

SQL

SQL: "storia,,

- Sviluppato nella metà degli anni settanta (1974) presso il laboratorio di ricerca IBM di S.Josè
- Dal 1983 ca. "standard di fatto"
- È il linguaggio di riferimento per l'interrogazione di DBMS relazionali

SQL

- Structured Query Language
 - contiene sia il DDL che DML
- DDL:
 - CREATE TABLE, ALTER TABLE, DROP TABLE, constraint, ...
- DML:
 - Estrazione dati:
 - SELECT
 - Aggiornamento dati:
 - INSERT, DELETE, UPDATE

SQL Conformance

- Diverse standardizzazioni ANSI/ISO sono state proposte:
 - SQL-86, SQL-89, SQL-92, SQL:1999, SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016
- **PostgreSQL-12** supports most of the major features of SQL:2016. Out of 179 mandatory features required for full Core conformance, PostgreSQL conforms to at least 160. In addition, there is a long list of supported optional features. It might be worth noting that at the time of writing, no current version of any database management system claims full conformance to Core SQL:2016.

Dipartimento

<u>Nome</u>	<u>Indirizzo</u>	<u>Città</u>
Amministrazione	Via Tito Livio, 27	Milano
Produzione	P.Le Lavater, 3	Torino
Distribuzione	Via Segre, 9	Roma
Direzione	Via Tito Livio, 27	Milano
Ricerca	Via Morone, 6	Milano

Impiegato

<u>Nome</u>	<u>Cognome</u>	<u>Dipart</u>	<u>Ufficio</u>	<u>Stipendio</u>	<u>Città</u>
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Istruzione SELECT

- Nella sua forma più semplice l'estrazione di dati avviene attraverso il seguente comando:

SELECT ListaAttributi
FROM Tabella

Che ha come effetto quello di estrarre da una tabella i valori relativi ad un sottoinsieme degli attributi della tabella stessa

Istruzione SELECT

Impiegato

<u>Nome</u>	<u>Cognome</u>	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Select nome, cognome, dipart
From impiegato

Nome	Cognome	Dipart
Mario	Rossi	Amministrazione
Carlo	Bianchi	Produzione
Giuseppe	Verdi	Amministrazione
Franco	Neri	Distribuzione
Carlo	Rossi	Direzione
Lorenzo	Lanzi	Direzione
Paola	Borroni	Amministrazione
Marco	Franco	Produzione

Istruzione SELECT

- Tale operazione è equivalente alla **PROIEZIONE** nell'algebra relazionale
- L'effetto dell'istruzione

Select nome, cognome, dipart
From impiegato

è quello di proiettare sugli attributi nome, cognome e dipart il contenuto della tabella impiegato

- La lista di attributi che segue la keyword SELECT è detta **target list**

Istruzione SELECT

- Se al posto della lista di attributi compare il simbolo * il risultato contiene i dati relativi a tutti gli attributi della tabella:

Select * from impiegato

presenta il contenuto dell'intera tabella
impiegato

Case sensitivity

- In PostgreSQL unquoted names are case-insensitive. The following statements are equivalent:
 - **SELECT * FROM impiegato**
 - **SELECT * FROM IMPIEGATO**
- However, quoted names are case-sensitive. The following statements are NOT equivalent:
 - **SELECT * FROM "impiegato"**
 - **SELECT * FROM "Impiegato"**

Istruzione SELECT

- L'istruzione select può essere estesa in modo da effettuare una **SELEZIONE** delle righe che devono comparire nel risultato
- Questo richiede la specifica di una condizione di selezione:
 - Una formula booleana da calcolarsi (riga per riga) in base al valore di alcuni attributi
- Il risultato conterrà solo quelle righe per le quali la condizione di selezione dà valore vero

Istruzione SELECT

Select nome, cognome

From impiegato

Where cognome='Rossi'

La condizione di selezione specificata dopo la keyword WHERE prende il nome di *clausola where*

Impiegato

<u>Nome</u>	<u>Cognome</u>	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Nome	Cognome
Mario	Rossi
Carlo	Rossi

Istruzione SELECT

- Estrarre nome e cognome degli impiegati di cognome "Rossi" che lavorano nell'ufficio 10

Select nome, cognome

From impiegato

Where (cognome='Rossi') AND (ufficio=10)

Nome	Cognome
Mario	Rossi

**La proiezione è fatta
dopo la selezione
(altrimenti non
potrebbe operare
ufficio=10)**

Impiegato

<u>Nome</u>	<u>Cognome</u>	<u>Dipart</u>	<u>Ufficio</u>	<u>Stipendio</u>	<u>Città</u>
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Istruzione SELECT

- Estrarre nome e cognome degli impiegati di cognome “Rossi” con stipendio compreso tra 60 e 100

Select nome, cognome

From impiegato

Where (cognome='Rossi') AND stipendio between 60 and 100

Nome	Cognome
Carlo	Rossi

Impiegato

<u>Nome</u>	<u>Cognome</u>	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Interrogazione 2:

- Estrarre tutte le informazioni presenti nella tabella Impiegato per gli impiegati di cognome “Rossi”.

```
select *  
from Impiegato  
where Cognome='Rossi'
```

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Direzione	20	80	Milano
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

SELECT SQL

- Nella target list possono comparire generiche **espressioni aritmetiche** sul valore degli attributi di ciascuna riga selezionata.
- Sia gli attributi che le tabelle possono essere rinominati per mezzo di **Alias** con il costrutto **as**.

