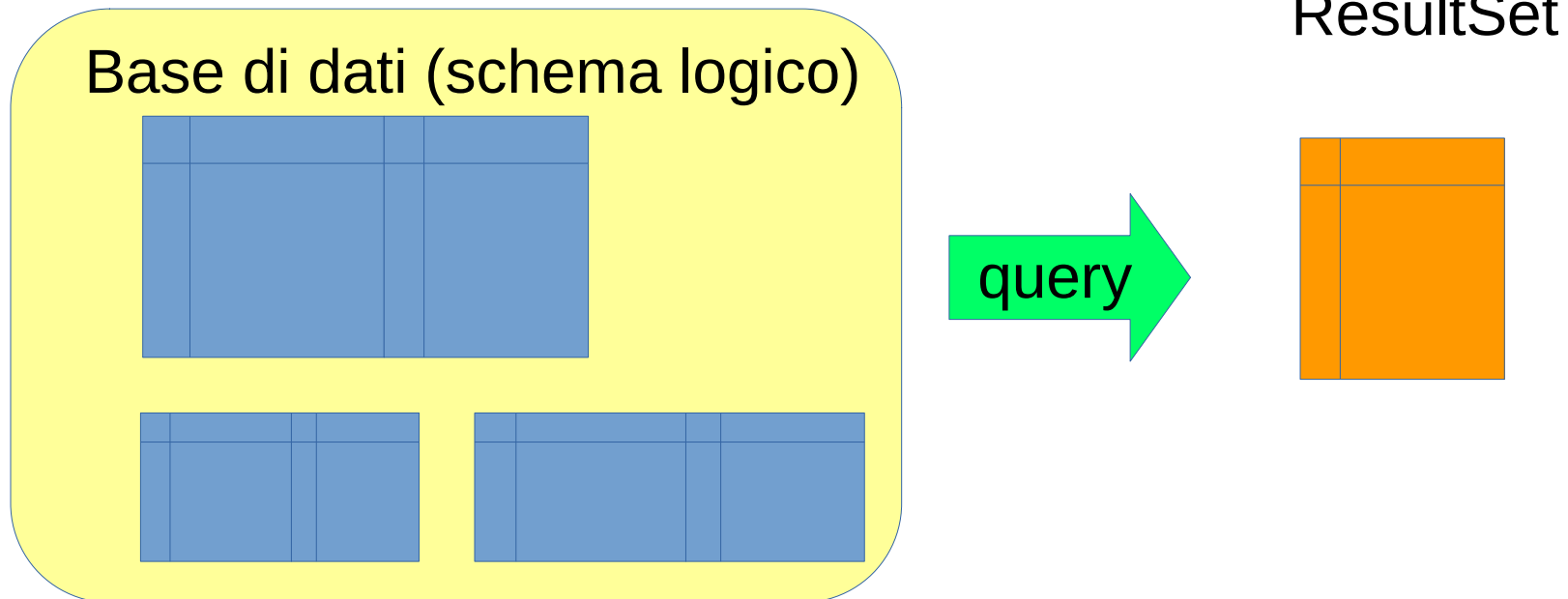


DML: SELECT



Interrogazione della base di dati

- ❑ Le query sono operazioni di interrogazione che non cambiano ne' lo schema ne' il contenuto delle tabelle
- ❑ Il Resultset è il risultato della query ed è in forma tabellare



Istruzione SELECT (versione base)

```
SELECT target_list  
FROM lista_tabelle  
[ WHERE Condizione ]
```

target_list: lista delle colonne che voglio nel ResultSet, separate da una virgola

