

# Verifica su Query SQL

## Contesto di riferimento

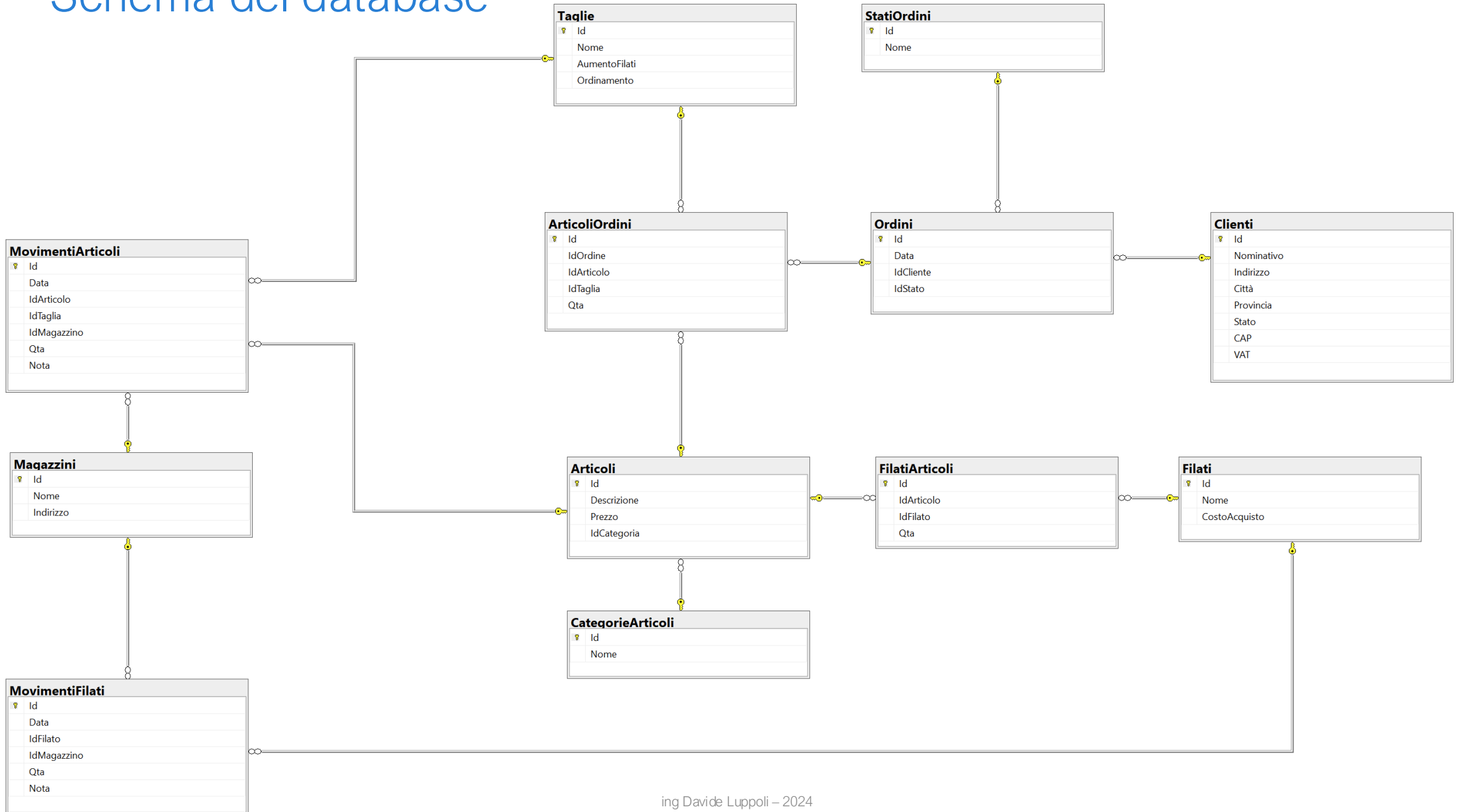
Una azienda di moda che produce vestiti utilizza un semplice database Microsoft Sql per memorizzare i dati relativi agli ordini, agli articoli ordinati e alle giacenze di magazzino. Il database contiene anche alcune informazioni utili per la produzione dei vestiti, come la distinta base dei filati necessari.

Ogni vestito è disponibile in più taglie (dalla XS alla XL) mentre la disponibilità in molteplici colori è gestita tramite una codifica differente.

