIO: "PROFESSORI CHE NON SONO DI INGEGNERIA"

Professori

<u>cod</u>	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

```
SELECT *  
FROM Professori  
WHERE facolta <> 'Ingegneria'
```

<u>cod</u>	cognome	nome	qualifica	facolta
CV	Vieri	Christian	associato	Scienze

```
SELECT *  
FROM Professori  
WHERE facolta <> 'Ingegneria'  
OR facolta IS NULL
```

<u>cod</u>	cognome	nome	qualifica	facolta
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

PROIEZIONE: CLAUSOLA SELECT

Funzione

- ▶ estrarre alcune delle colonne di una tabella (PROIEZIONE)
- ▶ `SELECT [DISTINCT] <attributi> | *`
 - ▶ <attributi>
 - ▶ lista di nomi di attributo
 - ▶ usare **AS** per le ridenominazione

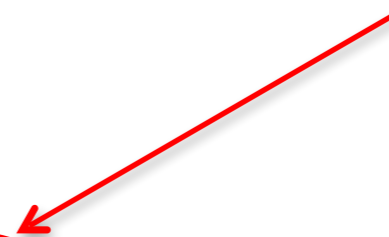
Esempio

`SELECT *` (SENZA PROIEZIONE)

Oppure

`SELECT S.Cognome, S.nome AS NomeStud` (PROIEZIONE)

Ridenominazione



PROIEZIONE: CLAUSOLA SELECT

➤ Schema del risultato

- attributi dello schema originale su cui si effettua la proiezione

➤ Istanza del risultato

- restrizione (“proiezione”) delle ennuple originali agli attributi specificati

➤ **ATTENZIONE**

- se nel risultato non sopravvivono chiavi dello schema originale possono esserci duplicati

ESEMPIO DI PROIEZIONE

- Estrarre il cognome degli studenti

SELECT Cognome

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

Cognome
Rossi
Neri
Verdi
Rossi
Bruni

PROIEZIONE E DUPLICATI

- La proiezione può generare duplicati
 - nel caso nel risultato non sopravvivano chiavi
- Filosofia dell'SQL (e quindi dell'algebra)
 - l'utente può scegliere se eliminare o meno i duplicati
- Costrutto DISTINCT
 - elimina da una relazione R i duplicati

DISTINCT

- Il risultato di una selezione può contenere delle ennuple duplicate
- **Esempio**
 - Estrarre i tipi di “ciclo” degli studenti

```
SELECT S.ciclo  
FROM Studenti AS S
```

Ciclo
laurea tr.
laurea tr.
laurea tr.
laurea sp.
laurea sp.

```
SELECT DISTINCT S.ciclo  
FROM Studenti AS S
```

Ciclo
laurea tr.
laurea sp.

ESEMPIO: "COGNOMI E ANNI DI CORSO DEGLI STUDENTI"

Studenti

<u>matr</u>	cognome	nome	ciclo	anno	relatore
111	Rossi	Mario	laurea tr.	1	null
222	Neri	Paolo	laurea tr.	2	null
333	Rossi	Maria	laurea tr.	1	null
444	Pinco	Palla	laurea tr.	3	FT
77777	Bruno	Pasquale	laurea sp.	1	FT
88888	Pinco	Pietro	laurea sp.	1	CV

```
SELECT S.cognome, S.anno  
FROM Studenti AS S
```

```
SELECT DISTINCT S.cognome, S.anno  
FROM Studenti AS S
```

cognome	anno
Rossi	1
Neri	2
Rossi	1
Pinco	3
Bruno	1
Pinco	1

cognome	anno
Rossi	1
Neri	2
Pinco	3
Bruno	1
Pinco	1

ORDINAMENTO

- E' possibile gestire anche l'ordinamento
- Costrutto **ORDER BY**
 - sintassi: **ORDER BY** *lista attributi*
 - semantica: riordina le ennuple di una relazione R utilizzando i valori degli attributi specificati come chiavi di ordinamento (dal primo in avanti) in ordine crescente/descrescente

CLAUSOLA ORDER BY

- Funzione

- Ordina le tuple in modo discendente o ascendente

- ORDER BY <attributi>

- <attributi>

- lista di attributi

- <attributo> {ASC | DESC}

Esempio

ORDER BY S.Cognome ASC, S.Nome DESC

ORDER BY S.Cognome DESC, S.Nome DESC

ORDER BY S.Cognome, S.Nome (default ASC)

- Semantica

- Ordinamento delle tuple

ESEMPIO DI ORDER BY

- Estrarre le informazioni degli studenti che si chiamano Rossi

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

SELECT *

FROM Studenti **AS** S

WHERE S.Cognome='Rossi'