lista_tabelle: tabelle da cui prendere i dati

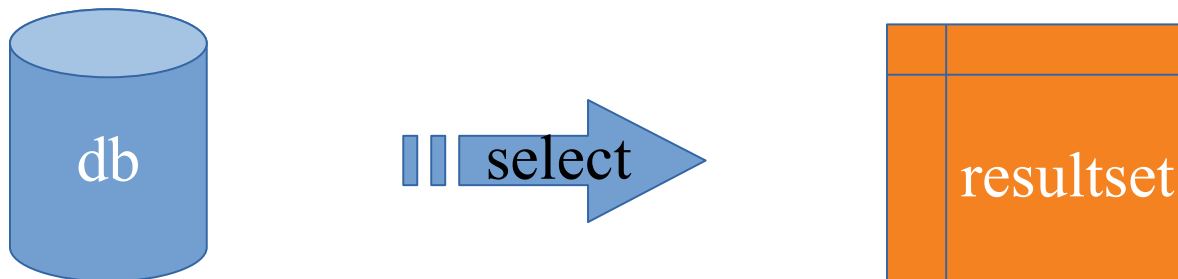
clausola WHERE: condizione che le righe devono soddisfare per comparire nel ResultSet



SELECT (colloquio)

L'istruzione SELECT

- Appartiene alle DML
- recupera dati dal database
- Implementa tutte le operazioni dell'algebra relazionale
 - **Selezione** (scelta delle righe)
 - **Proiezione** (scelta delle colonne)
 - **Ridenominazione** (nomi delle colonne del Resultset)
 - **Join** (correlazione dei dati di più tabelle)
- L'esecuzione di una select NON modifica l'istanza e lo schema della base di dati.



Istruzioni DML (colloquio)

Statement	Description
SELECT	Retrieves data from one or more tables.
INSERT	Adds one or more new rows to a table.
UPDATE	Changes one or more existing rows in a table.
DELETE	Deletes one or more existing rows from a table.

- Quali istruzioni fanno parte delle istruzioni CRUD
- Quali istruzioni cambiano l'istanza della base di dati?
- Quali istruzioni cambiano lo schema della base di dati?
- Quali istruzioni restituiscono un ResultSet?



BASE SELECT



Esempio

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Proiezione, senza selezione

Reddito e nome di tutte le persone

Voglio nel Resultset solo le colonne Nome e Reddito

```
select reddito,nome  
from persone;
```

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Proiezione, senza selezione

```
select reddito,nome  
from persone;
```

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



Reddito	Nome
21	Andrea
15	Aldo
42	Maria
35	Anna
30	Filippo
40	Luigi
20	Franco
41	Olga
35	Sergio
87	Luisa

Esercizio

- Voglio un ResultSet con età, reddito e nome delle persone con meno di 30 anni

Scrivere la query



Esercizio

- Voglio un ResultSet con età, reddito e nome delle persone con meno di 30 anni

```
select età, reddito, nome  
from persone  
where eta < 30;
```



Esercizio

PERSONE

Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

```
select età, reddito, nome  
from persone  
where età < 30;
```

- Scrivere il ResultSet



Esercizio

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



Età	Reddito	Nome
27	21	Andrea
25	15	Aldo
26	30	Filippo

```
Select età, reddito, nome  
from persone  
where età < 30;
```



Proiezione, con selezione

Reddito e nome di tutte le persone che non si chiamano Andrea

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Proiezione, con selezione

```
select reddito,nome  
from persone  
where nome != 'Andrea';
```

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



Reddito	Nome
15	Aldo
42	Maria
35	Anna
30	Filippo
40	Luigi
20	Franco
41	Olga
35	Sergio
87	Luisa

SELECT, star wildcard

Con * verranno visualizzate nel Resultset tutte le colonne della tabella **rispettandone l'ordine**

```
select *  
from persone  
where eta < 30
```


Ex 1

AUTO

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AA333	Ferrari	GTO	rosso	3800
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750

1Voglio conoscere targa,marca,modello,colore,cilindrata delle auto della Fiat

Resultset

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750



Ex 1

AUTO

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AA333	Ferrari	GTO	rosso	3800
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750

```
select *  
from AUTO  
where  Marca='Fiat'
```

Resultset

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750



Ex 2

AUTO

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AA333	Ferrari	GTO	rosso	3800
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750

Voglio conoscere la targa e la cilindrata delle Fiat Panda.
Scrivere query e Resultset



