

definizione di database

modello entità-relazione

cardinalità

DML (Data Manipulation Language)

Manipola i dati

-Insert into

-Update

-Delete

DDL (Data Definition Language)

Definisce la struttura di un database

-Create

-Alter

-Drop

CREATE TABLE "Nome Tabella"(Attributi...)

Crea una tabella con i seguenti tipi di dati nelle colonne corrispondenti

```
CREATE TABLE Studente(  
    ID Int,  
    Cognome varchar(255),  
    Nome varchar(255),  
    DataDiNascita date,  
    AltezzaInMetri decimal(3,2),  
    PesoInKG decimal(5,2)  
)
```

INSERT INTO "Nome Tabella"(Nomi Colonne...)

Values(Valori...)

Insert into inserisce nella tabella i nomi delle colonne

Values invece inserisce i valori nelle colonne corrispondenti

```
INSERT INTO Studente(ID, Cognome, DataDiNascita, AltezzaInMetri)  
Values(0, 'Tolentino', '2005-06-22', 1.66)
```

UPDATE "nome tabella"

Permette la modifica dei dati delle tabelle

SET "Colonna = Dato che si vuole modificare"

WHERE "Riga interessate"

```
UPDATE Studiante  
SET PesoInKG = 77  
WHERE ID = 0;
```

DELETE FROM "nome tabella" WHERE "cosa si vuole cancellare"

Permette di cancellare in questo caso dalla tabella studente tutti i tolentino

```
DELETE FROM Studiante WHERE Cognome = 'Tolentino';
```

SELECT \* FROM "nome tabella"

Ci permette di visualizzare la tabella

```
y2.sql - 192...lentino.mirko (93)* SQLQuer  
SELECT * FROM Studiante
```

ALTER TABLE "nome tabella"

ADD "nome nuova colonna" "tipo di dati che ci saranno all'interno"

Alter Table e Add in questo caso ci permette di aggiungere una nuova colonna

```
ALTER TABLE Studiante  
ADD Telefono int
```

ALTER TABLE "nome tabella"

ALTER COLUMN "nome colonna" "tipo"

Ci permette di cambiare il tipo di dato in una colonna

```
y2.sql - 192...lentino.mirko (93)* SQLQuery y2.sql - 192...lentino.mirko (100)  
ALTER TABLE Studiante  
ALTER COLUMN Telefono varchar(10)
```

IDENTITY("inizio", "incremento")

Ci permette di creare automaticamente l'id

PRIMARY KEY

è una chiave univoca in questo caso l'id

```
CREATE TABLE Studente(  
    ID int IDENTITY(0,1) PRIMARY KEY,  
    Nome varchar(50),  
    Cognome varchar(50)  
)
```

FOREIGN KEY REFERENCES "Nome tabella("Nome colonna")"

Ci permette di prendere una chiave esterna proveniente da un'altra tabella

```
CREATE TABLE Telefono(  
    ID int IDENTITY(0,1) PRIMARY KEY,  
    Numero varchar(50),  
    IDS int FOREIGN KEY REFERENCES Studente(ID)  
)
```

Con inner join ci mostrerà la tabella con le partecipazioni e la lista dei progetti

campi in comune tra partecipazione e progetto è l'id

(da ricordare il foreign key)

```
Select * from partecipazione1  
inner join progetto  
on partecipazione1.IDP = progetto.ID
```

```

Select progetto.nome, dipendente.nome, dipendente.cognome from partecipazione1
inner join progetto
on partecipazione1.IDP = progetto.ID
inner join dipendente
on partecipazione1.IDD = dipendente.id

```

=

	nome	nome	cognome
1	Esselunga Viale Suzzani	Stefano	Bosoni
2	Esselunga	Antonio	Grieco

```

order by dipendente.nome

