**Risposta Domanda 1**:

Il database è un contenitore di dati (o repository) che sono organizzati in maniera consistente

**Risposta Domanda 2**:

Il database management system è un software che permette di recuperare e organizzare i dati.

**Risposta domanda 3:**

5)SELECT – indica campi, costanti o espressioni del risultato. Permette di organizzare il Result SET (che non è una tabella).

1. From – indica da dove prendere i dati (la sorgente dei dati)
2. Where- indica il filtro da applicare ai dati
3. Group by – permette dei raggruppamenti simil PIVOT
4. Having – è la condizione (o filtro) che inserisco al Group by

6) Order by – permette di scegliere ordine di visualizzazione.

**Risposta domanda 4:**

Es.

|  |  |
| --- | --- |
| **TOPICPOST** | **Commenti** |
| MSCMeraviglia | 10 |
| MSCMeraviglia | 3 |
| MSCMeraviglia | 1 |
| MSCMediterraneo | 8 |

Query:

**SELECT** Topicpost, SUM (Commenti)

**FROM** MSCnavi

**GROUP BY** Topicpost

Il risultato è:

|  |  |
| --- | --- |
| **TOPICPOST** | **Commenti** |
| MSCMediterraneo | 8 |
| MSCMeraviglia | 14 |

Se volessi aggiungere Having

Query:

**SELECT** Topicpost, SUM (Commenti)

**FROM** MSCnavi

**GROUP BY** Topicpost

**HAVING** Topicpost **LIKE** ‘MSCMERAVIGLIA’

|  |  |
| --- | --- |
| **TOPICPOST** | **Commenti** |
| MSCMeraviglia | 14 |

**Risposta domanda 5:**

**OLTP** – è una base dati per ottimizzare le operazioni di CRUD. Garantisce la gestione dei dati e implementa dei vincoli. È una base normalizzata. La normalizzazione è la distribuzione dei dati di una entità su più tabelle. Nell’OLTP non c’è ridondanza dei dati.

**OLAP –** è una base dati che permette attraverso la denormalizzazione, di prendere i dati e inserirli in un'unica tabella. La ridondanza che si crea è necessaria per svolgere le analisi. Un esempio di OLAP è LA DATA WAREHOUSE.

OLTP e OLAP sono nello stesso cloud.

**Risposta domanda 6:**

La differenza tra una Inner Join e una query innestata, è che le colonne della query interna non compaiono.

**Risposta domanda 7:**

**DDL: Data definition language – si crea qualcosa (tabelle, database..)**

CREATE – Crea

Alter – modifica

Drop- elimina

**DML: Data manipulation language**, permettono di manipolare i dati.

SELECT- seleziona

INSERT- inserisci

UPDATE- aggiorna

DELETE- cancella

**Risposta 8**

Select YEAR (‘2024-01-29’) AS ANNO

Select YEAR (nomecolonnacondata) AS ANNO

**Risposta 9:**

L’operatore AND si verifica nel momento in cui le condizioni sono vere, OR se una delle condizioni è vera.

**Risposta 10:**

Le query innestate possono essere messe in qualunque livello della query. Anche nella SELECT.

**Risposta 11:**

L’operatore OR si verifica se una delle condizioni è vera. L’operatore IN si verifica nel momento in cui c’è corrispondenza con specifici valori.

La tabella finale è uguale, ma cambia la lunghezza della stringa e a livello concettuale.

**Risposta 12:**

L’operatore logico BETWEEN comprende anche gli estremi del range

**Risposta 13:**

La Self Join è una join su una stessa tabella.

Solitamente nella tabella in analisi ci sono dei legami (gerachici, di categoria…)

|  |  |  |
| --- | --- | --- |
| Codiceproduzione | FOOD | Categoria di appartenenza |
| 1 | Formaggi | NULL |
| 2 | mozzarelle | 1 |
| 3 | Ricotta | 1 |
| 4 | salumi | NULL |
| 5 | Mortadella | 4 |
| 6 | Crudo | 4 |
| 7 | salame | 4 |

Nella colonna codice di produzione ho un numero legato sia la macro che alla micro categoria, nella colonna food ho sia la macrocategia (Formaggi e Salumi) che la microcategoria (mozzarelle,ricotta…), nella categoria di appartenenza ho per ogni microcategoria il codice di corrispondenza della macro.

