

Utilizzo della SELECT per recuperare dati
Filtrare e ordinare i risultati di una Query

AGENDA

- ☐ L'istruzione SELECT
- ☐ L'istruzione FROM
- ☐ L'istruzione WHERE
- ☐ L'istruzione ORDER BY

Interrogare una base dati significa innanzitutto saper recuperare tutte le informazioni utili da una particolare tabella e, restituire un risultato utile, chiaro e parlante!

Alla fine di questo modulo sarai in grado di:

- ✓ Recuperare i campi utili da una particolare tabella
- ✓ Scrivere campi calcolati
- ✓ Filtrare e ordinare i risultati di una query
- ✓ Capire la logica sottostante le query

Cosa impererai

Interrogare una base dati significa innanzitutto saper recuperare tutte le informazioni utili da una particolare tabella e, restituire un risultato utile, chiaro e parlante!

Alla fine di questo modulo sarai in grado di:

- ✓ Recuperare i campi utili da una particolare tabella
- ✓ Scrivere campi calcolati
- ✓ Filtrare e ordinare i risultati di una query
- ✓ Capire la logica sottostante le query

SQL: **SELECT** *campo1, campo2,..* **FROM** *nome_tabella*

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante) al fine di esporre tutte i campi e i record disponibili.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT *
FROM Product
```



	ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
1	680	HL Road Frame - Black, 58	1431,50	1059,31	58	Black	NULL
2	706	HL Road Frame - Red, 58	1431,50	1059,31	58	Red	NULL
3	707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
4	708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
5	709	Mountain Bike Socks, M	9,50	3,3963	M	White	2006-06-30
6	710	Mountain Bike Socks, L	9,50	3,3963	L	White	2006-06-30
7	711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
8	712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
9	713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
10	714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

1. **FROM** : consente di specificare la tabella (o le tabelle) da interrogare
2. **SELECT** : consente di indicare quali colonne devono essere restituite. Il carattere * significa 'tutte le colonne' (anziché specificare tutte le colonne della tabella)

SQL: **SELECT** *campo1, campo2,..* **FROM** *nome_tabella*

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante) al fine di esporre i soli campi: **Name**, **StandardCost**, **ListPrice**

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT Name, StandardCost, ListPrice
FROM Product
```



	Name	ListPrice	StandardCost
1	HL Road Frame - Black, 58	1431,50	1059,31
2	HL Road Frame - Red, 58	1431,50	1059,31
3	Sport-100 Helmet, Red	34,99	13,0863
4	Sport-100 Helmet, Black	34,99	13,0863
5	Mountain Bike Socks, M	9,50	3,3963
6	Mountain Bike Socks, L	9,50	3,3963
7	Sport-100 Helmet, Blue	34,99	13,0863
8	AWC Logo Cap	8,99	6,9223
9	Long-Sleeve Logo Jersey, S	49,99	38,4923
10	Long-Sleeve Logo Jersey, M	49,99	38,4923
11	Long-Sleeve Logo Jersey, L	49,99	38,4923

1. **FROM** : consente di specificare la tabella (o le tabelle) da interrogare
2. **SELECT** : consente di indicare quali colonne devono essere restituite. Il carattere * significa 'tutte le colonne' (anziché specificare tutte le colonne della tabella)

SQL: **SELECT** *campo1, campo2,..* **FROM** *nome_tabella*

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante) al fine di esporre i campi: Name, StandardCost, ListPrice.