Interrogazione 3:

- Estrarre lo stipendio mensile dell'impiegato di Cognome "Bianchi":

```
select Stipendio/12 as StipendioMensile  
from Impiegato  
where Cognome='Bianchi'
```

StipendioMensile
3

Impiegato

<u>Nome</u>	<u>Cognome</u>	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Interrogazione 3:

- Estrarre nome, cognome e stipendio mensile degli impiegati che hanno stipendio mensile maggiore di 5:

```
select Nome, Cognome, Stipendio/12 as StipendioMensile  
from Impiegato  
where (Stipendio/12) > 5
```

Impiegato

Nome	Cognome	StipendioMensile
Carlo	Rossi	6.66
Lorenzo	Lanzi	6.08

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

SELECT SQL

- La clausola `where` ammette come argomento un'espressione booleana costruita combinando predicati atomici con gli operatori `and`, `or` e `not` .
- La sintassi assegna la precedenza all'operatore `not` ma non definisce una relazione di precedenza tra gli operatori `and` e `or`.

Interrogazione 6:

- Estrarre nome e cognome degli impiegati che lavorano nell'ufficio 20 del dipartimento Amministrazione:

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 6:

- Estrarre nome e cognome degli impiegati che lavorano nell'ufficio 20 del dipartimento Amministrazione:

```
select Nome, Cognome  
from Impiegato  
where (Dipart='Amministrazione') and (Ufficio=20)
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 7:

- Estrarre nome e cognome degli impiegati che lavorano nel dipartimento Amministrazione o in quello Produzione.

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 7:

- Estrarre nome e cognome degli impiegati che lavorano nel dipartimento Amministrazione o in quello Produzione.

```
select Nome, Cognome  
from Impiegato  
where (Dipart='Amministrazione') or  
(Dipart='Produzione')
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 8:

- Estrarre nome degli impiegati di cognome Rossi che lavorano nel dipartimento Amministrazione o in quello Produzione.

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 8:

- Estrarre nome degli impiegati di cognome Rossi che lavorano nel dipartimento Amministrazione o in quello Produzione.

```
select Nome  
from Impiegato  
where (Cognome='Rossi') and  
((Dipart='Amministrazione') or  
(Dipart='Produzione'))
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

SELECT SQL

- Oltre ai normali operatori di confronto relazionali (=, <, >, <=, >=, <>), SQL mette a disposizione l'operatore **like** per il confronto di stringhe.
- Questo supporta il confronto tra stringhe in cui compaiono due caratteri speciali:
 - **_** (?) può rappresentare **un qualsiasi carattere**
 - **%** (*) può rappresentare una stringa di **una qualsiasi sequenza di caratteri** (eventualmente anche nulla)

Interrogazione 9:

- Estrarre nome e cognome degli impiegati il cui cognome inizia con la "C":

Select nome, cognome

From impiegato

Where cognome like 'C%'

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 9:

- Estrarre nome e cognome degli impiegati il cui cognome ha una “o” in seconda posizione e finisce con “i”.

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

JOIN

- Tipicamente, le informazioni memorizzate in un database sono organizzate su più tabelle
- I legami tra righe di tabelle diverse sono mantenuti attraverso codici attraverso cui correlare le righe
- L'estrazione di informazioni contenute su più tabelle richiede pertanto la specifica di:
 - Quali siano le tabelle da prendere in considerazione
 - Quale sia il criterio per correlare le righe delle diverse tabelle

JOIN

- Questa operazione prende il nome di **JOIN** (o più precisamente theta-join) tra le tabelle e viene eseguita attraverso due passi:
 - Prodotto cartesiano tra le due tabelle
 - Concatenare ogni riga della prima tabella con ogni riga della seconda tabella
 - Selezione delle righe conformi al criterio espresso nel predicato booleano

JOIN

- Supponiamo di voler estrarre i dati relativi a ciascun impiegato ed al dipartimento presso cui lavora...

Dipartimento

<u>Nome</u>	<u>Indirizzo</u>	<u>Città</u>
Amministrazione	Via Tito Livio, 27	Milano
Produzione	P.Le Lavater, 3	Torino
Distribuzione	Via Segre, 9	Roma
Direzione	Via Tito Livio, 27	Milano
Ricerca	Via Morone, 6	Milano

Impiegato

<u>Nome</u>	<u>Cognome</u>	<u>Dipart</u>	<u>Ufficio</u>	<u>Stipendio</u>	<u>Città</u>
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

JOIN

- Il criterio per correlare le righe presenti nelle due tabelle è costituito dall'avere il valore dell'attributo Nome nella tabella Dipartimento uguale al valore dell'attributo Dipart nella tabella Impiegato

Dipartimento

<u>Nome</u>	<u>Indirizzo</u>	<u>Città</u>
Amministrazione	Via Tito Livio, 27	Milano
Produzione	P.Le Lavater, 3	Torino
Distribuzione	Via Segre, 9	Roma
Direzione	Via Tito Livio, 27	Milano
Ricerca	Via Morone, 6	Milano

Impiegato

<u>Nome</u>	<u>Cognome</u>	<u>Dipart</u>	<u>Ufficio</u>	<u>Stipendio</u>	<u>Città</u>
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

JOIN

- Un modo per specificare questa interrogazione è attraverso il seguente comando:

Select *

From dipartimento, impiegato

Where dipartimento.nome=dipart

Notazione necessaria per la presenza di due attributi con lo stesso nome nelle due tabelle (in al.relaz. sarebbe richiesta ridenomin.)

Dipartimento

<u>Nome</u>	Indirizzo	Città
Amministrazione	Via Tito Livio, 27	Milano
Produzione	P.Le Lavater, 3	Torino
Distribuzione	Via Segre, 9	Roma
Direzione	Via Tito Livio, 27	Milano
Ricerca	Via Morone, 6	Milano

Impiegato

<u>Nome</u>	<u>Cognome</u>	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Interrogazione 5:

- Invece che riscrivere il nome per intero delle tabelle, è possibile associare degli **alias** al nome delle tabelle.
- Estrarre i nomi degli impiegati e le città in cui lavorano:

```
select I.Nome, Cognome, D.Città  
from Impiegato as I, Dipartimento as D  
where Dipart=D.Nome
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Uso di variabili

- Nella clausola where si può specificare una condizione da valutare riga per riga:
 - Come fare per specificare una condizione di confronto tra valori presenti su righe diverse?
 - Per esempio, individuare gli impiegati con lo stesso cognome



Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Uso di variabili

- Bisogna disporre di due istanze della tabella e confrontare ogni riga della prima tabella con ogni riga della seconda

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Uso di variabili

- Utilizzando gli alias è possibile fare riferimento a più esemplari della stessa tabella.
- Tutte le volte che si introduce un alias per una tabella si dichiara una **variabile** di tipo tabella che possiede come valore il contenuto della tabella di cui è alias.
- Quando una tabella compare più volte (con alias diversi) è necessario considerare ciascun alias come una nuova variabile.

Interrogazione 17:

- Estrarre nome e cognome degli impiegati che hanno lo stesso cognome, ma diverso nome

```
select I1.Nome, I1.Cognome  
from Impiegato as I1, Impiegato as I2  
where I1.Cognome=I2.Cognome and  
I1.Nome<>I2.Nome
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 17:

- Estrarre nome e cognome degli impiegati che hanno lo stesso cognome, ma diverso nome, di impiegati del dipartimento Produzione

```
select I1.Nome, I1.Cognome  
from Impiegato as I1, Impiegato as I2  
where I1.Cognome=I2.Cognome and  
I1.Nome<>I2.Nome and  
I2.Dipart='Produzione'
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Alias

- Per la specifica dell'alias la parola chiave **as** può essere omessa:

```
select I1.Nome, I1.Cognome  
from Impiegato I1, Impiegato I2  
where I1.Cognome=I2.Cognome and  
      I1.Nome<>I2.Nome and  
      I2.Dipart='Produzione'
```


SQL – Valori nulli

- SQL fornisce un predicato per la gestione dei valori nulli. La sintassi è:

`Attributo is [not] null`

e risulta vero solo se l'attributo ha valore nullo (o non nullo se usato nella forma `is not null`).