Voglio trasformare il dato in maniera tale che venga risaltato il rapporto tra macro e micro categoria.

Mi si creano 2 tabelle immaginarie con la inner join sulla medesima tabella.

tabella Cateogira

|  |  |  |
| --- | --- | --- |
| Codiceproduzione | FOOD | Categoria di appartenenza |
| 1 | Formaggi | NULL |
| 2 | mozzarelle | 1 |
| 3 | Ricotta | 1 |
| 4 | salumi | NULL |
| 5 | Mortadella | 4 |
| 6 | Crudo | 4 |
| 7 | salame | 4 |

Tabella Sottocategoria

|  |  |  |
| --- | --- | --- |
| Codiceproduzione | FOOD | Categoria di appartenenza |
| 1 | Formaggi | NULL |
| 2 | mozzarelle | 1 |
| 3 | Ricotta | 1 |
| 4 | salumi | NULL |
| 5 | Mortadella | 4 |
| 6 | Crudo | 4 |
| 7 | salame | 4 |

**SELECT** Categia.CODICEPRODUZIONE **AS** CODICEDISETTORE,Categoria.FOOD **AS** SETTORE,Sottocategoria.CODICEPRODUZIONE **AS** NUMEROINDETTAGLIO, Sottocategoria.FOOD **AS** NOMEDETTAGLIO

**FROM** CIBO **AS** Categoria

**INNER JOIN** CIBO **AS** Sottocategoria

**ON** Categoria.CODICEPRODUZIONE**=** Sottocategoria.CATEGORIAdiappartenza

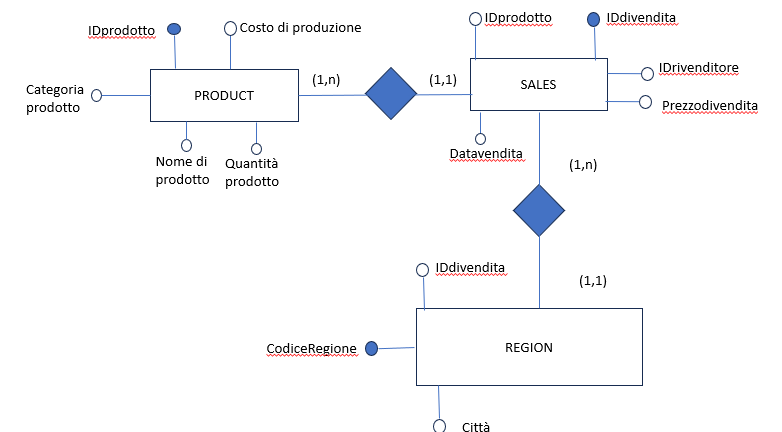
Descrivendo e rapportando le 2 colonne delle due tabelle. Codice produzione dalla tabella 1 e categoria dalla tabella 2. I numeri che io trovo in corrispondenza sia del codice di produzione che della categoria di appartenenza sono l’1 e il 4.

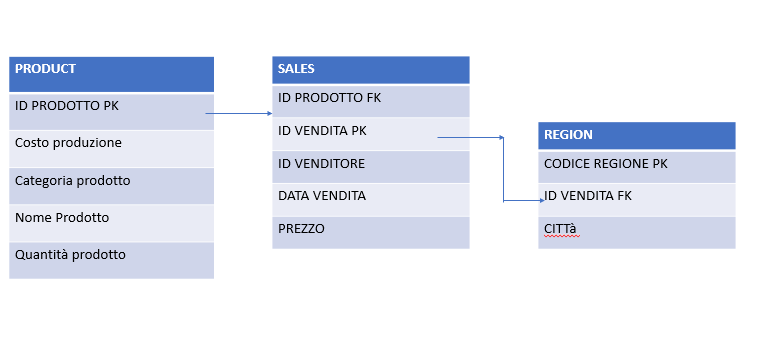
**Sono questi i numeri di intersezione.**

Dopo di che ho bisogno di dirgli cosa vedere nel result set.