Supponiamo di dover esporre in un campo calcolato la differenza tra ListPrice e StandardCost (Markup)

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT Name, StandardCost, ListPrice,  
ListPrice - StandardCost  
FROM Product
```



	Name	ListPrice	StandardCost	(No column name)
1	HL Road Frame - Black, 58	1431,50	1059,31	372,19
2	HL Road Frame - Red, 58	1431,50	1059,31	372,19
3	Sport-100 Helmet, Red	34,99	13,0863	21,9037
4	Sport-100 Helmet, Black	34,99	13,0863	21,9037
5	Mountain Bike Socks, M	9,50	3,3963	6,1037
6	Mountain Bike Socks, L	9,50	3,3963	6,1037
7	Sport-100 Helmet, Blue	34,99	13,0863	21,9037
8	AWC Logo Cap	8,99	6,9223	2,0677
9	Long-Sleeve Logo Jersey, S	49,99	38,4923	11,4977
10	Long-Sleeve Logo Jersey, M	49,99	38,4923	11,4977
11	Long-Sleeve Logo Jersey, L	49,99	38,4923	11,4977

1. **FROM** : consente di specificare la tabella (o le tabelle) da interrogare
2. **SELECT** : consente di indicare quali colonne devono essere restituite. Consente di includere anche espressioni (campi calcolati ad esempio)

SQL: ALIAS

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante) al fine di esporre i campi: Name, StandardCost, ListPrice.

Supponiamo di dover esporre in un campo calcolato la differenza tra ListPrice e StandardCost (Markup)

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT Name, StandardCost, ListPrice,  
ListPrice - StandardCost AS Markup  
FROM Product
```



	Name	ListPrice	StandardCost	Markup
1	HL Road Frame - Black, 58	1431,50	1059,31	372,19
2	HL Road Frame - Red, 58	1431,50	1059,31	372,19
3	Sport-100 Helmet, Red	34,99	13,0863	21,9037
4	Sport-100 Helmet, Black	34,99	13,0863	21,9037
5	Mountain Bike Socks, M	9,50	3,3963	6,1037
6	Mountain Bike Socks, L	9,50	3,3963	6,1037
7	Sport-100 Helmet, Blue	34,99	13,0863	21,9037
8	AWC Logo Cap	8,99	6,9223	2,0677
9	Long-Sleeve Logo Jersey, S	49,99	38,4923	11,4977
10	Long-Sleeve Logo Jersey, M	49,99	38,4923	11,4977
11	Long-Sleeve Logo Jersey, L	49,99	38,4923	11,4977

La Keyword **AS** puo' essere utilizzata per assegnare degli *alias* a ciascun colonna del *result set*.

La keyword potrebbe essere anche omessa (ma poi il codice risulterebbe meno leggibile...)

SQL: ALL

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante) al fine di esporre i campi: Name, StandardCost, ListPrice.

Supponiamo di dover esporre in un campo calcolato la differenza tra ListPrice e StandardCost (Markup)

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT Name, StandardCost, ListPrice,  
ListPrice - StandardCost  
FROM Product
```



	Color
1	Black
2	Red
3	Red
4	Black
5	White
6	White
7	Blue
8	Multi
9	Multi
10	Multi

- Per impostazione predefinita, la clausola SELECT include una parola chiave ALL ☐ vengono recuperate tutte le righe

SQL: ALL

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante) al fine di esporre l'elenco dei diversi colori disponibili

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT Color  
FROM Product
```

```
SELECT ALL Color  
FROM Product
```



	Color
1	Black
2	Red
3	Red
4	Black
5	White
6	White
7	Blue
8	Multi
9	Multi
10	Multi

Per impostazione predefinita, la clausola **SELECT** sottintende la keyword **ALL** (di solito si omette) che include i duplicati.

In altre parole, vengono recuperate tutte le righe compresi i duplicati!