Ex 2

AUTO

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AA333	Ferrari	GTO	rosso	3800
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750

```
SELECT targa, cilindrata
FROM auto
WHERE  marca='Fiat' AND modello='Panda'
```

Resultset

targa	cilindrata
XY111	600
EE666	750



Ex 3

AUTO

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AA333	Ferrari	GTO	rosso	3800
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750

Voglio conoscere modello e targa delle auto con cilindrata > 1200

Scrivere query e Resultset



Ex 3

AUTO

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AA333	Ferrari	GTO	rosso	3800
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750

```
select modello, targa  
from AUTO  
where cilindrata > 1200
```

Resultset

modello	targa
GTO	AA333



Ex 4

AUTO

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AA333	Ferrari	GTO	rosso	3800
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750

Voglio conoscere modello e colore delle Ferrari o delle auto con cilindrata > 1000

Scrivere Resultset e query SQL



Ex 4

AUTO

targa	marca	modello	colore	cilindrata
XY111	Fiat	Panda	bianco	600
AA333	Ferrari	GTO	rosso	3800
AM777	Fiat	Punto	rosso	1200
EE666	Fiat	Panda	rosso	750

```
select modello, colore  
from AUTO  
Where cilindrata > 1000 OR Marca='Ferrari'
```

Resultset

modello	colore
GTO	rosso
Punto	rosso



Esempio

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Ridenominazione dei campi del ResultSet

```
SELECT nome AS nominativo, reddito AS incasso  
FROM persone  
WHERE eta < 27;
```

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



nominativo	incasso
Aldo	15
Filippo	30

NB La ridenominazione non cambia i nomi delle colonne delle tabelle (schema) ma i nomi delle colonne del ResultSet

Query 'What if'

Il linguaggio SQL consente di effettuare delle query che eseguono e presentano delle elaborazioni sui dati

Questa tipologia di query è utilizzata in Data Analysis per fare stime, previsioni e proiezioni.

Schema e istanza, ovviamente, non cambiano

Query 'what if'

```
SELECT nome, reddito, reddito+2
FROM Persone
WHERE Nome = 'Anna'
```

NB Il reddito di Anna non è cambiato ('query what if')

Nome	reddito	reddito+2
Anna	35	37

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Espressioni nella target list

```
SELECT reddito/2  
FROM Persone  
WHERE Nome = 'Luigi'
```

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

reddito/2

20

Espressioni nella target list e ridenominazione

```
SELECT reddito/2 AS reddito_semestrale  
FROM Persone  
WHERE Nome = 'Luigi'
```

reddito_semestrale
20

Marketing: stime di spesa residua

```
create table clienti(nome varchar(20), spesa_mensile int
    unsigned);
```

```
insert into clienti values('rino', 30), ('gino',50),
    ('pino',60);
```

Spesa trimestrale di Rino

```
select spesa_mensile * 3 AS "spesa trimestrale" from clienti
    where nome = 'rino';
```

Spesa trimestrale residua di Rino

```
select spesa_mensile * 3 - 50 AS "spesa trimestrale residua
    di rino" from clienti where nome = 'rino';
```

Agenzia entrate

Individuare il nome delle persone sotto i 30 e sopra i 60 che incassano più di 25

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87



Creare la tabella Persone

```
create table persone(  
  nome varchar(20),  
  età tinyint unsigned,  
  reddito tinyint  
);
```

Aggiungere le seguenti righe a Persone

```
insert into persone values ('Filippo',26,30);  
insert into persone values ('Sergio',85,35);  
insert into persone values ('Luisa',75,87);  
insert into persone values ('Rino',75,20);  
insert into persone values ('Gino',20,20);  
insert into persone values ('Anna',50,40);
```

Condizioni con operatori logici

```
SELECT nome  
FROM persone  
WHERE reddito > 25 AND eta < 30 OR eta > 60;
```

```
SELECT nome  
FROM persone  
WHERE (reddito > 25 AND eta < 30) OR eta > 60;
```

```
SELECT nome  
FROM persone  
WHERE reddito > 25 AND (eta < 30 OR eta > 60);
```