ORDER BY S.Cognome, S.Nome **DESC**

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
587614	Rossi	Luca	laurea sp.	2	FT

CLAUSOLA FROM: JOIN

- E' nella natura del modello relazionale frammentare i dati tra le tabelle
- molto spesso è necessario **correlare dati provenienti da tabelle diverse**
- è possibile utilizzare il **prodotto cartesiano**
- Sul prodotto cartesiano si applicano le condizioni del WHERE indicando quelle che esprimono il legame tra le tabelle per esprimere il JOIN

ESEMPIO DI JOIN

STUDENTI

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

PROFESSORI

Codice	Cognome	Nome	Qualifica	Facolta
FT	Monreale	Anna	Ricercatore	Informatica
CV	Tesconi	Maurizio	Ricercatore	Ingegneria

Matricola	Cognome	Nome	Ciclo	Anno	Relatore	Codice	Cognome	Nome	Qualifica	Facolta
276545	Rossi	Maria	laurea tr.	1	null	FT	Monreale	Anna	Ricercatore	Informatica
276545	Rossi	Maria	laurea tr.	1	null	CV	Tesconi	Maurizio	Ricercatore	Ingegneria
485745	Neri	Anna	laurea tr.	2	null	FT	Monreale	Anna	Ricercatore	Informatica
485745	Neri	Anna	laurea tr.	2	null	CV	Tesconi	Maurizio	Ricercatore	Ingegneria
200768	Verdi	Fabio	laurea tr.	3	FT	FT	Monreale	Anna	Ricercatore	Informatica
200768	Verdi	Fabio	laurea tr.	3	FT	CV	Tesconi	Maurizio	Ricercatore	Ingegneria
587614	Rossi	Luca	laurea sp.	2	FT	FT	Monreale	Anna	Ricercatore	Informatica
587614	Rossi	Luca	laurea sp.	2	FT	CV	Tesconi	Maurizio	Ricercatore	Ingegneria
937653	Bruni	Mario	laurea sp.	1	CV	FT	Monreale	Anna	Ricercatore	Informatica
937653	Bruni	Mario	laurea sp.	1	CV	CV	Tesconi	Maurizio	Ricercatore	Ingegneria

ESEMPIO DI JOIN

STUDENTI

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

PROFESSORI

Codice	Cognome	Nome	Qualifica	Facolta
FT	Monreale	Anna	Ricercatore	Informatica
CV	Tesconi	Maurizio	Ricercatore	Ingegneria

Matricola	Cognome	Nome	Ciclo	Anno	Relatore	Codice	Cognome	Nome	Qualifica	Facolta
276545	Rossi	Maria	laurea tr.	1	null	FT	Monreale	Anna	Ricercatore	Informatica
276545	Rossi	Maria	laurea tr.	1	null	CV	Tesconi	Maurizio	Ricercatore	Ingegneria
485745	Neri	Anna	laurea tr.	2	null	FT	Monreale	Anna	Ricercatore	Informatica
485745	Neri	Anna	laurea tr.	2	null	CV	Tesconi	Maurizio	Ricercatore	Ingegneria
200768	Verdi	Fabio	laurea tr.	3	FT	FT	Monreale	Anna	Ricercatore	Informatica
200768	Verdi	Fabio	laurea tr.	3	FT	CV	Tesconi	Maurizio	Ricercatore	Ingegneria
587614	Rossi	Luca	laurea sp.	2	FT	FT	Monreale	Anna	Ricercatore	Informatica
587614	Rossi	Luca	laurea sp.	2	FT	CV	Tesconi	Maurizio	Ricercatore	Ingegneria
937653	Bruni	Mario	laurea sp.	1	CV	FT	Monreale	Anna	Ricercatore	Informatica
937653	Bruni	Mario	laurea sp.	1	CV	CV	Tesconi	Maurizio	Ricercatore	Ingegneria

ESEMPIO DI JOIN

STUDENTI

Matricola	Cognome	Nome	Ciclo	Anno	Relatore
276545	Rossi	Maria	laurea tr.	1	null
485745	Neri	Anna	laurea tr.	2	null
200768	Verdi	Fabio	laurea tr.	3	FT
587614	Rossi	Luca	laurea sp.	2	FT
937653	Bruni	Mario	laurea sp.	1	CV

PROFESSORI

Codice	Cognome	Nome	Qualifica	Facolta
FT	Monreale	Anna	Ricercatore	Informatica
CV	Tesconi	Maurizio	Ricercatore	Ingegneria