SQL: DISTINCT

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante) al fine di esporre l'elenco dei diversi colori disponibili

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT DISTINCT Color  
FROM Product
```



	Color
1	Black
2	Blue
3	Multi
4	Red
5	White

È possibile utilizzare la keyword **DISTINCT** che consente di rimuovere i duplicati.

In altre parole, introducendo **DISTINCT** nell'istruzione **SELECT** è possibile includere i valori univoci.

SQL: ORDER BY

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti ordinati per **ListPrice** crescente. Il result set deve contenere i campi **ProductID**, **Name**, **ListPrice**

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice
FROM Product
ORDER BY ListPrice ASC
```



	ProductID	Name	ListPrice
1	712	AWC Logo Cap	8,99
2	709	Mountain Bike Socks, M	9,50
3	710	Mountain Bike Socks, L	9,50
4	711	Sport-100 Helmet, Blue	34,99
5	707	Sport-100 Helmet, Red	34,99
6	708	Sport-100 Helmet, Black	34,99
7	713	Long-Sleeve Logo Jersey, S	49,99
8	714	Long-Sleeve Logo Jersey, M	49,99
9	680	HL Road Frame - Black, 58	1431,50
10	706	HL Road Frame - Red, 58	1431,50

L'istruzione **ORDER BY** consente di ordinare il result set. La Keyword **ASC** consente di specificare il tipo di ordinamento: crescente (ASC) o decrescente (DESC). Di default, l'ordinamento è crescente per cui la keyword ASC puo' essere, in questo caso, omessa.

SQL: ORDER BY

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti ordinati per **ListPrice** crescente. Il result set deve contenere i campi **ProductID**, **Name**, **ListPrice**

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice
FROM Product
ORDER BY ListPrice
```



	ProductID	Name	ListPrice
1	712	AWC Logo Cap	8,99
2	709	Mountain Bike Socks, M	9,50
3	710	Mountain Bike Socks, L	9,50
4	711	Sport-100 Helmet, Blue	34,99
5	707	Sport-100 Helmet, Red	34,99
6	708	Sport-100 Helmet, Black	34,99
7	713	Long-Sleeve Logo Jersey, S	49,99
8	714	Long-Sleeve Logo Jersey, M	49,99
9	680	HL Road Frame - Black, 58	1431,50
10	706	HL Road Frame - Red, 58	1431,50

L'istruzione **ORDER BY** consente di ordinare il result set. La Keyword **ASC** consente di specificare il tipo di ordinamento: crescente (ASC) o decrescente (DESC). Di default, l'ordinamento è crescente per cui la keyword ASC puo' essere, in questo caso, omessa.

SQL: ORDER BY

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti ordinati per **ListPrice** decrescente. Il result set deve contenere i campi **ProductID**, **Name**, **ListPrice**

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice
FROM Product
ORDER BY ListPrice DESC
```



	ProductID	Name	ListPrice
1	680	HL Road Frame - Black, 58	1431,50
2	706	HL Road Frame - Red, 58	1431,50
3	713	Long-Sleeve Logo Jersey, S	49,99
4	714	Long-Sleeve Logo Jersey, M	49,99
5	711	Sport-100 Helmet, Blue	34,99
6	707	Sport-100 Helmet, Red	34,99
7	708	Sport-100 Helmet, Black	34,99
8	709	Mountain Bike Socks, M	9,50
9	710	Mountain Bike Socks, L	9,50
10	712	AWC Logo Cap	8,99

L'istruzione **ORDER BY** consente di ordinare il result set. Di default, l'ordinamento è crescente. Se è necessario un ordinamento decrescente bisogna necessariamente specificarlo utilizzando la keyword **DESC**.

SQL: ORDER BY

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti ordinati per **Markup decrescente**.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice
FROM Product
ORDER BY ListPrice DESC, Name ASC
```



	ProductID	Name	ListPrice
1	680	HL Road Frame - Black, 58	1431,50
2	706	HL Road Frame - Red, 58	1431,50
3	714	Long-Sleeve Logo Jersey, M	49,99
4	713	Long-Sleeve Logo Jersey, S	49,99
5	708	Sport-100 Helmet, Black	34,99
6	711	Sport-100 Helmet, Blue	34,99
7	707	Sport-100 Helmet, Red	34,99
8	710	Mountain Bike Socks, L	9,50
9	709	Mountain Bike Socks, M	9,50
10	712	AWC Logo Cap	8,99

Il resul set è ordinato per **ListPrice** decrescente, e ogni set di prodotti con lo stesso Prezzo sono in ordine crescente per **Name** (ordine alfabetico dalla A alla Z essendo un testo; se avessimo scritto DESC avremmo ottenuto l'ordinamento alfabetico dalla Z alla A)

SQL: ORDER BY

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti ordinati per **Markup decrescente**.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT Name, StandardCost, ListPrice,  
ListPrice - StandardCost AS Markup  
FROM Product  
ORDER BY Markup DESC
```