```

metterà solo i progetti che il nome "ING"

```

where progetto.nome= 'ING'

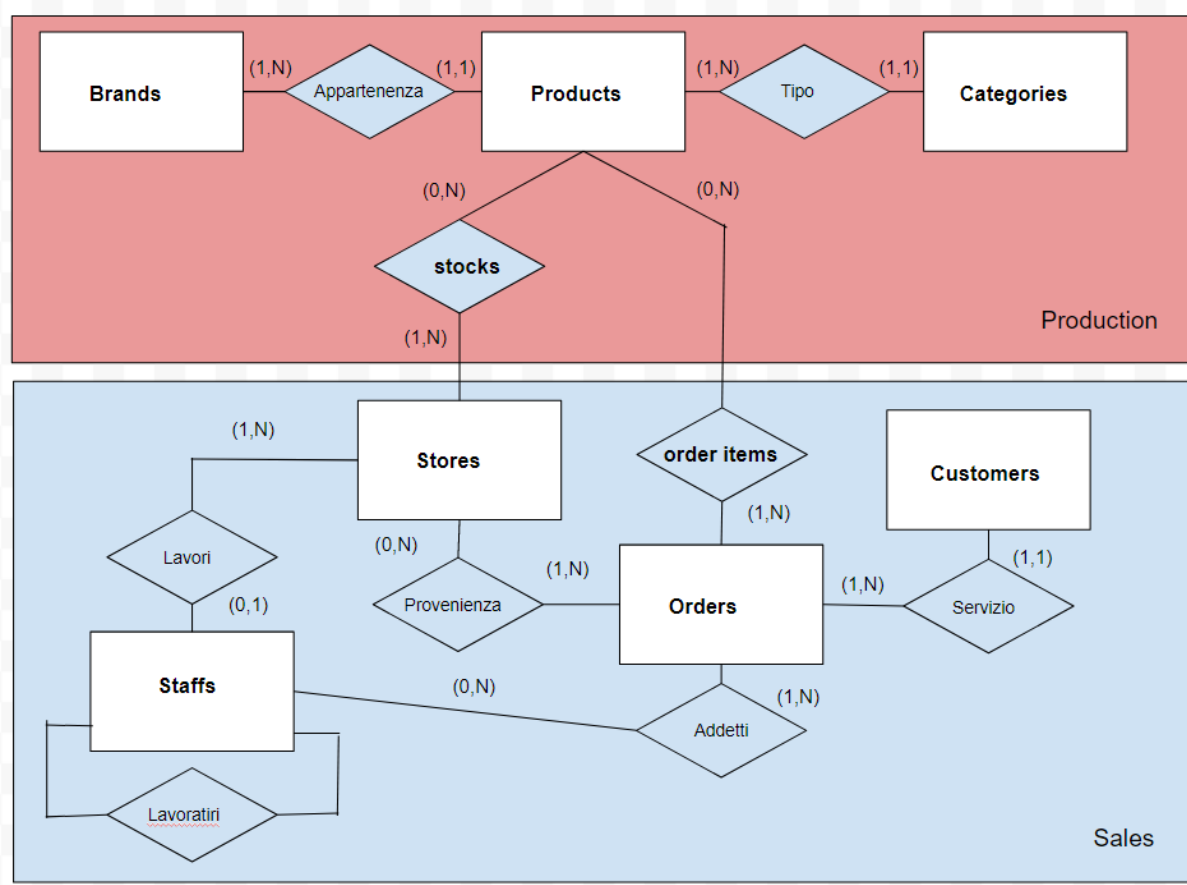
```

## Reverse engineering

Quando si vuole scrivere una query:

1. Individuare in quali tabelle si trovano le informazioni;  
se le info si trovano in più tabelle si fa una join individuando i campi che hanno in relazione
2. Visualizzare i campi richiesti dal testo all'interno della select (Selezione verticale, prendo solo alcuni campi o colonne)
3. Se non sono necessari tutti i record ma solo alcuni, utilizzare la clausola **WHERE** (selezione orizzontale, prendo solo alcuni record o righe della tabella)

## Esempio di esercizi: Schema entità-relazione del database



### Modello logico dello schema

#### 1. i nomi dei brand

```
select brand_name from production.brands
```

	brand_name
1	Electra
2	Haro
3	Heller
4	Pure Cycles
5	Ritchey
6	Strider
7	Sun Bicycles
8	Surly
9	Trek

#### 2. i nomi dei prodotti con il brand

```
select production.product_name,
production.brands.brand_name
from production.products
```

```
inner join production.brands
on production.brands.brand_id = production.products.brand_id
```