Condizioni con operatori logici

```
SELECT nome  
FROM persone  
WHERE reddito > 25 AND eta < 30 OR eta > 60;
```

X

```
SELECT nome  
FROM persone  
WHERE (reddito > 25 AND eta < 30) OR eta > 60;
```

equivalente

X

```
SELECT nome  
FROM persone  
WHERE reddito > 25 AND (eta < 30 OR eta > 60);
```

OK



Valore null

Individuare le persone la cui età è sconosciuta

```
select *  
from persone  
where eta is null
```

Valore null

Individuare le persone la cui età è nota

```
select *  
from persone  
where eta is not null
```

Valore null

Individuare le persone la cui età è maggiore di 40 o sconosciuta

$S_{\text{Età} > 40 \text{ OR Et\`a IS NULL}}(\text{Persone})$ (algebra relazionale)

```
select *  
from persone  
where eta > 40 OR eta is null
```

Query su insiemi di valori : clausola IN

Visualizzare tutte le informazioni relative agli studenti che si chiamano Cino ,Pino o Lino

```
SELECT *  
FROM studenti  
WHERE nome='Cino' OR nome = 'Pino' OR nome ='Lino';
```

Questa query può essere scritta in maniera più sintetica mediante l'operatore IN

```
SELECT *  
FROM studenti  
WHERE nome IN ('Cino', 'Pino', 'Lino');
```

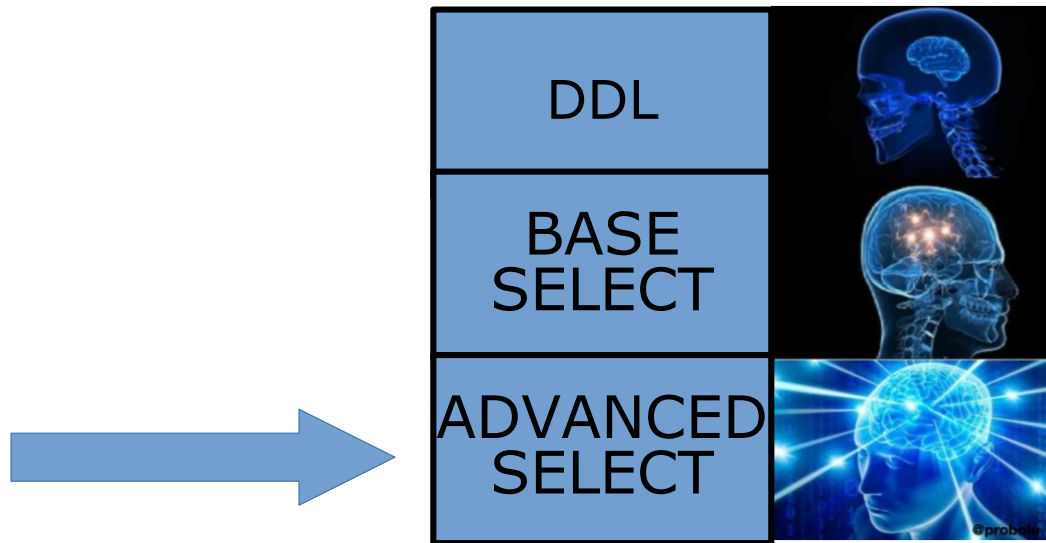
Domanda:quante righe ha il resultset della query

```
SELECT *  
FROM studenti  
WHERE nome='Cino' AND nome = 'Pino' AND nome ='Lino';
```

AUTONOLEGGIO: ESERCIZI SELECT BASE



Advanced Select



Operatore LIKE

Si tratta di un operatore utilizzabile nella condizione WHERE della query. E' particolarmente utile per effettuare le ricerche su dati parziali ('matching parziale').

```
SELECT <campi>  
FROM <tabella>  
WHERE <campo LIKE ricerca>
```

Wildcard

Con l'operatore LIKE si utilizzano wildcard ('caratteri jolly')

- Il simbolo % ('percentuale') rappresenta 'zero o più caratteri'
- Il simbolo _ ('underscore') rappresenta 'uno e un solo carattere'