	Name	StandardCost	ListPrice	Markup
1	HL Road Frame - Black, 58	1059,31	1431,50	372,19
2	HL Road Frame - Red, 58	1059,31	1431,50	372,19
3	Sport-100 Helmet, Red	13,0863	34,99	21,9037
4	Sport-100 Helmet, Black	13,0863	34,99	21,9037
5	Sport-100 Helmet, Blue	13,0863	34,99	21,9037
6	Long-Sleeve Logo Jersey, S	38,4923	49,99	11,4977
7	Long-Sleeve Logo Jersey, M	38,4923	49,99	11,4977
8	Mountain Bike Socks, M	3,3963	9,50	6,1037
9	Mountain Bike Socks, L	3,3963	9,50	6,1037
10	AWC Logo Cap	6,9223	8,99	2,0677

È possibile indicare nell'istruzione **ORDER BY** l'alias assegnato al campo calcolato perchè l'istruzione **ORDER BY** viene processata dopo l'istruzione **SELECT**: viene processata prima l'istruzione **FROM** poi l'istruzione **SELECT** poi l'istruzione **ORDER BY**.

SQL: WHERE - condizione di ricerca.. Come filtrare il risultato della query!

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti per cui il colore sia *red*.

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Color = 'Red'
```



	ProductID	NAME	ListPrice	Color
1	706	HL Road Frame - Red, 58	1431,50	Red
2	707	Sport-100 Helmet, Red	34,99	Red

L'istruzione **WHERE** consente di filtrare le righe utilizzando un predicato. Il predicato è un filtro o, più formalmente, una condizione di ricerca in base alla quale filtrare le righe restituite dalla **FROM**.

Verranno restituite tutte le righe per le quali la condizione di ricerca è valutata come vera!

In questo caso, è utilizzato l'operatore di confronto = che consente di ricercare (e ottenere come risultato) tutti i record per cui è vero il predicato Color = 'red'.

SQL: WHERE - condizione di ricerca.. Come filtrare il risultato della query!

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti per cui il colore sia *red*.

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Color = 'Red'
```

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

1. 'recupera le righe della tabella prodotto' (FROM Product)
2. 'cerca le righe dove Color è uguale a Red' (WHERE Color = 'Red')
3. 'seleziona le colonne ProductID, Name, ListPrice, Color'

L'ordine con cui vengono processate le istruzioni è:
FROM, WHERE, SELECT

SQL: WHERE - condizione di ricerca.. Come filtrare il risultato della query!

Cosa succede 'dietro le quinte' di questa istruzione?

1. recupera le righe della tabella prodotto (FROM Product)

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

2. cerca le righe dove Color è uguale a Red (WHERE Color = 'Red')

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL

3. seleziona le colonne ProductID, Name, ListPrice, Color'

ProductID	Name	ListPrice	Color
706	HL Road Frame - Red, 58	1431,5	Red
707	Sport-100 Helmet, Red	34,99	Red

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Color = 'Red'
```



	ProductID	NAME	ListPrice	Color
1	706	HL Road Frame - Red, 58	1431,50	Red
2	707	Sport-100 Helmet, Red	34,99	Red

SQL: WHERE - condizione di ricerca.. Come filtrare il risultato della query!

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti per cui il prezzo di listino sia maggiore di 34,99.

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE ListPrice > 34.99
```



	ProductID	Name	ListPrice	Color
1	680	HL Road Frame - Black, 58	1431,50	Black
2	706	HL Road Frame - Red, 58	1431,50	Red
3	713	Long-Sleeve Logo Jersey, S	49,99	Multi
4	714	Long-Sleeve Logo Jersey, M	49,99	Multi

SQL: WHERE - condizione di ricerca.. Come filtrare il risultato della query!