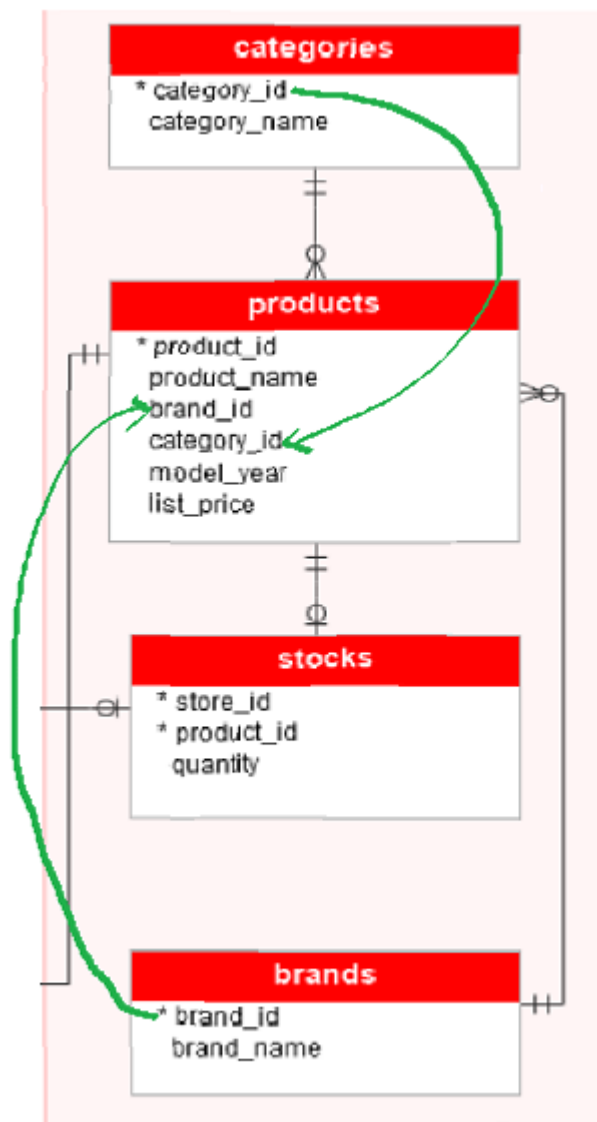
	product_name	brand_name
1	Trek 820 - 2016	Trek
2	Ritchey Timberwolf Frameset - 2016	Ritchey
3	Surly Wednesday Frameset - 2016	Surly
4	Trek Fuel EX 8 29 - 2016	Trek
5	Heller Shagamaw Frame - 2016	Heller
6	Surly Ice Cream Truck Frameset - 2016	Surly
7	Trek Slash 8 27.5 - 2016	Trek
8	Trek Remedy 29 Carbon Frameset - 2016	Trek
9	Trek Conduit+ - 2016	Trek
10	Surly Straggler - 2016	Surly
11	Surly Straggler 650b - 2016	Surly
12	Electra Townie Original 21D - 2016	Electra
13	Electra Cruiser 1 (24-Inch) - 2016	Electra
14	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	Electra
15	Electra Moto 1 - 2016	Electra
16	Electra Townie Original 7D EQ - 2016	Electra
17	Pure Cycles Vine 8-Speed - 2016	Pure Cycles
18	Pure Cycles Western 3-Speed - Women's - 2015/2016	Pure Cycles
19	Pure Cycles William 3-Speed - 2016	Pure Cycles

3. i nomi dei prodotti con il brand e la categoria a cui appartiene il prodotto

```
select production.product_name,
production.brands.brand_name,
production.categories.category_name,
from production.products
```

```
inner join production.brands
on production.brands.brand_id = production.products.brand_id
```

```
inner join production.categories
on production.categories.category_id =
production.products.category_id
```





su che tabella vuoi andare a prendere quelle colonne

3. i nomi dei prodotti con il brand e la categoria a cui appartiene il prodotto

```
select production.product_name,
production.brands.brand_name,
production.categories.category_name,
from production.products
```