Vi è richiesta la realizzazione di una serie di query che operano sul database con l'obiettivo di rispondere ad alcune necessità aziendali.

Il database vi è stato passato via github ed è composto da 12 tabelle, collegate tra di loro con le relazioni riportate nello schema che segue:

# Schema del database



# Schema tabelle

```
CREATE TABLE Filati(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Nome VARCHAR(200) NOT NULL,  
    CostoAcquisto FLOAT  
)  
  
CREATE TABLE Taglie(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Nome VARCHAR(10) NOT NULL,  
    AumentoFilati FLOAT NOT NULL DEFAULT 0,  
    Ordinamento INT NOT NULL  
)
```

# Schema tabelle

```
CREATE TABLE CategorieArticoli(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Nome VARCHAR(200) NOT NULL  
)
```

```
CREATE TABLE StatiOrdini(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Nome VARCHAR(200) NOT NULL  
)
```

# Schema tabelle

```
CREATE TABLE Clienti(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Nominativo VARCHAR(1024) NOT NULL,  
    Indirizzo VARCHAR(1024),  
    Città VARCHAR(1024),  
    Provincia VARCHAR(200),  
    Stato VARCHAR(200),  
    CAP VARCHAR(50),  
    VAT VARCHAR(50)  
)
```

# Schema tabelle

```
CREATE TABLE Articoli (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Descrizione VARCHAR(1024) NOT NULL,  
    Prezzo FLOAT CHECK(Prezzo > 0),  
    IdCategoria INT NOT NULL,  
    FOREIGN KEY(IdCategoria) REFERENCES CategorieArticoli(Id)  
)
```

# Schema tabelle

```
CREATE TABLE Ordini(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Data Date NOT NULL,  
    IdCliente INT NOT NULL,  
    IdStato INT NOT NULL,  
    FOREIGN KEY(IdCliente) REFERENCES Clienti(Id),  
    FOREIGN KEY(IdStato) REFERENCES StatiOrdini(Id)  
)
```



# Schema tabelle

```
CREATE TABLE FilatiArticoli(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    IdArticolo INT NOT NULL,  
    IdFilato INT NOT NULL,  
    Qta FLOAT CHECK(Qta > 0),  
    FOREIGN KEY(IdArticolo) REFERENCES Articoli(Id),  
    FOREIGN KEY(IdFilato) REFERENCES Filati(Id)  
)
```

# Schema tabelle

```
CREATE TABLE ArticoliOrdini(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    IdOrdine INT NOT NULL,  
    IdArticolo INT NOT NULL,  
    IdTaglia INT NOT NULL,  
    Qta FLOAT CHECK(Qta > 0),  
    FOREIGN KEY(IdOrdine) REFERENCES Ordini(Id),  
    FOREIGN KEY(IdArticolo) REFERENCES Articoli(Id),  
    FOREIGN KEY(IdTaglia) REFERENCES Taglie(Id)  
)
```

# Schema tabelle

```
CREATE TABLE Magazzini(  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Nome VARCHAR(100) NOT NULL,  
    Indirizzo VARCHAR(1000) NOT NULL  
)
```

# Schema tabelle

```
CREATE TABLE MovimentiArticoli (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Data DATE NOT NULL,  
    IdArticolo INT NOT NULL,  
    IdTaglia INT NOT NULL,  
    IdMagazzino INT NOT NULL,  
    Qta FLOAT NOT NULL,  
    Nota VARCHAR(1024),  
    FOREIGN KEY (IdArticolo) REFERENCES Articoli(Id),  
    FOREIGN KEY (IdTaglia) REFERENCES Taglie(Id),  
    FOREIGN KEY (IdMagazzino) REFERENCES Magazzini(Id)  
)
```

# Schema tabelle

```
CREATE TABLE MovimentiFilati (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Data DATE NOT NULL,  
    IdFilato INT NOT NULL,  
    IdMagazzino INT NOT NULL,  
    Qta FLOAT NOT NULL,  
    Nota VARCHAR(1024),  
    FOREIGN KEY (IdFilato) REFERENCES Filati(Id),  
    FOREIGN KEY (IdMagazzino) REFERENCES Magazzini(Id)  
)
```

# Esercizi da risolvere

## ESERCIZIO 1

Calcolare quanti ordini in stato "Confermato" esistono nel database

OrdiniConfermati
3

## ESERCIZIO 2

Recuperare l'elenco di tutti gli articoli completo di categoria, ordinati per codice articolo

Id	Descrizione	Prezzo	Categoria
1	Pantalone elegante nero	100	Pantalone
2	Pantalone elegante bianco	100	Pantalone
...	...	...	...