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti per cui il prezzo di listino sia maggiore o uguale a 34,99. Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE ListPrice >= 34.99
```



	ProductID	Name	ListPrice	Color
1	680	HL Road Frame - Black, 58	1431,50	Black
2	706	HL Road Frame - Red, 58	1431,50	Red
3	707	Sport-100 Helmet, Red	34,99	Red
4	708	Sport-100 Helmet, Black	34,99	Black
5	711	Sport-100 Helmet, Blue	34,99	Blue
6	713	Long-Sleeve Logo Jersey, S	49,99	Multi
7	714	Long-Sleeve Logo Jersey, M	49,99	Multi

SQL: WHERE - condizione di ricerca.. Come filtrare il risultato della query!

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti il cui colore è diverso da Multi.

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Color <> 'Multi'
```



	ProductID	Name	ListPrice	Color
1	680	HL Road Frame - Black, 58	1431,50	Black
2	706	HL Road Frame - Red, 58	1431,50	Red
3	707	Sport-100 Helmet, Red	34,99	Red
4	708	Sport-100 Helmet, Black	34,99	Black
5	709	Mountain Bike Socks, M	9,50	White
6	710	Mountain Bike Socks, L	9,50	White
7	711	Sport-100 Helmet, Blue	34,99	Blue

SQL: Operatori di confronto e logici

Operatori di confronto

Operatore	Significato
= (uguale a)	Uguale a
> (maggiore di)	Maggiore di
< (minore di)	Minore di
>= (maggiore o uguale a)	Maggiore o uguale a
<= (minore o uguale a)	Minore o uguale a
<> (diverso da)	Diverso da

Gli operatori di confronto consentono di confrontare due espressioni (non di tipo testo)!

Operatori logici

Operatore	Significato
AND	Restituisce TRUE se entrambe le espressioni booleane sono TRUE.
BETWEEN	Restituisce TRUE se l'operando è compreso nell'intervallo specificato.
IN	Restituisce TRUE se l'operando è uguale a una delle espressioni elencate.
LIKE	Restituisce TRUE se l'operando corrisponde a un modello.
NOT	Inverte il valore degli altri operatori booleani.
OR	Restituisce TRUE se una delle due espressioni booleane è TRUE.

SQL: utilizzo dell'operatore AND

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti il cui colore è diverso da Multi e con un prezzo di listino maggiore di 40.

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Color <> 'Multi' AND ListPrice > 40
```



	ProductID	Name	ListPrice	Color
1	680	HL Road Frame - Black, 58	1431,50	Black
2	706	HL Road Frame - Red, 58	1431,50	Red

Vengono restituiti i record per cui sono vere 'contemporaneamente' entrambi i predicati!

SQL: utilizzo dell'operatore BETWEEN

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti il cui prezzo di listino sia maggiore o uguale a 9,5 e minore o uguale a 49,99. (come dire con un prezzo di listino compreso tra 9,5 e 49,99).

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE ListPrice >= 9.5 AND ListPrice <= 49.95
```

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE ListPrice BETWEEN 9.5 AND 49.99
```



	ProductID	Name	ListPrice	Color
1	707	Sport-100 Helmet, Red	34,99	Red
2	708	Sport-100 Helmet, Black	34,99	Black
3	709	Mountain Bike Socks, M	9,50	White
4	710	Mountain Bike Socks, L	9,50	White
5	711	Sport-100 Helmet, Blue	34,99	Blue
6	713	Long-Sleeve Logo Jersey, S	49,99	Multi
7	714	Long-Sleeve Logo Jersey, M	49,99	Multi

Le query sono equivalenti in termini di risultato perché l'operatore **BETWEEN** include gli estremi del range specificato! È estremamente utile per le date!

SQL: utilizzo dell'operatore OR

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti il cui prezzo di listino sia minore o uguale a 50 oppure il cui colore sia *Red*.

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE ListPrice <= 50 OR Color = 'Red'
```