Quali colonne vuoi prendere

```
inner join production.brands
on production.brands.brand_id = production.products.brand_id
inner join production.categories
on production.categories.category_id =
production.products.category_id
```

Inner join praticamente ti dice quale tabella vuoi appiccicare a quella scelta

e trova i valori che hanno in comune

	product_name	brand_name	category_name
1	Trek 820 - 2016	Trek	Mountain Bikes
2	Ritchey Timberwolf Frameset - 2016	Ritchey	Mountain Bikes
3	Surly Wednesday Frameset - 2016	Surly	Mountain Bikes
4	Trek Fuel EX 8 29 - 2016	Trek	Mountain Bikes
5	Heller Shagamaw Frame - 2016	Heller	Mountain Bikes
6	Surly Ice Cream Truck Frameset - 2016	Surly	Mountain Bikes
7	Trek Slash 8 27.5 - 2016	Trek	Mountain Bikes
8	Trek Remedy 29 Carbon Frameset - 2016	Trek	Mountain Bikes
9	Trek Conduit+ - 2016	Trek	Electric Bikes
10	Surly Straggler - 2016	Surly	Cyclocross Bicycles
11	Surly Straggler 650b - 2016	Surly	Cyclocross Bicycles
12	Electra Townie Original 21D - 2016	Electra	Cruisers Bicycles
13	Electra Cruiser 1 (24-Inch) - 2016	Electra	Cruisers Bicycles
14	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	Electra	Cruisers Bicycles
15	Electra Moto 1 - 2016	Electra	Cruisers Bicycles
16	Electra Townie Original 7D EQ - 2016	Electra	Cruisers Bicycles
17	Pure Cycles Vine 8-Speed - 2016	Pure Cycles	Cruisers Bicycles
18	Pure Cycles Western 3-Speed - Women's - 2015/2016	Pure Cycles	Cruisers Bicycles
19	Pure Cycles William 3-Speed - 2016	Pure Cycles	Cruisers Bicycles

4. i nomi dei prodotti con le quantità disponibili in stock

```
select product_name, quantity
from production.products inner join production.stocks
```

```
on production.stocks.product_id =  
production.products.product_id
```

	product_name	quantity
1	Trek 820 - 2016	27
2	Trek 820 - 2016	14
3	Trek 820 - 2016	14
4	Ritchey Timberwolf Frameset - 2016	24
5	Ritchey Timberwolf Frameset - 2016	16
6	Ritchey Timberwolf Frameset - 2016	5
7	Surly Wednesday Frameset - 2016	6
8	Surly Wednesday Frameset - 2016	28
9	Surly Wednesday Frameset - 2016	0
10	Trek Fuel EX 8 29 - 2016	11
11	Trek Fuel EX 8 29 - 2016	2
12	Trek Fuel EX 8 29 - 2016	23
13	Heller Shagawaw Frame - 2016	22
14	Heller Shagawaw Frame - 2016	1
15	Heller Shagawaw Frame - 2016	3
16	Surly Ice Cream Truck Frameset - 2016	27
17	Surly Ice Cream Truck Frameset - 2016	11
18	Surly Ice Cream Truck Frameset - 2016	0
19	Trek Slash 8 27.5 - 2016	8

5. i nomi dei prodotti che hanno uno stock superiore a 10

```
SELECT production.products.product_name  
FROM production.products
```

```
INNER JOIN production.stocks  
ON production.products.product_id =  
production.stocks.product_id
```

```
WHERE production.stocks.quantity > 10
```

Risultati		Messaggi
	product_name	
1	Trek 820 - 2016	
2	Trek 820 - 2016	
3	Trek 820 - 2016	
4	Ritchey Timberwolf Frameset - 2016	
5	Ritchey Timberwolf Frameset - 2016	
6	Surly Wednesday Frameset - 2016	
7	Trek Fuel EX 8 29 - 2016	
8	Trek Fuel EX 8 29 - 2016	
9	Heller Shagamaw Frame - 2016	
10	Surly Ice Cream Truck Frameset - 2016	
11	Surly Ice Cream Truck Frameset - 2016	
12	Trek Slash 8 27.5 - 2016	
13	Trek Remedy 29 Carbon Frameset - 2016	
14	Trek Conduit+ - 2016	
15	Trek Conduit+ - 2016	
16	Trek Conduit+ - 2016	
17	Surly Straggler - 2016	
18	Surly Straggler - 2016	
19	Surly Straggler - 2016	