|  |  |  |  |
| --- | --- | --- | --- |
| Codicedisettore | SETTORE | NUMERO IN DETTAGLIO | NOME DETTAGLIO |
| 1 | Formaggi | 2 | Mozzarelle |
| 1 | Formaggi | 3 | ricotta |
| 4 | salumi | 5 | mortadella |
| 4 | salumi | 6 | Crudo |
| 4 | salumi | 7 | salame |

**Task 1: progettazione concettuale e logica**





**Task2:**

CREATE TABLE PRODOTTI

(IDPRODOTTO INT,

COSTOPRODUZIONE INT,

CATEGORIAPRODOTTO CHAR (20),

NOMEPRODOTTO CHAR (20),

STATOAVANZAMENTO INT)

CREATE TABLE VENDITE

(IDPRODOTTO INT,

IDVENDITA INT,

PREZZO INT

Datavendita DATE)

CREATE TABLE REGION

(CODICEREGIONE INT,

IDVENDITA INT,

CITTà CHAR(20)

)

**Task3:**

CREATE TABLE PRODOTTI

(IDPRODOTTO INT,

COSTOPRODUZIONE INT,

CATEGORIAPRODOTTO CHAR (20),

NOMEPRODOTTO CHAR (20),

quantitòprodotto INT)

INSERT INTO PRODOTTI

VALUES (1234,20,'bicicletta','graziella',3),

(1235,23,'bicicletta','graziellaf',1),

(1236,40,'vestiti', 'guanti',4),

(1237,35,'vestiti', 'casco',6)

CREATE TABLE VENDITE

(IDPRODOTTO INT,

IDVENDITA INT,

PREZZO INT,

Datavendita DATE)

INSERT INTO VENDITE

VALUES (1234,0001, 40, '2023-04-15'),

(1235,0002, 45,'2023-05-22'),

(1236,0003,55,'2023-07-12'),

(1237,0004,60,'2023-08-23')

create table REGION

(CODICEREGIONE INT,

IDVENDITA INT,

CITTà CHAR(20))

INSERT INTO REGION

values

(40406,0001,'NewYork'),

(40408,0002,'Chicago'),

(40409,0003, 'Seattle'),

(40410,0004,'Lasvegas')

Per definire chiavi interne ed esterne su SQL utilizzo:

CREATE TABLE PRODOTTI

(IDPRODOTTO INT PRIMARY KEY,

COSTOPRODUZIONE INT,

CATEGORIAPRODOTTO CHAR (20),

NOMEPRODOTTO CHAR (20),

quantitòprodotto INT)

INSERT INTO PRODOTTI

VALUES (1234,20,'bicicletta','graziella',3),

(1235,23,'bicicletta','graziellaf',1),

(1236,40,'vestiti', 'guanti',4),

(1237,35,'vestiti', 'casco',6)

CREATE TABLE VENDITE

(IDPRODOTTO INT,

IDVENDITA INT PRIMARY KEY,

PREZZO INT,

Datavendita DATE,

FOREIGN KEY (IDPRODOTTO) REFERENCES PRODOTTI (IDPRODOTTO))

INSERT INTO VENDITE

VALUES (1234,0001, 40, '2023-04-15'),

(1235,0002, 45,'2023-05-22'),

(1236,0003,55,'2023-07-12'),

(1237,0004,60,'2023-08-23')

create table REGION

(CODICEREGIONE INT PRIMARY KEY,

IDVENDITA INT,

CITTà CHAR(20)

FOREIGN KEY (IDVENDITA) REFERENCES VENDITE (IDVENDITA))

INSERT INTO REGION

values

(40406,0001,'NewYork'),

(40408,0002,'Chicago'),

(40409,0003, 'Seattle'),

(40410,0004,'Lasvegas')

**FOGLIO 4**

1)

SELECT IDPRODOTTO, COUNT(IDPRODOTTO) as conteggio\_duplicati

FROM PRODOTTI

GROUP BY IDPRODOTTO

**2)**

SELECT B.IDVENDITA,B.Datavendita,A.NOMEPRODOTTO,A.CATEGORIAPRODOTTO,C.CITTà,C.CODICEREGIONE,

case when B.Datavendita< GETDATE()-180 THEN 'True' ELSE 'False' END AS Vendite\_sopra\_180\_fgiorni