	ProductID	Name	ListPrice	Color
1	706	HL Road Frame - Red, 58	1431,50	Red
2	707	Sport-100 Helmet, Red	34,99	Red
3	708	Sport-100 Helmet, Black	34,99	Black
4	709	Mountain Bike Socks, M	9,50	White
5	710	Mountain Bike Socks, L	9,50	White
6	711	Sport-100 Helmet, Blue	34,99	Blue
7	712	AWC Logo Cap	8,99	Multi
8	713	Long-Sleeve Logo Jersey, S	49,99	Multi
9	714	Long-Sleeve Logo Jersey, M	49,99	Multi

Le query restituisce tutti i record per cui è vera o l'una o l'altra condizione di ricerca! La query restituisce tutti i prodotti il cui colore sia *Red* (a prescindere dal prezzo) e tutti i prodotti il cui prezzo di listino sia minore o uguale a 50 (a prescindere dal colore)!

SQL: utilizzo dell'operatore IN

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti il cui colore sia incluso nella lista: *Black, Red, White*! Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Color IN ('Black', 'Red', 'White')
```

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Color='Black' OR Color='Red' OR Color='White'
```



	ProductID	Name	ListPrice	Color
1	680	HL Road Frame - Black, 58	1431,50	Black
2	706	HL Road Frame - Red, 58	1431,50	Red
3	707	Sport-100 Helmet, Red	34,99	Red
4	708	Sport-100 Helmet, Black	34,99	Black
5	709	Mountain Bike Socks, M	9,50	White
6	710	Mountain Bike Socks, L	9,50	White

L'operatore **IN** consente di ricercare la corrispondenza in una lista di valori specificati.

L'utilizzo dell'operatore **OR** (tante volte quanti sono i valori di cui cercare una corrispondenza) è equivalente!

SQL: utilizzo dell'operatore NOT

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario esporre l'elenco dei prodotti il cui colore non sia incluso nella lista: *Black, Red, White*! Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Color NOT IN ('Black', 'Red', 'White')
```



	ProductID	Name	ListPrice	Color
1	711	Sport-100 Helmet, Blue	34,99	Blue
2	712	AWC Logo Cap	8,99	Multi
3	713	Long-Sleeve Logo Jersey, S	49,99	Multi
4	714	Long-Sleeve Logo Jersey, M	49,99	Multi

Ricorda che le condizioni di ricerca restituiscono tutti i record per i quali il predicato logico (filtro tramite operatori di confronto e/o logici) risulta vero!

L'operatore **NOT** inverte la logica dell'operatore!

SQL: utilizzo dell'operatore LIKE

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario recuperare tutti i prodotti con nome *Sport-100 Helmet, Blue.**

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

*non è detto che il nome dell'articolo sia univoco...

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Name LIKE 'Sport-100 Helmet, Blue'
```



	ProductID	Name	ListPrice	Color
1	711	Sport-100 Helmet, Blue	34,99	Blue

L'operatore **LIKE** consente di ricercare la corrispondenza dei valori di una colonna a un pattern specificato (stringa). Il risultato espone l'unico articolo il cui nome ha una corrispondenza esatta con il pattern indicato.

Ma se non conoscessimo o non avessimo evidenza della stringa completa? Se conoscessimo parte della stringa da ricercare? Come facciamo a cercare tutti i prodotti il cui nome inizia con *Sport-100 Helmet* per esempio?

