1. Un database è un insieme di dati che vengono organizzati in modo coerente e logico, così da renderne più semplice la lettura, scrittura, modifica o l’eliminazione dei dati.
2. Un DBMS (Database Management System) è un software che ha lo scopo di creare, manipolare e gestire i database.
3. 5🡪SELECT: è l’istruzione che viene scritta all’inizio dello statement ma in realtà viene elaborata dopo altre 4 istruzioni. Nel SELECT vengono indicate le colonne che vogliamo includere nel result set della query.  
   1🡪FROM: qui si specifica quale tabella interrogare per recuperare i dati.  
   2🡪WHERE: è l’istruzione che funge da filtro, il result set sarà “influenzato” dalle condizioni imposte nel WHERE.  
   3🡪GROUP BY: questa clausola consente di raggruppare righe in base a quelle che indichiamo.  
   4🡪HAVING: simile al WHERE, anche questa clausola funge da filtro, in particolare per i raggruppamenti.   
   6🡪ORDER BY: infine, questa clausola ha lo scopo di ordinare (ASC o DESC) il result set in base alle colonne che specifichiamo.
4. Negozio di elettronica.  
   Immagine che contiene testo, schermata, Carattere, numero

   Descrizione generata automaticamente  
     
   Il result set sarà:  
   Immagine che contiene testo, Carattere, schermata, numero

   Descrizione generata automaticamente

Attraverso la clausola GROUP BY è stato possibile raggruppare le righe in base alla categoria del prodotto, così da ottenere il ricavato per ciascuna categoria.

SELECT Categoria, SUM(Prezzo) AS Totale Ricavato  
FROM Prodotti  
GROUP BY Categoria

1. OLTP (Online Transaction Processing): è un software che ha lo scopo di gestire transazioni e interazioni con il database. Ad esempio, si utilizza per sistemi di prenotazione.  
   OLAP (Online Analytical Processing): è un software necessario per l’analisi di grandi volumi di dati e l’interrogazione degli stessi. Ad esempio, viene utilizzato nei data warehouse per analisi di vendite nelle aziende.
2. La Join un’operazione utilizzata per ottenere una combinazione di tabelle in base a una determinata condizione. Dall’altra parte, la subquery viene utilizzata per filtrare i risultati di una tabella, dando come input della query esterna il risultato della subquery stessa.  
   In un medesimo scenario di analisi, è possibile ottenere lo stesso risultato con entrambe le operazioni; quindi, se è necessario combinare tra loro due tabelle, si utilizza la join, mentre se si vuole applicare filtrare i risultati di una query allora si utilizza la subquery.
3. DML (Data Manipulation Language): linguaggio utilizzato per la manipolazione dei dati di un database attraverso le istruzioni che lo caratterizzano, ovvero INSERT, UPDATE, DELETE  
   DDL (Data Definition Language): linguaggio utilizzato per definire la struttura di un database; in questo caso, le istruzioni incluse sono CREATE, ALTER E DROP.

SELECT DATEPART(YEAR, OrderDate) AS Anno

FROM Sales

SELECT YEAR(DataOrdine) AS Anno

FROM Sales

1. La differenza tra questi due operatori è che con AND viene restituito TRUE se entrambe le condizioni sono vere, mentre con OR almeno una delle due deve essere vera.
2. Sì, è possibile, in questo caso si parla di query innestata o subquery e serve per filtrare i risultati di una tabella
3. Gli operatori logici OR e IN vengono utilizzati in situazioni diverse; il primo si utilizza per combinare delle condizioni nella clausola WHERE, se è vera anche solo una di queste allora verrà restituito TRUE. IN invece, si utilizza per fare un confronto tra un valore e un elenco di valori indicati, restituendo così il valore che corrisponde ad uno di quelli specificati in questo elenco.
4. Sì, include anche gli estremi
5. Con SELF JOIN si intende un particolare tipo di JOIN in cui una tabella viene combinata con sé stessa e si utilizza quando è necessario confrontare record della tabella stessa.  
   Nel caso in cui avessimo una tabella degli ordini di diversi clienti e volessimo confrontare gli ordini di uno stesso cliente in date diverse, allora la SELF JOIN si potrebbe fare così:  
     