OPERATORE LIKE

Le persone che hanno un nome che inizia per 'A'

```
SELECT *  
FROM persone  
WHERE nome LIKE 'A%';
```

OPERATORE LIKE

I numeri di telefono fissi di Roma

```
SELECT *  
FROM utenti  
WHERE numero LIKE '06%';
```

OPERATORE LIKE

Le persone che hanno un nome che inizia per 'A' e finisce con 'o'

```
SELECT *  
FROM persone  
WHERE nome LIKE 'A%o';
```

OPERATORE LIKE

Le persone che hanno un nome che contiene la stringa 'nto'

```
SELECT *  
FROM persone  
WHERE nome LIKE '%nto%';
```

OPERATORE LIKE

Gli indirizzi email appartenenti ai domini di yahoo (.it, .com etc.)

```
SELECT *  
FROM persone  
WHERE email LIKE '%@yahoo.%;'
```


Condizione 'LIKE'

Le persone che hanno un nome lungo
5 caratteri

```
select *  
from persone  
where nome like '_____';
```

Condizione 'LIKE'

Le persone che hanno un nome che inizia per 'A' e ha una 'd' come terza lettera

```
SELECT *  
FROM persone  
WHERE nome LIKE 'A_d%';
```

Clausola DISTINCT

La clausola DISTINCT elimina i valori ripetuti dalle colonne a cui è applicato

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Rino	25	15
Maria	55	42
Rino	50	35

```
SELECT DISTINCT nome  
FROM persone;
```



Nome
Andrea
Rino
Maria

Clausola DISTINCT

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Rino	25	15
Maria	55	42
Rino	50	35
Maria	22	42

```
SELECT DISTINCT nome, reddito  
FROM persone;
```



Nome	Reddito
Andrea	21
Rino	15
Maria	42
Rino	35

Clausola DISTINCT: uso errato

PERSONE		
Nome	Età	Reddito
Andrea	27	21
Rino	25	15
Maria	55	42
Rino	50	35
Maria	22	42

```
SELECT nome DISTINCT reddito  
FROM persone;
```



```
ERROR 1064 (42000): You have  
an error in your SQL syntax;  
check the manual that  
corresponds to your MySQL  
server version for the right  
syntax to use near 'distinct  
reddito from persone' at line  
1
```

Ordinamento

→ Nome e reddito delle persone con meno di trenta anni in ordine alfabetico

```
select nome, reddito  
from persone  
where eta < 30  
order by nome ASC;
```

→

Ordinamento del risultato

La clausola `ORDER BY` permette di ordinare il risultato rispetto ad una o più colonne

- in ordine crescente (default)
- In ordine discendente

Modificatori

- `ASC` per l'ordinamento ascendente (default, può essere omesso)
- `DESC` per l'ordinamento discendente

Ordinamento del risultato

```
mysql> select *  
      -> from studenti;
```

nome
luigi
anna
davide
vincenzo
matteo
ettore
giovanni

```
mysql> select *  
      -> from studenti  
      -> order by nome ASC;
```

nome
anna
davide
ettore
giovanni
luigi
matteo
vincenzo

```
mysql> select *  
      -> from studenti  
      -> order by nome DESC;
```

nome
vincenzo
matteo
luigi
giovanni
ettore
davide
anna

Ordinamento del risultato

Ordinamento del Resultset rispetto al nome e rispetto al cognome

```
mysql> select nome, cognome, eta from dipendenti ORDER BY nome, cognome;
```

nome	cognome	eta
Azzurra	Felce	14
Dario	Drome	23
Dario	Lampa	31
Pino	Silvestre	300
Rino	Rano	16

Operatore LIMIT MySQL

- *L'operatore LIMIT limita il numero di righe mostrate*
- *Il DBMS determina il Resultset e ne mostra solo il numero indicato*

```
mysql> select * from dipendenti;
+----+-----+-----+-----+
| id | nome   | cognome | eta  |
+----+-----+-----+-----+
| 1  | Pino   | Silvestre | 300 |
| 2  | Azzurra | Felce   | 14  |
| 3  | Dario  | Lampa   | 31  |
| 4  | Rino   | Rano    | 16  |
| 17 | Dario  | Drome   | 23  |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```



