

## Esercitiamoci<sup>1</sup> con uso SQL SERVER

- Creare / popolare / eliminare tabelle (account somee.com)
- Uso di funzioni: SUM, COUNT, AVG, MAX, MIN

### Creazione tabelle senza PK:

<pre>CREATE TABLE Persons (   PersonID int,   LastName varchar(255),   FirstName varchar(255),   Address varchar(255),   City varchar(255) )</pre>	<pre>CREATE TABLE Ordini (   id int,   id_cliente int,   prezzo int,   prodotto varchar(255) )</pre>
--	--

### Popoliamo tabella Persons

INSERT INTO Persons VALUES (1,'Perla','Giada', 'via Giotto 7', 'Milano')

Select \* FROM Persons

PersonID	LastName	FirstName	Address	City
1	Perla	Giada	via Giotto 7	Milano

### Cancllando DROP TABLE [Persons] oppure DROP TABLE Persons

SQL\_SERVER non DROP Table IF EXISTS *nomeTab* ma  
**IF OBJECT\_ID ('*nomeTab*') IS NOT NULL DROP TABLE *nomeTab***

### Reinserendo

INSERT INTO Persons VALUES (2,'Dori','Marco', 'via Pisa 6', 'Genova')

Select \* FROM Persons

PersonID	LastName	FirstName	Address	City
1	Perla	Giada	via Giotto 7	Milano
2	Dori	Marco	via Pisa 6	Genova

### Aggiorniamo

```
UPDATE table_name
SET column1=value1,column2=value2,...
WHERE some_column=some_value;
```

UPDATE Persons SET LastName='Perlo' WHERE PersonID=1

Select \* FROM Persons

PersonID	LastName	FirstName	Address	City
1	Perlo	Giada	via Giotto 7	Milano
2	Dori	Marco	via Pisa 6	Genova

### Equivalenza:

<pre>SELECT LastName FROM Persons GROUP BY LastName ORDER BY LastName ASC;</pre>	<pre>SELECT DISTINCT LastName FROM Persons ORDER BY LastName ASC;</pre>
--	---

Nella sua forma di utilizzo più semplice la clausola **GROUP BY** produce un risultato analogo a **SELECT DISTINCT**.

LastName
Dori
Perlo

**NB:** Si ricordi che tutte le colonne che compaiono accanto alla parola SELECT devono essere incluse nella clausola **GROUP BY** oppure devono essere argomenti di una funzione di aggregazione

<sup>1</sup> Esercizi tratti da [http://www.mrwebmaster.it/sql/group-by-having\\_11991.html](http://www.mrwebmaster.it/sql/group-by-having_11991.html)

Popoliamo la tabella **ORDINI** (usando ; è possibile **eseguire come unico statement** - pulsante **Run single query**)

MS SQL Database: dbDesign

```
INSERT INTO Ordini VALUES (1, 1, 50, 'Scarpe');
INSERT INTO Ordini VALUES (2, 1,1500, 'Vestito');
INSERT INTO Ordini VALUES (3, 2, 40, 'Scarpe');
INSERT INTO Ordini VALUES (4, 2, 500, 'Vestito');
INSERT INTO Ordini VALUES (5, 1, 60, 'Camicia');
INSERT INTO Ordini VALUES (6, 1,70, 'Cravatta');
```

```
INSERT INTO Ordini VALUES (1, 1, 50, 'Scarpe');
INSERT INTO Ordini VALUES (2, 1,1500, 'Vestito');
INSERT INTO Ordini VALUES (3, 2, 40, 'Scarpe');
INSERT INTO Ordini VALUES (4, 2, 500, 'Vestito');
INSERT INTO Ordini VALUES (5, 1, 60, 'Camicia');
INSERT INTO Ordini VALUES (6, 1,70, 'Cravatta');
```

select \* from Ordini

id	id_cliente	prezzo	prodotto
1	1	50	Scarpe
2	1	1500	Vestito
3	2	40	Scarpe
4	2	500	Vestito
5	1	60	Camicia
6	1	70	Cravatta

Quindi:

1 Perlo scarpe 50 + vestito 1500 + camicia 60 + cravatta 70  
 2 Dori scarpe 40 + vestito 500

**ATTENZIONE<sup>2</sup>:**

```
SELECT id_cliente, SUM(prezzo) AS spesa
FROM ordini
GROUP BY id_cliente
HAVING spesa >= 1000; /* Errore ALIAS in SQL SERVER è inteso come testo */
```

```
SELECT id_cliente, SUM (prezzo) AS spesa
FROM ordini
GROUP BY id_cliente
Having SUM(prezzo)>=1000 /* interi */
```

id_cliente	spesa
1	1680

```
SELECT id_cliente, Count(id) AS acquisti, SUM(prezzo) AS spesa
FROM ordini
GROUP BY id_cliente
HAVING spesa >= 1000 OR acquisti >= 2; /* spesa e acquisti non interi */
```

```
SELECT id_cliente, Count (id) AS acquisti, SUM (prezzo) AS spesa
FROM ordini
GROUP BY id_cliente
HAVING SUM (prezzo) >= 1000 OR Count(id) >= 2; /* interi */
```

id_cliente	acquisti	spesa
1	4	1680
2	2	540

La clausola **Where** (applicabile a righe single) non può essere usata con funzioni aggregate

In alcuni casi può essere necessario escludere singole righe dai gruppi (utilizzando una clausola **WHERE**) prima di applicare una **condizione ai gruppi** (utilizzando una clausola **HAVING**).

La clausola **HAVING** può essere applicata solo alle **colonne presenti anche nella clausola GROUP BY** o in una **funzione di aggregazione**

```
SELECT LastName, Count (id) AS acquisti, SUM (prezzo) AS spesa
FROM ordini, Persons
Where id_cliente = PersonID
GROUP BY LastName
HAVING (SUM (prezzo) >= 1000) OR (Count (id) >= 2);
```

LastName	acquisti	spesa
Dori	2	540
Perlo	4	1680

<sup>2</sup> Confrontare nell'uso di [phpMyAdmin](http://phpmyadmin.org) gestendo DBMS MariaDB (nato da un fork di MySQL)



```
SELECT id_cliente, MAX(prezzo) AS spesa_massima
FROM ordini
GROUP BY id_cliente;
```

id_cliente	spesa_massima
1	1500
2	500

```
SELECT LastName, MAX(prezzo) AS spesa_massima
FROM ordini, Persons
Where id_cliente = PersonID
GROUP BY LastName
```

LastName	spesa_massima
Dori	500
Perlo	1500

```
SELECT LastName, MIN(prezzo) AS spesa_minima
FROM ordini, Persons
Where id_cliente = PersonID
GROUP BY LastName
```

LastName	spesa_minima
Dori	40
Perlo	50

```
SELECT LastName, AVG(prezzo) AS spesa_media
FROM ordini, Persons
Where id_cliente = PersonID
GROUP BY LastName
```

LastName	spesa_media
Dori	270
Perlo	420

```
SELECT LastName, AVG(prezzo) AS spesa_media
FROM ordini, Persons
Where id_cliente = PersonID
GROUP BY LastName
ORDER BY spesa_media DESC;
```

LastName	spesa_media
Perlo	420
Dori	270

### Altri raggruppamenti ed ordinamenti

```
SELECT prodotto, prezzo
FROM ordini
GROUP BY prodotto, prezzo
ORDER BY prodotto ASC;
```

prodotto	prezzo
Camicia	60
Cravatta	70
Scarpe	40
Scarpe	50
Vestito	500
Vestito	1500

```
SELECT prodotto, prezzo
FROM ordini
GROUP BY prodotto, prezzo
ORDER BY prezzo DESC;
```

prodotto	prezzo
Vestito	1500
Vestito	500
Cravatta	70
Camicia	60
Scarpe	50
Scarpe	40

```
SELECT prodotto, prezzo
FROM ordini
GROUP BY prodotto, prezzo
ORDER BY prezzo ASC;
```

prodotto	prezzo
Scarpe	40
Scarpe	50
Camicia	60
Cravatta	70
Vestito	500
Vestito	1500

### Attuali tabelle

name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date
Persons	174623665		1	0	U	USER_TABLE	12/13/2015 3:40:34 AM	12/13/2015 3:40:34 AM
Elenco	245575913		1	0	U	USER_TABLE	11/25/2015 11:52:37 PM	11/25/2015 11:52:37 PM
Amici	277576027		1	0	U	USER_TABLE	11/25/2015 11:52:58 PM	11/25/2015 11:52:58 PM
Ordini	430624577		1	0	U	USER_TABLE	12/13/2015 4:11:19 AM	12/13/2015 4:11:19 AM