   TABELLA ORDINI  
   Immagine che contiene testo, schermata, Carattere, numero

   Descrizione generata automaticamente  
     
     
     
     
     
     
     
     
     
   RESULT SET  
   Immagine che contiene testo, Carattere, schermata, linea

   Descrizione generata automaticamente

SELECT o1.ID\_Ordine AS ID\_Ordine1,

o1.ID\_Cliente,

o1.DataOrdine AS DataOrdine1,

o2.ID\_Ordine AS ID\_Ordine2,

o2.DataOrdine AS DataOrdine2

FROM Ordini AS o1

INNER JOIN Ordini AS o2

ON o1.ID\_Cliente = o2.ID\_Cliente

WHERE o1.ID\_Ordine <> o2.ID\_Ordine;

**TASK 1**: PROGETTAZIONE CONCETTUALE E LOGICA DELLA BASE DATI

Immagine che contiene diagramma, linea, testo, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

**TASK 2**:

-- TASK 2: Implementazione tabella Product

CREATE TABLE Product (

ProductID INT,

ProductName NVARCHAR(50),

ProductCategory NVARCHAR(50),  
Sold BIT

CONSTRAINT PK\_ProductID PRIMARY KEY (ProductID)

);

-- TASK 2: Implementazione tabella Region

CREATE TABLE Region (

RegionID INT,

RegionName NVARCHAR(50),

CountryName NVARCHAR(50)

CONSTRAINT PK\_RegionID PRIMARY KEY (RegionID)

);

-- TASK 2: Implementazione tabella Sales

CREATE TABLE Sales (

SalesID INT,

ProductID INT,

RegionID INT,

OrderDate DATE,

OrderQuantity INT,

UnitPrice DECIMAL(10, 2),

SalesAmount DECIMAL(10, 2),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID),

FOREIGN KEY (RegionID) REFERENCES Region(RegionID)

);

**TASK 3**:

-- TASK 3: Popolamento tabella Product

INSERT INTO Product (ProductID, ProductName, ProductCategory, Sold)

VALUES

(1, 'Puzzle', 'Toys', 1),

(2, 'Skates', 'Sport', 1),

(3, 'Bicycle', 'Sport', 1),

(4, 'Lego', 'Toys', 0),

(5, 'Soccer ball', 'Sport', 0);

-- TASK 3: Popolamento tabella Region

INSERT INTO Region (RegionID, RegionName, CountryName)

VALUES

(1, 'SouthEurope', 'Italy'),

(2, 'WestEurope', 'Germany'),

(3, 'NorthAmerica', 'United States'),

(4, 'SouthAmerica', 'Brazil'),

(5, 'Asia', 'China');

-- TASK 3: Popolamento tabella Sales

INSERT INTO Sales (SalesID, ProductID, RegionID, OrderDate, OrderQuantity, UnitPrice, SalesAmount)

VALUES

(1, 1, 1, '2018-04-22', 2, 10.00, 20.00),

(2, 2, 1, '2019-01-05', 1, 75.00, 75.00),

(3, 3, 5, '2019-06-07', 1, 100.00, 100.00),

(4, 2, 4, '2020-02-17', 3, 75.00, 225.00),

(5, 3, 4, '2020-11-08', 5, 100.00, 500.00),

(6, 4, NULL, NULL, NULL, 15.00, NULL),

(7, 1, 1, '2021-05-19', 1, 10.00, 10.00),

(8, 5, NULL, NULL, NULL, 20.00, NULL),

(9, 1, 2, '2021-09-07', 3, 10.00, 30.00),

(10, 2, 3, '2021-12-02', 4, 75.00, 300.00);

**TASK 4: SVOLGIMENTO NEL FILE “W12D4\_Consegna.sql”**