Interrogazione 17:

- Estrarre nome e cognome degli impiegati per i quali non è definito il dipartimento di affiliazione:

Select nome, cognome

From impiegato

Where dipart is null

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Duplicati

- Nel modello relazionale non si ammette che una relazione possa contenere righe uguali
- In SQL possono esserci più righe uguali (dette duplicati) nel risultato di una select:
 - eliminare i duplicati è costoso (richiede tempo) e spesso non necessario
- SQL esegue la rimozione dei duplicati solo quando esplicitamente richiesto nell'interrogazione
 - Costrutto **select distinct**

Interrogazione 10:

- Estrarre le città dove risiedono persone di cognome Rossi:

```
select Città  
from Impiegato  
where Cognome='Rossi'
```

Città
Milano
Milano

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Interrogazione 11:

- Estrarre le città delle persone di cognome Rossi facendo comparire ogni città al più una volta:

**select distinct Città
from Impiegato
where Cognome='Rossi'**

Città
Milano

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Esercizi

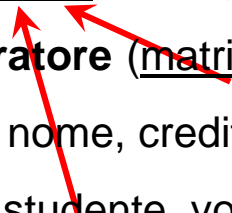
- Matricola, Nome e Cognome degli studenti che hanno dato almeno due esami

Studente (matricola, nome, cognome)

StudenteLavoratore (matricola)

Corso (codice, nome, crediti, docente)

Esame (corso, studente, voto, data)



Join interni ed esterni

- Sino ad ora la specifica della condizione di JOIN (theta-join) è stata inclusa nella clausola where
- Una sintassi alternativa permette di distinguere le condizioni di join da quelle di selezione tra le righe.
- In questo modo è anche possibile specificare le forme esterne dell'operatore di join.

Interrogazione 12:

- Estrarre nomi e cognomi degli impiegati con stipendio maggiore di 40 e le città in cui lavorano:

```
select I.Nome, I.Cognome, D.Città  
from Impiegato I, Dipartimento D  
where I.Dipart=D.Nome AND stipendio>40
```

```
select I.Nome, I.Cognome, D.Città  
from Impiegato I join Dipartimento D on I.Dipart=D.Nome  
where stipendio>40
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Join interni ed esterni

- la condizione di join non compare nella **clausola where** ma nella **clausola from**:

```
select ListaAttributi
from Tabella { [TipoJoin] join Tabella on
               CondizioneJoin }
[ where AltraCondizione ]
```

TipoJoin specifica quale tipo di join utilizzare:

- **inner** (valore di default, è il theta-join dell'algebra relazionale),
right, left, full (questi ultimi tre possono essere seguiti dalla parola chiave outer).

Infrazioni

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Vigili

<u>Matricola</u>	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Auto

<u>Prov</u>	<u>Numero</u>	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Interrogazione 13:

- Estrarre i dati sulle infrazioni ed i vigili che le hanno rilevate, mantenendo nel risultato anche i vigili che non hanno rilevato infrazioni:

Infrazioni(Codice, Data, Vigile, Prov, Numero)

Vigile(Matricola, Cognome, Nome) **Auto**(Prov, Numero, Cognome, Nome)

Codice	Data	Vigile	Prov	Numero	Matricola	Cognome	Nome
34321	1/2/95	3987	MI	39548K	3987	Rossi	Luca
53524	4/3/95	3295	TO	E39548	3295	Neri	Piero
64521	5/4/96	3295	PR	839548	3295	Neri	Piero
73321	5/2/98	9345	PR	839548	9345	Neri	Mario
NULL	NULL	NULL	NULL	NULL	7543	Mori	Gino

SELECT SQL

- Un'estensione di SQL-2 permette di far precedere ad ogni join la parola chiave **natural**.
- In questo modo si consente la specifica del join natural presente nell'algebra relazionale, che prevede di utilizzare nel join di due tabelle una condizione di implicita uguaglianza su tutti gli attributi caratterizzati dallo stesso nome.

Interrogazione 16:

GUIDATORE

Nome	Cognome	NroPatente
Mario	Rossi	VR 2030020Y
Carlo	Bianchi	PZ 1012436B
Marco	Neri	AP 4544442R

AUTOMOBILE

Targa	Marca	Modello	NroPatente
AB 574 WW	Fiat	Punto	VR 2030020Y
AA 652 FF	Fiat	Brava	VR 2030020Y
BJ 747 XX	Lancia	Delta	PZ 1012436B
BB 421 JJ	Fiat	Uno	MI 2020030U

- Estrarre tutti i guidatori ed auto mostrando le relazioni tra loro esistenti:

select *

**from Guidatore as G natural full join
Automobile as A**

JOIN IMPLICITI ED ESPLICITI

- Nella clausola from possono comparire più operazioni di JOIN

select ...

from (T1 join T2 on C1) join T3 on C3

- Nella clausola from possono comparire contemporaneamente JOIN espliciti e non

select ...

from (T1 join T2 on C1), T3

where C3

Ordinamento

- In SQL è possibile specificare che i risultati di una interrogazione siano ordinati sulla base del valore di alcuni attributi.
- Questo è consentito attraverso l'uso della clausola **order by** posta a seguire la clausola where.

Sintassi:

```
order by Attributo1 [ asc | desc ] { ,  
Attributo2  [ asc | desc ] }
```



Interrogazione 19:

- Estrarre il contenuto della tabella Automobile ordinando le righe per marca (discendente) e modello:

```
select *  
from Automobile  
order by Marca desc, Modello
```

AUTOMOBILE	Targa	Marca	Modello	NroPatente
	AB 574 WW	Fiat	Punto	VR 2030020Y
	AA 652 FF	Fiat	Brava	VR 2030020Y
	BJ 747 XX	Lancia	Delta	PZ 1012436B
	BB 421 JJ	Fiat	Uno	MI 2020030U

Esercizio SQL - Azienda

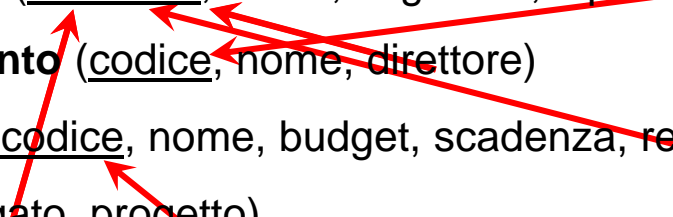
- Il modello relazionale sotto riportato organizza i dati di interesse su dipendenti e progetti di una azienda. In particolare,
 - Per ciascun dipendente sono memorizzate matricola, nome, cognome, dipartimento di affiliazione e stipendio
 - Per ciascun dipartimento sono memorizzati il codice, nome, e matricola del direttore (che è uno dei dipendenti dell'azienda)
 - Nell'azienda vengono portati avanti un certo numero di progetti, per ciascuno dei quali è noto un codice di identificazione, un nome, un budget complessivo, una data di scadenza e la matricola del responsabile di progetto (che è uno dei dipendenti dell'azienda).
 - Vale il vincolo per cui ogni impiegato non può partecipare a più di un progetto, pur essendo prevista la possibilità che un impiegato partecipi ad un progetto e svolga contemporaneamente il ruolo di responsabile per un secondo progetto.
 - La relazione PP tiene traccia dei progetti cui partecipano gli impiegati. Essere responsabile di un progetto non equivale a parteciparvi.

Impiegato (matricola, nome, cognome, dipartimento, stipendio)

Dipartimento (codice, nome, direttore)

Progetto (codice, nome, budget, scadenza, responsabile)

PP (impiegato, progetto)



Esercizio SQL

Interrogazioni:

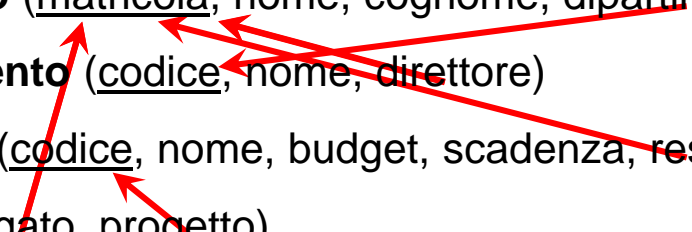
1. Selezionare nome e cognome di ciascun direttore di dipartimento
2. Selezionare nome e cognome dei direttori di dipartimento che sono anche responsabili di progetto
3. Selezionare nome e cognome degli impiegati che sono responsabili di almeno due progetti
4. Selezionare le matricole degli impiegati che guadagnano più dei loro responsabili di progetto
5. Selezionare codici di progetto e matricola del responsabile per i progetti che hanno lo stesso budget
6. Selezionare la matricola degli impiegati che hanno come responsabile di progetto il direttore del dipartimento presso cui sono affiliati
7. Selezionare nome, cognome e stipendio dei responsabili di progetto ordinando il risultato per valori di stipendio dal più grande al più piccolo.

Impiegato (matricola, nome, cognome, dipartimento, stipendio)

Dipartimento (codice, nome, direttore)

Progetto (codice, nome, budget, scadenza, responsabile)

PP (impiegato, progetto)



Operatori aggregati

- Gli operatori dell'algebra relazionale consentono di estrarre informazioni contenute in singole tuple delle tabelle su cui operano:
 - il risultato di una operazione è sempre qualcosa che compare in qualcuna delle tuple degli operandi (o ne è una funzione).
- Una operazione impossibile:
 - Contare le tuple di una relazione. Per esempio: quanti impiegati lavorano nel dipartimento 'Produzione' ?
- E' una operazione impossibile perché il numero di tuple non è un dato che compare esplicitamente in qualcuna delle tuple

Operatori aggregati

- Gli **operatori aggregati** consentono di estrarre informazioni che rappresentano **proprietà di gruppi di tuple**.
- Per esempio, proprietà calcolabili attraverso operazioni di conteggio:
 - Calcolare il numero di impiegati
- ...operazioni di somma:
 - Calcolare la somma degli stipendi degli impiegati
- ...operazioni statistiche:
 - Calcolare valor medio e deviazione standard degli stipendi degli impiegati

Interrogazione 20:

- In SQL, l'operatore **count** effettua il conteggio delle righe del risultato di una select:
 - Estrarre il numero di righe della tabella impiegato:

```
select count(*)  
from Impiegato
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Operatori aggregati

- Gli operatori aggregati vengono gestiti dopo aver risolto l'interrogazione in base alle clausole from e where
- Lo standard SQL prevede cinque operatori aggregati:

count, sum, max, min, avg

Operatori aggregati: count

- La sintassi:

count (< * | [distinct | all] Attributo >)

- L'opzione ***** restituisce il numero di righe
- L'opzione **distinct** restituisce il numero di righe con valore diverso e non nullo di Attributo
- L'opzione **all** (è il valore di default) restituisce il numero di righe con valore non nullo di Attributo

Interrogazione 21:

- Estrarre il numero di diversi cognomi (non nulli) tra le righe della tabella Impiegato:

```
select count(distinct Cognome)  
from Impiegato
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 22:

- Estrarre il numero di valori non nulli nell'attributo cognome tra le righe della tabella Impiegato:

```
select count(all Cognome)  
from Impiegato
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Ancora su distinct e all

- Nota: in alcuni casi la presenza di **distinct** cambia completamente la semantica del risultato!

```
Select count(dipart)  
From Impiegato
```

```
Select count(distinct dipart)  
From Impiegato
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Operatori aggregati

- Gli operatori **sum**, **avg**, **max** e **min** calcolano la somma, la media, il massimo ed il minimo dei valori su un insieme di attributi.
- **sum** ed **avg** ammettono come argomento solo espressioni che rappresentano valori numerici o intervalli di tempo.
- **max** e **min** richiedono solamente che sull'espressione sia definito un ordinamento (si possono anche applicare a stringhe di caratteri e istanti di tempo).

**<sum | max | min | avg> ([distinct | all]
Attributo)**

L'uso delle parole chiave **distinct** ed **all** non ha alcun effetto sugli operatori max e min

Interrogazione 23:

- Estrarre la somma degli stipendi degli impiegati:

```
select sum(Stipendio)  
from Impiegato
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 26:

- Estrarre gli stipendi minimo, massimo e medio tra quelli di tutti gli impiegati:

```
select min(Stipendio) , max(Stipendio) , avg(Stipendio)  
from Impiegato
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Operatori aggregati

- Riassumendo: la target list può contenere nomi di attributi ed operatori aggregati calcolati sugli attributi
 - A ciascun **attributo** nella target list, corrispondono nel risultato della select tanti valori, uno per ogni tupla della tabella
 - A ciascun **operatore aggregato** nella target list, corrisponde nel risultato della select un solo valore calcolato su tutte le tuple della tabella

Operatori aggregati

```
select cognome, stipendio, max(stipendio) maxS  
from Impiegato
```

Cognome	Stipendio	MaxS
Rossi	45	80
Bianchi	36	80
Verdi	40	80
Neri	45	80
Rossi	80	80
Lanzi	73	80
Borroni	40	80
Franco	46	80

Nota: alcuni DBMS (tra cui PostgreSQL) notificano un errore a fronte di tale select

Impiegato

<u>Nome</u>	<u>Cognome</u>	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Operatori aggregati

- E' anche possibile calcolare il valore di un operatore aggregato su un sottoinsieme di tuple:
 - Estrarre la somma degli stipendi degli impiegati nel Dipartimento Amministrazione:

```
select sum(Stipendio)
from Impiegato
where Dipart='Amministrazione'
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Operatori aggregati

- Se si volesse la somma degli stipendi separatamente per ciascun dipartimento?

```
select sum(Stipendio)
from Impiegato
where Dipart='Amministrazione'
```

...

```
select sum(Stipendio)
from Impiegato
where Dipart='Produzione'
```

...

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazioni con raggruppamento

- In generale, capita spesso di voler applicare l'operatore aggregato non ad una intera tabella ma a sottoinsiemi di tuple della tabella
 - La somma degli stipendi degli impiegati che lavorano nel dip. Amministrazione, in quello Produzione, Sviluppo...
- In SQL, l'esecuzione di queste operazioni richiede la specifica di:
 - Un criterio in base al quale definire gruppi di tuple all'interno di una relazione
 - Un operatore aggregato da calcolare separatamente su ciascun gruppo di tuple (count, sum, max, min, ...)