Matricola	Cognome	Nome	Ciclo	Anno	Relatore	Codice	Cognome	Nome	Qualifica	Facolta
200768	Verdi	Fabio	laurea tr.	3	FT	FT	Monreale	Anna	Ricercatore	Informatica
587614	Rossi	Luca	laurea sp.	2	FT	FT	Monreale	Anna	Ricercatore	Informatica
937653	Bruni	Mario	laurea sp.	1	CV	CV	Tesconi	Maurizio	Ricercatore	Ingegneria

CLAUSOLA FROM

➤ Strategia a

FROM R, S

WHERE S.A=R.B

➤ Strategia b

FROM S **JOIN** R **ON** S.A=R.B

➤ Strategia c

FROM S **AS** p **JOIN** R **AS** f **ON** p.A=f.B

ALIAS



ALIAS

- Necessario quanto la stessa tabella può essere coinvolta più di una volta nello stesso join
- Esempio
 - “Cognomi e nomi degli studenti che all’esame di Programmazione hanno riportato un voto superiore a quello dei loro tutor”
- ATTENZIONE
 - si tratta di una interrogazione molto complessa

ALIAS

- In questo caso
 - devo confrontare il voto dello studente nell'esame di programmazione con quello del tutor
 - entrambe le informazioni vengono dalla tabella Studenti (che deve necessariamente essere usata due volte)
 - problema con i nomi: come distinguo l'attributo che corrisponde al voto dello studente da quello del tutor?

ALIAS

- Operatore di Alias per una Tabella
 - crea una copia di una tabella esistente
 - con un nome diverso (e quindi risolve il problema del nome degli attributi)
- Sintassi
 - R AS T
- Semantica
 - l'istanza di T è identica all'istanza di R
 - nello schema di T, l'attributo R.A assume il nome T.A

ALIAS

“Cognomi e nomi degli studenti che all’esame di Programmazione hanno riportato un voto superiore a quello dei loro tutor”

- Studenti, per i dati degli studenti
- Esami, per i dati sugli esami degli studenti
- Tutorato, per le relazioni tra studenti e tutor
- Esami di nuovo, per i dati sugli esami sostenuti dai tutor; è necessario un alias (Esami as EsamiTutor)

“STUDENTI, VOTI E TUTOR”

- Strategia
 - I Passo: tabella StudentiVoti, join tra Studenti ed Esami
 - II Passo: tabella StudentiVotiTutor, join tra StudentiVoti e Tutorato
 - III Passo: tabella StudentiVotiTutorVoti, join tra StudentiVotiTutor e Esami AS EsamiTutor

“STUDENTI, VOTI E TUTOR”

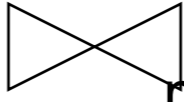
- I Passo: join tra Studenti ed Esami

StudentiVoti= Studenti  Esami
matr=studente

```
TABLE StudentiVoti (  
  Studenti.matr integer,  
  Studenti.cognome text,  
  Studenti.nome text,  
  Studenti.ciclo text,  
  Studenti.anno integer,  
  Studenti.relatore text,  
  Esami.studente integer  
  Esami.corso text  
  Esami.voto integer,  
  Esami.lode integer);
```

“STUDENTI, VOTI E TUTOR”

- Il Passo: join con Tutorato

StudentiVotiTutor= StudentiVoti  Tutorato
matr=Tutorato.studente

```
TABLE StudentiVoti (  
  Studenti.matr integer,  
  Studenti.cognome text,  
  Studenti.nome text,  
  Studenti.ciclo text,  
  Studenti.anno integer,  
  Studenti.relatore text,  
  Esami.studente integer  
  Esami.corso text  
  Esami.voto integer,  
  Esami.lode integer,  
  Tutorato.studente integer,  
  Tutorato.tutor integer);
```

“STUDENTI, VOTI E TUTOR”

- III Passo: join con Esami AS EsamiTutor

StudentiVotiTutorEsamitutor = StudentiVotiTutor

 Tutorato.tutor=EsamiTutor.studente (Esami AS EsamiTutor)

```
TABLE StudentiVotiTutorEsamitutor (  
  Studenti.matr integer,  
  Studenti.cognome text,  
  Studenti.nome text,  
  Studenti.ciclo text,  
  Studenti.anno integer,  
  Studenti.relatore text,  
  Esami.studente integer  
  Esami.corso text  
  Esami.voto integer,  
  Esami.lode integer,  
  Tutorato.studente integer,  
  Tutorato.tutor integer,  
  EsamiTutor.studente integer  
  EsamiTutor.corso text  
  EsamiTutor.voto integer,  
  EsamiTutor.lode integer);
```

“STUDENTI, VOTI E TUTOR”