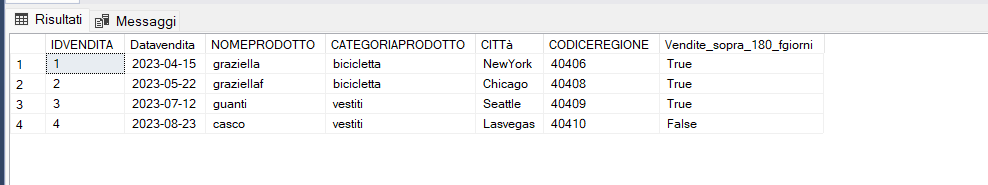
FROM PRODOTTI AS A

INNER JOIN VENDITE AS B

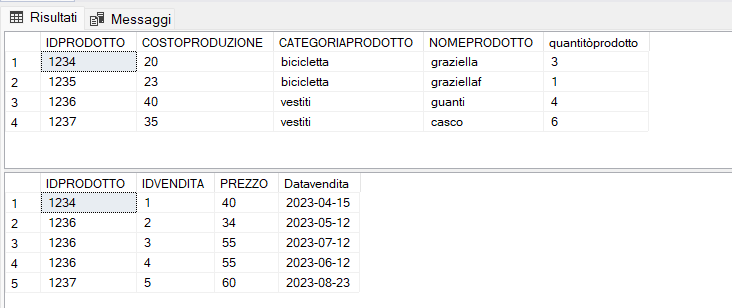
ON A.IDPRODOTTO=B.IDPRODOTTO

INNER JOIN REGION AS C

ON B.IDVENDITA=C.IDVENDITA

****

**3)** **Per fare esercizio 3 ho dovuto modificare le tabelle PRODOTTI E VENDITE e aggiungere il non venduto e più vendite per un prodotto.**

****

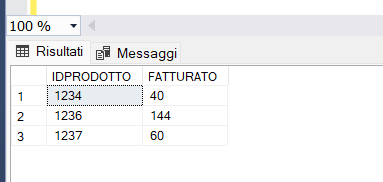
SELECT A.IDPRODOTTO, SUM(PREZZO) AS FATTURATO

FROM PRODOTTI AS A

INNER JOIN VENDITE AS B

ON A.IDPRODOTTO=B.IDPRODOTTO

GROUP BY A.IDPRODOTTO

****

**4)**

SELECT

R.CODICEREGIONE,

YEAR(V.Datavendita) AS Anno,

SUM(V.PREZZO) AS Fatturato

FROM

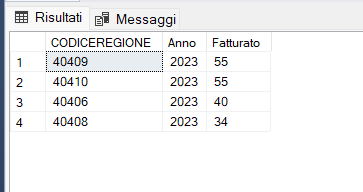
VENDITE as V

JOIN REGION as r ON V.IDVENDITA = R.IDVENDITA

GROUP BY

R.CODICEREGIONE, YEAR(V.Datavendita)

ORDER BY Anno DESC, Fatturato DESC

****

1. **È LA CATEGORIA VESTITI.**

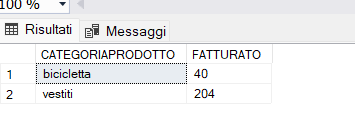
SELECT A.CATEGORIAPRODOTTO, SUM (B.PREZZO) AS FATTURATO

FROM PRODotti AS A

INNER JOIN VENDITE AS B

ON A.IDPRODOTTO=B.IDPRODOTTO

GROUP BY A.CATEGORIAPRODOTTO

****

**6)QUALI SONO I PRODOTTI INVENDUTI**

SELECT P.\*

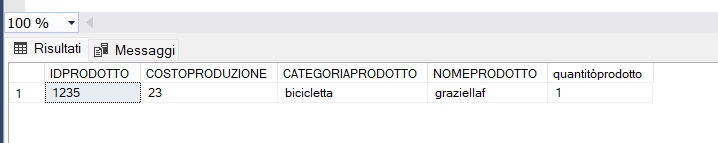
FROM PRODOTTI AS P

WHERE P.IDPRODOTTO NOT IN (SELECT IDPRODOTTO FROM VENDITE)