6. i nomi dei prodotti che hanno uno stock compreso tra 5 e 10

```
SELECT production.products.product_name FROM
production.products INNER JOIN production.stocks ON
production.products.product_id = production.stocks.product_id
WHERE production.stocks.quantity BETWEEN 5 AND 10
```

Risultati		Messaggi
	product_name	
1	Ritchey Timberwolf Frameset - 2016	
2	Surly Wednesday Frameset - 2016	
3	Trek Slash 8 27.5 - 2016	
4	Trek Slash 8 27.5 - 2016	
5	Surly Straggler 650b - 2016	
6	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	
7	Electra Moto 1 - 2016	
8	Pure Cycles Western 3-Speed - Women's - 2015/2016	
9	Electra Cruiser 1 (24-Inch) - 2016	
10	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	
11	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	
12	Electra Townie Original 21D - 2016	
13	Electra Townie Original 7D - 2015/2016	
14	Surly Big Dummy Frameset - 2017	
15	Surly Wednesday - 2017	
16	Surly Wednesday - 2017	
17	Surly Wednesday Frameset - 2017	
18	Surly Wednesday Frameset - 2017	
19	Trek Session DH 27.5 Carbon Frameset - 2017	

7. i nomi dei prodotti che hanno uno stock compreso tra 5 e 10 e che costano meno di 300 \$

```
SELECT production.products.product_name,
sales.order_items.list_price,
production.stocks.product_id_quantity
FROM production.products
```

```
INNER JOIN production.stocks
ON production.products.product_id =
production.stocks.product_id
```

```
WHERE production.stocks.quantity BETWEEN 5 AND 10 AND
production.products.list_price < 300
```

Risultati		Messaggi
	product_name	
1	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	
2	Electra Cruiser 1 (24-Inch) - 2016	
3	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	
4	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	
5	Sun Bicycles Revolutions 24 - Girl's - 2017	
6	Sun Bicycles Revolutions 24 - Girl's - 2017	
7	Trek Precaliber 12 Girls - 2017	
8	Trek Precaliber 12 Girls - 2017	
9	Trek Precaliber 16 Boys - 2017	
10	Trek Precaliber 16 Boys - 2017	
11	Trek Precaliber 16 Girls - 2017	
12	Haro Shredder Pro 20 - 2017	
13	Electra Girl's Hawaii 1 16" - 2017	
14	Electra Cruiser 1 - 2016/2017/2018	
15	Electra Cruiser 1 - 2016/2017/2018	
16	Electra Cruiser 1 Ladies' - 2018	
17	Trek MT 201 - 2018	
18	Trek MT 201 - 2018	
19	Strider Sport 16 - 2018	

8. i nomi dei prodotti che hanno la parola "Cruiser" nel nome

```
SELECT production.products.product_name
FROM production.products
WHERE production.products.product_name LIKE '%Cruiser%'
```

Risultati		Messaggi
	product_name	
1	Electra Cruiser 1 (24-Inch) - 2016	
2	Electra Cruiser 1 (24-Inch) - 2016	
3	Electra Cruiser Lux 1 - 2017	
4	Electra Cruiser Lux Fat Tire 1 Ladies - 2017	
5	Electra Cruiser 1 - 2016/2017/2018	
6	Electra Cruiser 7D - 2016/2017/2018	
7	Electra Cruiser 1 Ladies' - 2018	
8	Electra Cruiser 7D Ladies' - 2016/2018	
9	Electra Cruiser 1 Tall - 2016/2018	
10	Electra Cruiser Lux 3i - 2018	
11	Electra Cruiser Lux 7D - 2018	
12	Electra Cruiser 7D (24-Inch) Ladies' - 2016/2018	
13	Electra Cruiser 7D Tall - 2016/2018	
14	Electra Cruiser Lux 1 - 2016/2018	
15	Electra Cruiser Lux 1 Ladies' - 2018	
16	Electra Cruiser Lux 3i Ladies' - 2018	
17	Electra Cruiser Lux 7D Ladies' - 2018	
18	Electra Cruiser Lux Fat Tire 7D - 2018	
19	Electra Cruiser 7D (24-Inch) Ladies' - 2016/2018	