```
mysql> select * from dipendenti limit 2;
+----+-----+-----+-----+
| id | nome   | cognome | eta  |
+----+-----+-----+-----+
| 1  | Pino   | Silvestre | 300 |
| 2  | Azzurra | Felce   | 14  |
+----+-----+-----+-----+
```

Operatore LIMIT MySQL

- *Riduce il carico di rete e delle applicazioni quando non si ha bisogno di tutto il Resultset (ex paginazione di Amazon)*
- *Va posto sempre in fondo alla query*

```
mysql> select * from dipendenti;
+----+-----+-----+-----+
| id | nome   | cognome | eta  |
+----+-----+-----+-----+
| 1  | Pino   | Silvestre | 300 |
| 2  | Azzurra | Felce   | 14  |
| 3  | Dario  | Lampa   | 31  |
| 4  | Rino   | Rano    | 16  |
| 17 | Dario  | Drome   | 23  |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```



```
mysql> select * from dipendenti limit 2;
+----+-----+-----+-----+
| id | nome   | cognome | eta  |
+----+-----+-----+-----+
| 1  | Pino   | Silvestre | 300 |
| 2  | Azzurra | Felce   | 14  |
+----+-----+-----+-----+
```

LIMIT con Offset MySQL

```
mysql> select * from dipendenti;
+----+-----+-----+-----+
| id | nome   | cognome | eta  |
+----+-----+-----+-----+
| 1  | Pino   | Silvestre | 300 |
| 2  | Azzurra | Felce   | 14  |
| 3  | Dario  | Lampa   | 31  |
| 4  | Rino   | Rano    | 16  |
| 17 | Dario  | Drome   | 23  |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```

LIMIT può essere usato anche con due parametri

`limit skip,n`

`skip`= numero di righe del resultset da ignorare (offset)

`n`= numero righe da mostrare

```
mysql> select * from dipendenti limit 0,2;
+----+-----+-----+-----+
| id | nome   | cognome | eta  |
+----+-----+-----+-----+
| 1  | Pino   | Silvestre | 300 |
| 2  | Azzurra | Felce   | 14  |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select * from dipendenti limit 2,2;
+----+-----+-----+-----+
| id | nome   | cognome | eta  |
+----+-----+-----+-----+
| 3  | Dario  | Lampa   | 31  |
| 4  | Rino   | Rano    | 16  |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select * from dipendenti limit 2,3;
+----+-----+-----+-----+
| id | nome   | cognome | eta  |
+----+-----+-----+-----+
| 3  | Dario  | Lampa   | 31  |
| 4  | Rino   | Rano    | 16  |
| 17 | Dario  | Drome   | 23  |
+----+-----+-----+-----+
```

Ordinamento del risultato e limit MySQL

```
mysql> select *  
      -> from studenti;
```

nome
luigi
anna
davide
vincenzo
matteo
ettore
giovanni

```
mysql> select *  
      -> from studenti  
      -> order by nome DESC;
```

nome
vincenzo
matteo
luigi
giovanni
ettore
davide
anna