SELECT P.\*

FROM PRODOTTI AS P

LEFT JOIN VENDITE AS V ON P.IDPRODOTTO = V.IDPRODOTTO

WHERE V.IDPRODOTTO IS NULL****

**7)**

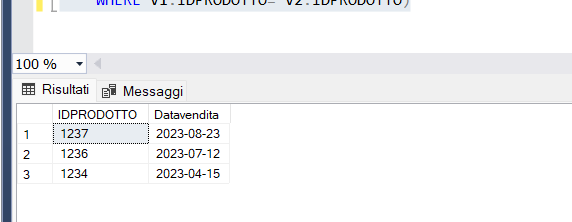
SELECT V1.IDPRODOTTO, V1.Datavendita

FROM VENDITE AS V1

WHERE V1.Datavendita = (SELECT MAX(V2.Datavendita) AS DATADIVENDITA

FROM VENDITE AS V2

WHERE V1.IDPRODOTTO= V2.IDPRODOTTO)

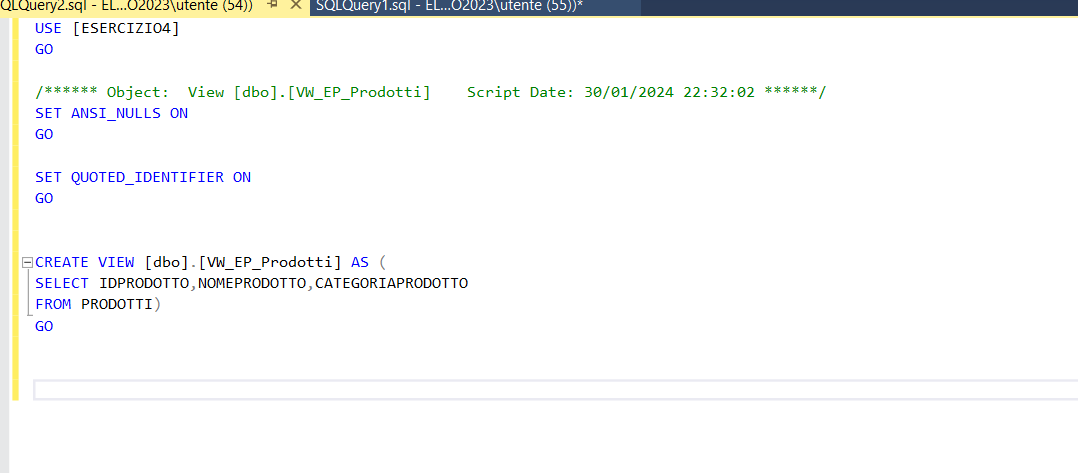
****

**8)**

CREATE VIEW VW\_EP\_Prodotti AS (

SELECT IDPRODOTTO,NOMEPRODOTTO,CATEGORIAPRODOTTO

FROM PRODOTTI)

****

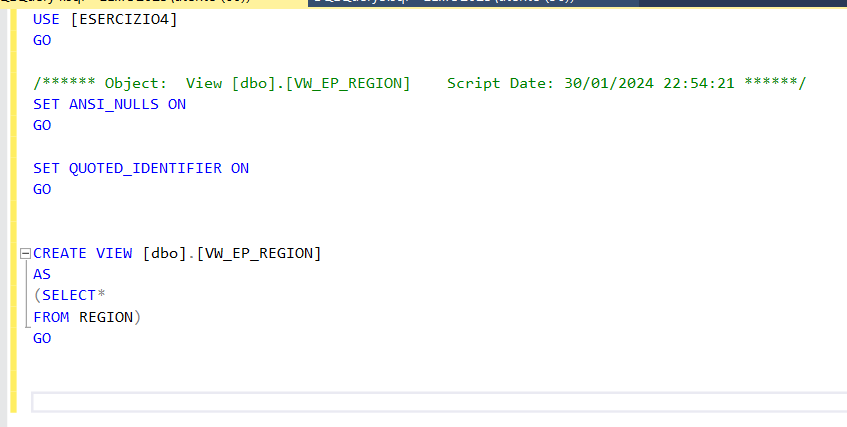
**9)**

CREATE VIEW VW\_EP\_REGION

AS

(SELECT\*

FROM REGION)

****