Interrogazioni con raggruppamento

- Criterio di raggruppamento: avere lo stesso valore su uno o più attributi

Raggruppare in base a Dipart

Impiegato

<u>Nome</u>	<u>Cognome</u>	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Interrogazioni con raggruppamento

- Criterio di raggruppamento: avere lo stesso valore su uno o più attributi

Raggruppare in base a Dipart, Ufficio

Impiegato

<u>Nome</u>	<u>Cognome</u>	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Interrogazioni con raggruppamento

- La specifica della lista di attributi su cui effettuare il raggruppamento avviene attraverso la clausola **group by**
- La query raggrupperà le righe che possiedono valori uguali sull'insieme di attributi della clausola group by
- Il valore di eventuali **operatori aggregati** sarà valutato separatamente per ciascun gruppo

Interrogazione 27:

- Estrarre per ogni dipartimento la somma degli stipendi degli impiegati che ci lavorano:
 1. Come definisco i gruppi ?
 2. Cosa calcolo per ogni gruppo ?

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Interrogazione 27:

- Estrarre per ogni dipartimento la somma degli stipendi degli impiegati che ci lavorano:
 - Come definisco i gruppi ?
 - Cosa calcolo per ogni gruppo ?

Dipart	Sum(Stipendio)
Amministrazione	125
Direzione	153
Distribuzione	45
Produzione	82

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Giuseppe	Bianchi	Produzione	20	36	Roma
Paolo	Bernini	Amministrazione	26	40	Verona
Carlo	Nessi	Distribuzione	16	36	Napoli
Carlo	Rossi	Direzione	74	80	Genova
Enrico	Nenzi	Distribuzione	76	43	Sapoli
Paolo	Bernini	Produzione	26	46	Verona
Marco	Bianchi	Produzione	20	36	Roma

Interrogazione 27:

- Estrarre per ogni dipartimento la somma degli stipendi degli impiegati che ci lavorano:

```
select Dipart, sum(Stipendio)
from Impiegato
group by Dipart
```

Dipart	Sum(Stipendio)
Amministrazione	125
Direzione	153
Distribuzione	45
Produzione	82

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Interrogazione 27:

- Esecuzione della select con operatori di raggruppamento:
 1. Si calcola il prodotto cartesiano della clausola from
 2. Si eliminano le tuple che non rispettano l'eventuale clausola where (in questo caso assente)
 3. Si costruiscono i gruppi di tuple in base a valori uguali sugli attributi group by
 4. Si proietta il risultato sugli attributi della target list e si calcolano per ogni gruppo i valori degli operatori aggregati

Interrogazione 28:

- In presenza della clausola group by i valori estratti dall'operazione di SELECT riflettono proprietà di ciascun gruppo, non di singole tuple:
 - La tabella prodotta avrà **una riga per ciascun gruppo**

nella target list possono comparire solo un sottoinsieme degli attributi della clausola group by (oltre agli operatori aggregati)

- La seguente interrogazione è scorretta:

```
select Ufficio, Dipart  
from Impiegato  
group by Dipart
```

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giuseppe	Verdi	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Lanzi	Direzione	7	73	Genova
Paola	Borroni	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

- ...perché all'interno di ciascun gruppo, righe diverse possono avere valori diversi nell'attributo Ufficio (mentre hanno lo stesso valore su Dipart)

Interrogazione 29:

- Per ogni dipartimento, estrarre il nome del dipartimento ed il numero di impiegati:

```
select Dipart, count(*)  
from Impiegato  
group by Dipart
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 29:

- Per ogni dipartimento, estrarre il codice, il numero di impiegati e la città dove il dipartimento ha sede:

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 29:

- Per ogni dipartimento, estrarre il codice, il numero di impiegati e la città dove il dipartimento ha sede:

sbagliata:

```
select Dipart, count(*), D.Città
from Impiegato I join Dipartimento D
    on (I.Dipart=D.Nome)
group by Dipart
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 29:

- Per ogni dipartimento, estrarre il codice, il numero di impiegati e la città dove il dipartimento ha sede:

corretta:

```
select Dipart, count(*), D.Città  
from Impiegato I join Dipartimento D  
    on (I.Dipart=D.Nome)  
group by Dipart, D.Città
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Predicati sui gruppi

- Con la clausola **group by** le righe possono essere raggruppate in sottoinsiemi.
- Le condizioni da verificare a livello di singole tuple compaiono nella **clausola where**.
- Tuttavia può essere necessario verificare delle condizioni a livello di gruppi. Per esempio:
 - Quali dipartimenti hanno più di 5 impiegati ?
- Questa interrogazione non può essere risolta imponendo una condizione nella clausola where perché il numero di impiegati di ciascun dipartimento non compare come attributo di una singola tupla
 - Inoltre i gruppi vengono costruiti **dopo** aver eliminato le righe che non rispettano la clausola where

Predicati sui gruppi

- Le condizioni da verificare a livello di gruppo sono specificate attraverso un nuovo costrutto: la clausola **having**.
- Questa consente la specifica di un predicato in cui compaiono gli operatori aggregati.

Interrogazione 31:

- Estrarre i dipartimenti con più di 3 impiegati:

```
select Dipart, count(nome, cognome)
from Impiegato
group by Dipart
having count(nome, cognome) > 3
```

- Solo i predicati dove compaiono operatori aggregati devono comparire come argomento della clausola having.

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Interrogazione 31:

- Estrarre i dipartimenti che spendono più di 100 in stipendi dei dipendenti:

```
select Dipart, sum(Stipendio) as SommaStipendi  
from Impiegato  
group by Dipart  
having sum(Stipendio) > 100
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

Predicati sui gruppi

- La sintassi prevede l'uso della clausola having anche in assenza di una clausola group by
- In questo caso l'intero insieme di righe è trattato come un unico raggruppamento (se la condizione non è verificata il risultato sarà una tabella vuota)

Predicati sui gruppi

- La condizione espressa nella clausola having viene valutata al termine dell'interrogazione
- In presenza di una clausola where e di una clausola having, la prima ha precedenza sulla seconda:
 - i gruppi vengono composti dopo aver verificato la clausola where e solo dopo la composizione dei gruppi si valuta la clausola having

Interrogazione 32:

- Estrarre i dipartimenti per cui la media degli stipendi degli impiegati che lavorano nell'ufficio 20 è superiore a 25 milioni:

```
select Dipart
from Impiegato
where Ufficio=20
group by Dipart
having avg(Stipendio)>25
```

Dipartimento

Nome	Indirizzo	Città
------	-----------	-------

Impiegato

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
------	---------	--------	---------	-----------	-------

SELECT SQL

- La forma generale di un'interrogazione SQL che includa le clausole sin'ora viste (altre le vedremo in seguito) è:

```
select ListaAttributi  
from ListaTabelle  
where CondizioniSemplici  
group by ListaAttributiDiRaggruppamento  
having CondizioniAggregate  
order by ListaAttributiDiOrdinamento
```

Esercizio SQL - Azienda

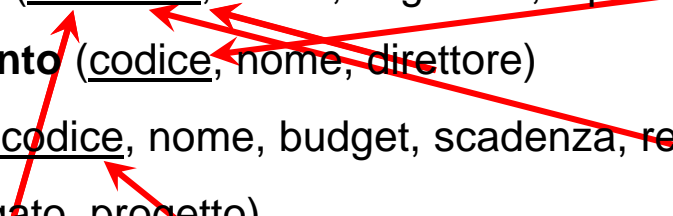
- Il modello relazionale sotto riportato organizza i dati di interesse su dipendenti e progetti di una azienda. In particolare,
 - Per ciascun dipendente sono memorizzate matricola, nome, cognome, dipartimento di affiliazione e stipendio
 - Per ciascun dipartimento sono memorizzati il codice, nome, e matricola del direttore (che è uno dei dipendenti dell'azienda)
 - Nell'azienda vengono portati avanti un certo numero di progetti, per ciascuno dei quali è noto un codice di identificazione, un nome, un budget complessivo, una data di scadenza e la matricola del responsabile di progetto (che è uno dei dipendenti dell'azienda).
 - La relazione PP tiene traccia dei progetti cui partecipano gli impiegati. **Un impiegato può partecipare a più progetti.** Essere responsabile di un progetto non implica parteciparvi.

Impiegato (matricola, nome, cognome, dipartimento, stipendio)

Dipartimento (codice, nome, direttore)

Progetto (codice, nome, budget, scadenza, responsabile)

PP (impiegato, progetto)



Esercizio SQL

Interrogazioni:

1. Selezionare la matricola di ciascun responsabile ed il numero di progetti che dirige
2. Selezionare nome e cognome di ciascun impiegato ed il numero di progetti a cui partecipa
3. Selezionare nome e cognome degli impiegati che sono responsabili di almeno due progetti
4. Selezionare per ciascun dipartimento, il nome del dipartimento ed il numero dei suoi affiliati che risultano essere responsabili di progetto
5. Selezionare per ciascun dipartimento, il nome del dipartimento ed il numero dei suoi affiliati che hanno cognome 'Casini'
6. Selezionare il massimo numero di impiegati affiliati ad un dipartimento

Impiegato (matricola, nome, cognome, dipartimento, stipendio)

Dipartimento (codice, nome, direttore)

Progetto (codice, nome, budget, scadenza, responsabile)

PP (impiegato, progetto)



Operatori aggregati

- In generale, **gli operatori aggregati non possono essere nidificati**:
 - SELECT avg(count(*)) ...
 - SELECT max(sum(valore)) ...
 - SELECT avg(sum(valore)) ...
- Sono tutte espressioni **SBAGLIATE !!!**
- L'estrazione di questo tipo di informazioni richiede il ricorso a costrutti diversi:
 - SELECT nidificate
 - Viste