```
mysql> select *  
      -> from studenti  
      -> order by nome DESC  
      -> limit 3;
```

nome
vincenzo
matteo
luigi

Focus: Ordine keyword nella SELECT MySQL

select *columns list*

from *tables*

[where *conditions]*

[order by *columns list]*

[limit *skip,n]*

Ex Mail

*Visualizzazione delle ultime 10 mail ricevute da
rino@gmail.com*

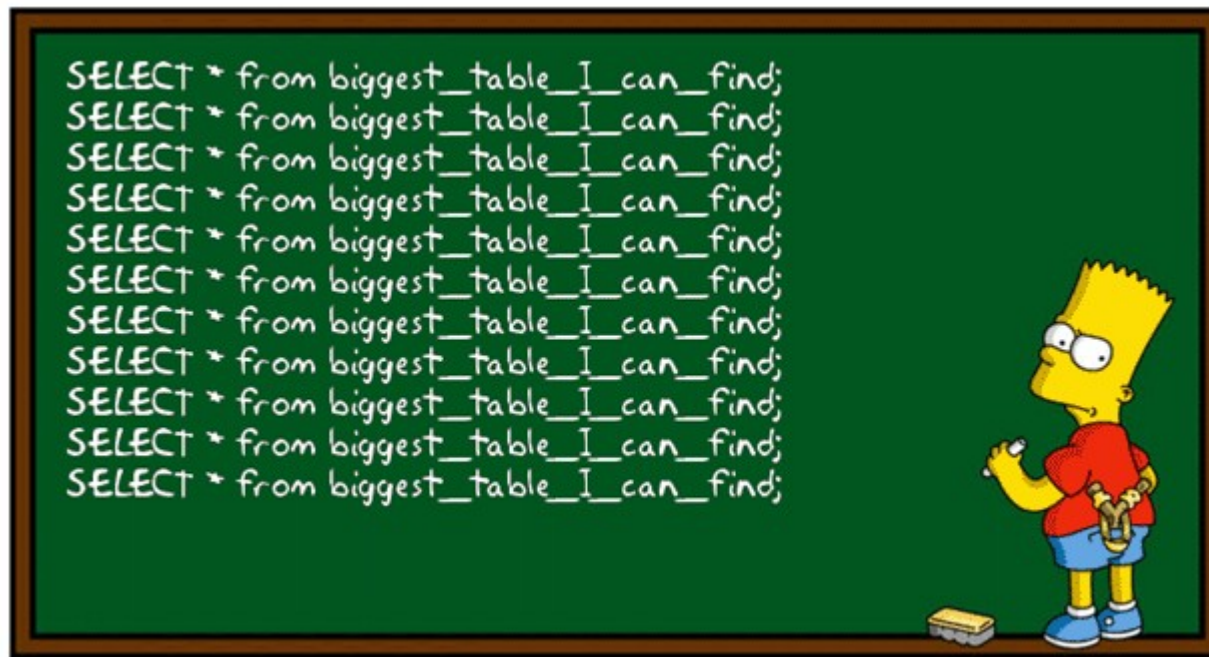
MAIL

id	destinatario	oggetto	testo	mittente	data_invio
1	rino@gmail.com	python	Bla bla bla	gino@yahoo.it	2022-01-01 12:39:40
2	pino@gmail.com	Hai vinto un iphone	Sei stato sorteggiato ..	nino@yahoo.com	2022-01-10 17:40:16
5	rino@gmail.com	Hai vinto un iphone	Sei stato sorteggiato ..	nino@yahoo.com	2022-01-10 17:40:17

*Visualizzazione delle ultime 10 mail ricevute da
rino@gmail.com*

```
SELECT oggetto, testo, mittente, data_invio  
FROM mail  
WHERE destinatario = 'rino@gmail.com'  
ORDER BY data_invio DESC  
LIMIT 0,10;
```


SELECT, star wildcard



OPERATORI E FUNZIONI

Concat() MySQL

La funzione `concat()` consente di concatenare due o più colonne

E' possibile utilizzare la ridenominazione

```
select concat(marca, '---', modello) from auto;
```

concat(marca, '---', modello)
fiat---punto
ford---focus
ferrari---GTO
fiat---panda
fiat---panda

length()

La funzione `length()` determina la lunghezza di una stringa

```
select marca,modello from auto where length(marca) = 4;
```

marca	modello
fiat	punto
ford	focus
fiat	panda
opel	corsa
opel	corsa
opel	mokka

```
select marca,length(marca) from auto;
```

marca	length(marca)
fiat	4
ford	4
ferrari	7
fiat	4
opel	4
opel	4
opel	4

Round()

La funzione `round(value,digit)` consente di stabilire il numero di cifre decimali di un certo valore