StudentiVotiTutorEsamitutor

Stud.matr	Studenti. cognome	Esami. studente	Esami. corso	Esami. voto	Tutorato. studente	Tutorato. tutor	ET. studente	ET. Corso	ET. voto
111	Rossi	111	PR1	27	111	77777	77777	PR1	21
111	Rossi	111	INFT	24	111	77777	77777	PR1	21
222	Neri	222	ASD	30	222	77777	77777	PR1	21
111	Rossi	111	PR1	27	111	77777	77777	ASD	20
111	Rossi	111	INFT	24	111	77777	77777	ASD	20
222	Neri	222	ASD	30	222	77777	77777	ASD	20

tutte le possibili coppie fatte di
un esame di uno studente ed un esame del suo tutor

“STUDENTI, VOTI E TUTOR”

➤ Selezioni e proiezioni finali

Risultato =

```
 $\Pi_{\text{cognome, nome}} \left( \sigma_{\text{Esami.corso}='Pr1' \text{ AND } \text{EsamiTutor.corso}='Pr1' \text{ AND } \text{Esami.voto} > \text{EsamiTutor.voto}} \left( \text{StudentiVotiTutorEsamitutor} \right) \right)$ 
```

```
TABLE Risultato (  
  Studenti.cognome varchar(20),  
  Studenti.nome varchar(20));
```

“STUDENTI, VOTI E TUTOR”

► Riassumendo

Risultato = Π cognome, nome (

σ Esami.corso='Pr1' AND

EsamiTutor.corso='Pr1' AND

Esami.voto > EsamiTutor.voto (

Studenti  matr=studente Esami

 matr=Tutorato.studente Tutorato

 Tutorato.tutor=EsamiTutor.studente (Esami AS EsamiTutor)

)

)

SOLUZIONE SQL

SELECT Cognome, nome

FROM STUDENTI

JOIN ESAMI **ON** matr=studente

JOIN Tutorato **ON** Tutorato.studente=matr

JOIN ESAMI **AS** EsamiTutor **ON**

Tutorato.tutor=EsamiTutor.studente

WHERE Esami.corso='Pr1' **AND**

EsamiTutor.corso='Pr1' **AND**

Esami.voto > EsamiTutor.voto

OPERATORI INSIEMISTICI



OPERATORI INSIEMISTICI

➤ Operatori binari

➤ **Union:** $R \cup S$

➤ **Intersect:** $R \cap S$

➤ **EXCEPT:** $R - S$

OPERATORI INSIEMISTICI

- Si applicano solo in alcuni casi
 - le tabelle R ed S devono avere lo stesso numero di attributi
- **Associazione posizionale**
 - gli attributi devono avere ordinatamente lo stesso tipo
- **Schema del risultato**
 - eredita i nomi degli attributi dalla prima tabella
- **Attenzione**
dal risultato degli operatori insiemistici vengono eliminati eventuali duplicati

UNIONE

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Specialisti

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cup Specialisti

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45
9297	Neri	33

```
SELECT *  
FROM Laureati  
  
UNION  
  
SELECT *  
FROM Specialisti
```

INTERSEZIONE

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Specialisti

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cap Specialisti

Matricola	Nome	Età
7432	Neri	54
9824	Verdi	45

SELECT *

FROM Laureati

INTERSECT

SELECT *

FROM Specialisti

INTERSEZIONE

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

```
SELECT *  
FROM Laureati
```

```
INTERSECT
```

```
SELECT *  
FROM Specialisti
```

Specialisti

Matricola	Nome	Età
9297	Neri	trenta
7432	Neri	venti
9824	Verdi	quaranta

```
SELECT matricola, nome  
FROM Laureati
```

```
INTERSECT
```

```
SELECT matricola, nome  
FROM Specialisti
```

INTERSEZIONE IN

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Specialisti

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati \cap Specialisti

Matricola	Nome	Età
7432	Neri	54
9824	Verdi	45

```
SELECT *  
FROM Laureati AS L  
WHERE L.Matricola  
IN (  
SELECT S.Matricola  
FROM Specialisti AS S  
)
```

DIFFERENZA

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Specialisti

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati – Specialisti

Matricola	Nome	Età
7274	Rossi	42

```
SELECT *  
FROM Laureati
```

```
EXCEPT
```

```
SELECT *  
FROM Specialisti
```

DIFFERENZA NOT IN

Laureati

Matricola	Nome	Età
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Specialisti

Matricola	Nome	Età
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Laureati – Specialisti

Matricola	Nome	Età
7274	Rossi	42

```
SELECT *  
FROM Laureati AS L  
WHERE L.matricola  
NOT IN(  
SELECT S.matricola  
FROM Specialisti  
)
```


FINE DELLA LEZIONE