```
INSERT INTO Studente (ID, Cognome, DataDiNascita, AltezzaInMetri, Telefono)
VALUES (0, 'Giodice', '2005-01-27', 1.78, '3207589090')

SELECT * FROM Studente

UPDATE Studente
SET PesoInKg = 60
WHERE ID = 0

DELETE FROM Studente WHERE Cognome = 'Giodice';

ALTER TABLE Studente
ALTER COLUMN Telefono varchar(255)

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

INSERT INTO Telefono(Numero, IDS)
VALUES('3383456444', 4)

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

SELECT * FROM Studente

CREATE TABLE Voto (
	ID int IDENTITY(0,1) PRIMARY KEY,
	Cognomes varchar(50) FOREIGN KEY REFERENCES Studente (Cognome),
	Cognomep varchar(50) FOREIGN KEY REFERENCES Professore (Cognome),
	Materia varchar(50) FOREIGN KEY REFERENCES Professore (Materia),
	Tipo varchar(50)
);



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

```
CREATE TABLE Voto (
        ID int IDENTITY(0,1) PRIMARY KEY,
        Cognomes varchar(50) FOREIGN KEY REFERENCES Studente (Cognome),
        Cognomep varchar(50)FOREIGN KEY REFERENCES Professore (Cognome),
        Valore int,
        Giorno date,
        Tipo varchar(50)
)


ALTER TABLE dipendente
ADD dataassunzione date;
```

https://www.sqlservertutorial.net/sql-server-sample-database/

Esercizi:
Si vuole creare un db per inserire i voti che gli studenti prendono nelle varie materie.
Ogni studente si deve memorizzare nome cognome classe, delle materie siamo interessati al nome e al professore che la insegna(nome e cognome).
Per il voto siamo interessati al suo valore, alla data in cui è stato ricevuto e al tipo(scritto orale o pratico)
una volta realizzato il db, aggiungere il campo anno scolastico al voto, inserire due studenti, due materie e 4 voti, due per ogni materia
scrivere le istruzioni per aggiungere un'unità a tutti i voti.
scrivere infine le istruzioni per eliminare tutti gli studenti

Risultati | Messaggi

	ID	Cognomes	Cognomep	Valore	Giorno	Tipo
1	0	Giodice	Apicella	7	2022-09-20	Orale
2	1	Giodice	Apicella	6	2022-08-24	Scritto
3	2	Ahmed	Apicella	8	2022-09-20	Orale
4	3	Ahmed	Apicella	7	2022-09-13	Scritto
5	4	Giodice	Titze	6	2022-08-16	Scritto
6	5	Giodice	Titze	5	2022-09-06	Scritto
7	6	Ahmed	Titze	7	2022-08-18	Orale
8	7	Ahmed	Titze	8	2022-09-16	Scritto

Quando si vuole scrivere o si deve scrivere una query
1. Individuare Dove sono le informazioni, se in più tabelle devo fare join individuando i campi che hanno in relazione le due tabelle.
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)
4.

—----------

Grouby

```
SELECT model_year, COUNT(*) FROM production.products
GROUP BY model_year

SELECT model_year, COUNT(*) AS totale_prodotti FROM production.products
GROUP BY model_year
ORDER BY totale_prodotti

SELECT model_year, COUNT(*) AS totale_prodotti FROM production.products
GROUP BY model_year
ORDER BY totale_prodotti desc

SELECT model_year, COUNT(*) AS totale_prodotti FROM production.products
GROUP BY model_year
HAVING COUNT(*) > 50
ORDER BY totale_prodotti desc

SELECT state,city, COUNT(*) AS clienti FROM sales.customers
GROUP BY state,city
ORDER BY clienti desc

SELECT state,city, COUNT(*) AS clienti FROM sales.customers
GROUP BY state,city
HAVING COUNT(*) > 10
ORDER BY clienti desc
```

somme del prezzo 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
```

```
--prezzo del prodotto con il prezzo piu alto
SELECT MAX(list_price) AS prezzo_max FROM production.products

SELECT production.products.product_name
FROM production.products
WHERE list_price = 11999.99

SELECT production.products.product_name
FROM production.products
WHERE list_price = (SELECT MAX(list_price) FROM production.products)



SELECT production.brands.brand_id, brand_name, max(list_price) AS somma_totale
FROM production.brands inner join production.products
ON production.brands.brand_id = production.products.brand_id
GROUP BY production.brands.brand_id, brand_name

CORREZIONE PREPARAZIONE VERIFICA

Ricetta(id*,nome,tempo,diff,proced)
Ingrediente(id*,nome)
Rivista(id*,numero,anno)
Pubblicazione(id_ric**, id_riv**,pagina
Preparazione(id_ric**, id_ingr**,quantita, unita_misura


CREATE TABLE ricetta(
        id int PRIMARY KEY IDENTITY(0,1),
        nome varchar(255),
        tempo_c int,
        tempo_p int,
        diff varchar(10),
        proced varchar(255)
)


ALTER TABLE preparazione
ADD id_ric int foreign key references ricetta(id)

INSERT INTO ricetta(nome,tempo_c,tempo_p,diff,proced)
VALUES('nome_ricetta',10,20,'facile','cucinare..')


UPDATE preparazione
SET quantita = 100, unita_misura = 'gr'
WHERE id_ingr = (select id from ingrediente where nome = 'Radicchio')
AND id_ric = (select id from ricetta where nome = 'nome_ricetta')
```