(14 righe in totale)

# Esercizi da risolvere

## ESERCIZIO 3

Recuperare l'elenco di tutti gli ordini completo di nome del cliente e stato dell'ordine.  
Ordinare per stato dell'ordine (alfabetico) e per nominativo cliente.

Id	Data	Nominativo	Stato
8	2022-01-15	Italienische Kleidung	Confermato
7	2022-01-15	La boutique del vestito	Confermato
...	...	...	...

(9 righe in totale)

# Esercizi da risolvere

## ESERCIZIO 4

Calcolare la quantità non ancora prodotta di ogni articolo. Considerare sia gli ordini confermati che in produzione. Ordinare per quantità da produrre decrescente e per codice articolo

Id	Descrizione	Qta
2	Pantalone elegante bianco	111
5	Polo a manica lunga bianca	101
8	Polo a manica corta bianca	101
..	..	..

(14 righe in totale)



# Esercizi da risolvere

## ESERCIZIO 5

Recuperare l'elenco di tutti gli articoli non ancora completamente prodotti (considerare sia gli ordini confermati che in produzione), riportando i numeri d'ordine e calcolandone le quantità. Ordinare per codice articolo.

Id	Descrizione	Ordine	Qta
1	Pantalone elegante nero	5	1
1	Pantalone elegante nero	8	10
2	Pantalone elegante bianco	5	1
..	..		..

(22 righe in totale)

# Esercizi da risolvere

## ESERCIZIO 6

Recuperare il fatturato generato da ogni cliente, considerando tutti gli ordini, anche quelli non ancora prodotti. Ordinare per fatturato decrescente.

Inserire anche i clienti che non hanno generato fatturato, riportando NULL nella colonna relativa.

Id	Nominativo	Fatturato
6	Italian firms	42500
3	La boutique del vestito	29950
..	..	...

(8 righe in totale)

# Esercizi da risolvere

## ESERCIZIO 7

Calcolare il fabbisogno di filati per produrre l'ordine numero 6, ordinando per quantità decrescente di filato necessario.

Id	Filato	Qta
3	Filato in cotone Bianco	175
15	Filato misto Bianco	40
16	Filato misto Rosso	2

(3 righe in totale)

# Esercizi da risolvere

## ESERCIZIO 8

Recuperare il dettaglio delle taglie da produrre (considerare sia gli ordini confermati che in produzione), riportando tutte le taglie in orizzontale. Ordinare per codice articolo.

Id	Descrizione	XS	S	M	L	XL
1	Pantalone elegante nero	2	2	3	2	2
2	Pantalone elegante bianco	22	22	23	22	22
3	Pantalone sportivo grigio	7	7	8	7	7
...	...	...	...	...	...	...

(14 righe in totale)

# Esercizi da risolvere

## ESERCIZIO 9

Creare due viste per presentare le giacenze di magazzino, con i seguenti schemi

- MagazzinoArticoli(IdArticolo, IdTaglia, IdMagazzino, Qta)
- MagazzinoFilati(IdFilato, IdMagazzino, Qta)

# Esercizi da risolvere

## ESERCIZIO 10

Creare una vista per rappresentare i filati da riordinare, ovvero i filati per i quali Giacenza – Necessaria < 10.

Restituire il seguente schema:

- FilatiDaOrdinare(IdFilato, Giacenza, Necessaria)

## Note pratiche

- Se viene richiesto un ordinamento, inserirlo esplicitamente nella query. Anche se i dati sembrano essere ordinati coerentemente anche in assenza di ORDER BY
- Attenzione al numero e ai nomi delle colonne. I risultati devono essere coerenti con quanto richiesto.
- Non ci sono UNION e le sottoquery, se servono, riguardano solo l'esercizio 8
- Consegnare un file per ogni esercizio, con le seguenti caratteristiche:
  - Il file si deve chiamare EsercizioN.sql, dove N è il numero dell'esercizio che si sta risolvendo
  - In ogni file deve essere presente l'istruzione USE
- Archiviare i file in un repository privato github fornendo diritti di visualizzazione al docente (account @dluppoli)