SQL: utilizzo dell'operatore LIKE

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario recuperare tutti i codici prodotto il cui nome comincia con *Sport-100 Helmet* a prescindere dal resto dei caratteri.

Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Name LIKE 'Sport-100 Helmet%'
```



	ProductID	Name	ListPrice	Color
1	707	Sport-100 Helmet, Red	34,99	Red
2	708	Sport-100 Helmet, Black	34,99	Black
3	711	Sport-100 Helmet, Blue	34,99	Blue

L'operatore LIKE può essere utilizzato in combinazione al carattere jolly *.

Il carattere jolly (o wildcard) * consente di cercare tutti valori che corrispondono al pattern indicato seguito da qualsiasi stringa (a prescindere da eventuali altri caratteri che lo seguono..)

Nell'esempio, il risultato espone tutti i prodotti il cui nome comincia per *Sport-100 Helmet* a prescindere da eventuali altri caratteri che lo seguono a completamento, in questo caso, del nome prodotto.

SQL: utilizzo dell'operatore LIKE

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario recuperare tutti i codici prodotto il cui nome termine con il carattere *M* (a prescindere da tutto ciò che c'è prima...)
Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Name LIKE '%M'
```



	ProductID	Name	ListPrice	Color
1	709	Mountain Bike Socks, M	9,50	White
2	714	Long-Sleeve Logo Jersey, M	49,99	Multi

L'operatore LIKE può essere utilizzato in combinazione al caratteri jolly *.

Il carattere jolly (o wildcard) * consente di cercare tutti valori che corrispondono al pattern indicato preceduto da qualsiasi stringa (a prescindere da eventuali altri caratteri che lo precedono...)

Nell'esempio, il risultato espone tutti i prodotti il cui termina con il carattere *M*, a prescindere da eventuali altri caratteri che lo precedono a completamento, in questo caso, del nome prodotto!

SQL: utilizzo dell'operatore LIKE

Supponiamo di dover interrogare la tabella **Product** (che ha le colonne e le righe indicate nell'immagine sottostante). È necessario recuperare tutti i codici prodotto il cui nome contiene *Logo*. Il result set deve contenere i campi: ProductID, Name, ListPrice, Color.

ProductID	Name	ListPrice	StandardCost	Size	Color	SellEndDate
680	HL Road Frame - Black, 58	1431,5	1059,31	58	Black	NULL
706	HL Road Frame - Red, 58	1431,5	1059,31	58	Red	NULL
707	Sport-100 Helmet, Red	34,99	13,0863	NULL	Red	NULL
708	Sport-100 Helmet, Black	34,99	13,0863	NULL	Black	NULL
709	Mountain Bike Socks, M	9,5	3,3963	M	White	30/06/2023
710	Mountain Bike Socks, L	9,5	3,3963	L	White	30/06/2023
711	Sport-100 Helmet, Blue	34,99	13,0863	NULL	Blue	NULL
712	AWC Logo Cap	8,99	6,9223	NULL	Multi	NULL
713	Long-Sleeve Logo Jersey, S	49,99	38,4923	S	Multi	NULL
714	Long-Sleeve Logo Jersey, M	49,99	38,4923	M	Multi	NULL

```
SELECT ProductID, Name, ListPrice, Color
FROM Product
WHERE Name LIKE '%Logo%'
```



	ProductID	Name	ListPrice	Color
1	712	AWC Logo Cap	8,99	Multi
2	713	Long-Sleeve Logo Jersey, S	49,99	Multi
3	714	Long-Sleeve Logo Jersey, M	49,99	Multi

L'operatore LIKE può essere utilizzato in combinazione al carattere jolly *.

Il carattere jolly (o wildcard) * consente di cercare tutti i valori che contengono (in qualsiasi posizione) il pattern *Logo*. Il carattere jolly * indicato all'inizio ed alla fine del pattern consente di ignorare tutti i caratteri che lo precedono o lo seguono.

SQL
LET'S
TAKE A
LOOK!



GRAZIE
Epicode