9. i nomi dei prodotti che non hanno la parola "Cruiser" nel nome

```
SELECT production.products.product_name
FROM production.products
WHERE production.products.product_name NOT LIKE '%Cruiser%'
```

Risultati

Messaggi

	product_name
1	Trek 820 - 2016
2	Ritchey Timberwolf Frameset - 2016
3	Surly Wednesday Frameset - 2016
4	Trek Fuel EX 8 29 - 2016
5	Heller Shagamaw Frame - 2016
6	Surly Ice Cream Truck Frameset - 2016
7	Trek Slash 8 27.5 - 2016
8	Trek Remedy 29 Carbon Frameset - 2016
9	Trek Conduit+ - 2016
10	Surly Straggler - 2016
11	Surly Straggler 650b - 2016
12	Electra Townie Original 21D - 2016
13	Electra Girl's Hawaii 1 (16-inch) - 2015/2016
14	Electra Moto 1 - 2016
15	Electra Townie Original 7D EQ - 2016
16	Pure Cycles Vine 8-Speed - 2016
17	Pure Cycles Western 3-Speed - Women's - 2015/2016
18	Pure Cycles William 3-Speed - 2016
19	Electra Townie Original 7D EQ - Women's - 2016

Groupby

```
select model_year, count(*) from production.products
group by model_year
```

Risultati		Messaggi
	model_year	(Nessun nome di colonna)
1	2016	26
2	2017	85
3	2018	204
4	2019	6

**Where** non funzionerà con il **groupby** ma si utilizzerà **having**

```

select model_year, count(*) as totale_prodotti from production.products
group by model_year
having count(*) > 50
order by totale_prodotti desc

```

	model_year	totale_prodotti
1	2018	204
2	2017	85

```

select state,city,count(*) as provenienza from sales.customers
group by state,city
having count(*) >10
order by provenienza

```

	state	city	provenienza
10	CA	South El Monte	11
11	CA	Upland	11
12	NY	Webster	12
13	NY	Sunnyside	12
14	NY	Merick	12
15	NY	Smithtown	12
16	NY	Richmond Hill	12
17	NY	Ronkonkoma	12
18	TX	San Angelo	12
19	CA	Canyon Country	12
20	NY	Howard Beach	12
21	NY	Astoria	12
22	NY	Floral Park	13
23	TX	Longview	13
24	NY	Ossining	13
25	NY	Canandaigua	14
26	NY	Ballston Spa	17
27	NY	Scarsdale	17
28	NY	Mount Vernon	20

somma dei prezzi dei prodotti di ogni categoria



```

select category_name, sum(list_price) as somma_totale from production.products
inner join production.categories
on production.products.category_id = production.categories.category_id

group by products.category_id, category_name
having sum(list_price) > 50000
order by sum(list_price) desc

```

	category_name	somma_totale
1	Road Bikes	190521.44
2	Mountain Bikes	98985.44
3	Electric Bikes	78759.76
4	Cruisers Bicycles	56972.16

raggruppiamo category id ma non lo mostriamo

```

--i prodotti acquistati dal cliente Martin Spencer
Select production.products.*
from sales.customers

inner join sales.orders
on sales.customers.customer_id = sales.orders.customer_id

inner join sales.order_items
on sales.orders.order_id = sales.orders.order_id
--ogni volta che facciamo l'inner join non si attacca la tabella base
--a quella che vogliamo attaccare ad esso (sono sono tipo a 3 tabelle di distanza)
--ma si fanno le inner join con la tabella precedente aquella interessata
inner join production.products
on production.products.product_id = sales.order_items.product_id

where first_name = 'Johnathan' and last_name = 'Velazquez'

```

**Viste**

**Es:**

```
CREATE VIEW sales.product_info
AS
SELECT
    product_name,
    brand_name,
    list_price
FROM
    production.products p
INNER JOIN production.brands b
    ON b.brand_id = p.brand_id;
```