```
select round(2.12345678,3);
```

```
+-----+
| round(2.12345678,3) |
+-----+
|                2.123 |
+-----+
```

```
select avg(prezzo_giornaliero) from categoria;
```

```
+-----+
| avg(prezzo_giornaliero) |
+-----+
|             141.2500 |
+-----+
```

```
1 row in set (0.03 sec)
```

```
mysql> select round(avg(prezzo_giornaliero),2) from categoria;
```

```
+-----+
| round(avg(prezzo_giornaliero),2) |
+-----+
|             141.25 |
+-----+
```

Funzioni per le date MYSQL

Le funzioni `year()`, `month()`, `day()` estraggono valori da campi di tipo `datetime` e `date`

```
select year('2022-01-26 21:04:28');
+-----+
| year('2022-01-26 21:04:28') |
+-----+
|                2022         |
+-----+
1 row in set (0.00 sec)
```

```
select month('2022-01-26 21:04:28');
+-----+
| month('2022-01-26 21:04:28') |
+-----+
|                1             |
+-----+
1 row in set (0.00 sec)
```

```
select day('2022-01-26 21:04:28');
+-----+
| day('2022-01-26 21:04:28')   |
+-----+
|                26            |
+-----+
```

Funzioni per gli orari

Le funzioni `hour()`, `minute()`, `second()` estraggono valori da campi di tipo `datetime` e `time`

```
select hour('2022-01-26 21:04:28');
+-----+
| hour('2022-01-26 21:04:28') |
+-----+
|                               21 |
+-----+
1 row in set (0.00 sec)
```

```
select minute('2022-01-26 21:04:28');
+-----+
| minute('2022-01-26 21:04:28') |
+-----+
|                               4 |
+-----+
1 row in set (0.00 sec)
```

```
select second('2022-01-26 21:04:28');
+-----+
| second('2022-01-26 21:04:28') |
+-----+
|                               28 |
+-----+
```

Funzioni per i campi datetime

La funzione `datediff()` visualizza la differenza in giorni tra due date

```
select datediff('2022-12-31','2022-01-01');
```

```
+-----+
| datediff('2022-12-31','2022-01-01') |
+-----+
|                                364 |
+-----+
```

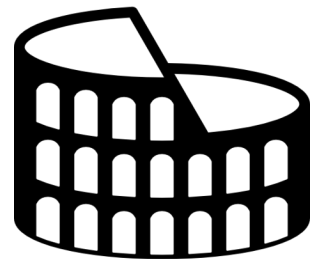
```
select datediff(now(),'1492-10-12');
```

```
+-----+
| datediff(now(),'1492-10-12') |
+-----+
|                193319 |
+-----+
```


Esercizio

Determinare gli anni trascorsi dal 17 giugno 2001

```
select round(datediff(now(),'2001-06-17')/365,0);  
+-----+  
| round(datediff(now(),'2001-06-17')/365,0) |  
+-----+  
|                                     21 |  
+-----+
```



Funzioni per i campi datetime

La funzione `timediff()` visualizza la differenza in ore:minuti:secondi tra due datetime o due time

```
select timediff('12:30:00', '10:00:00');
```

```
+-----+  
| timediff('12:30:00', '10:00:00') |  
+-----+  
| 02:30:00 |  
+-----+
```

```
select timediff('2022-01-02 12:30:00', '2022-01-01 10:00:00');
```

```
+-----+  
| timediff('2022-01-02 12:30:00', '2022-01-01 10:00:00') |  
+-----+  
| 26:30:00 |  
+-----+
```

AUTONOLEGGIO: ESERCIZI SELECT ADVANCED



Riepilogo

- 1) Quali istruzioni DML modificano l'istanza
- 2) Quale istruzione DML restituisce un ResultSet
- 3) In quali istruzioni è prevista la clausola `WHERE`?
- 4) L'operatore `limit` velocizza una query di ordinamento?
- 5) A cosa serve la clausola `DISTINCT`?
- 6) Tipo 'ENUM'
- 7) Cos'è l'inserimento bulk?
 - *Vantaggi*
 - *svantaggi*