CREATE DATABASE TOYS

USE TOYS

CREATE TABLE Product

(ProductID INT PRIMARY KEY,

ProductName VARCHAR(30),

OrderQuantity INT,

ListPrice DECIMAL,

Category VARCHAR (30)

);

ALTER TABLE Product

ALTER COLUMN ListPrice DECIMAL (5,2)

CREATE TABLE Region

(TerritoryID INT PRIMARY KEY,

RegionName VARCHAR (25),

StateName VARCHAR (25)

);

CREATE TABLE Sales1

(ShopID INT PRIMARY KEY,

SalesOrderNumber INT,

ProductID INT FOREIGN KEY (ProductID) REFERENCES Product (ProductID),

OrderDate DATE,

TotalProductCost INT,

SalesAmount DECIMAL (8,2),

TerritoryID INT FOREIGN KEY (TerritoryID) REFERENCES Region (TerritoryID)

);

INSERT INTO Product (ProductID,ProductName,OrderQuantity,ListPrice,Category)

VALUES (1,'Cluedo',5,40.00,'Board Games'),

(2,'Monopoly',8,30.00,'Board Games'),

(3,'Scarabeo',null,35.90,'Board Games'),

(4,'Sapientino',2,24.99,'Childrens-games' ),

(5,'Barbie',3,34.99,'Childrens-games' ),

(6,'YU-GI-OH!MONDO-OSCURO',3,14.99,'CardGames'),

(7,'CARTEMARVELMISSIONARENATRADING',4,5.99,'CardGames'),

(8,'POKÉMONTEMPESTAARGENTATA',1,30.00,'CardGames'),

(9,'LEGOSTARWARS',2,259.00,'Collectible-Lego'),

(10, 'Millennium Falcon',2,169.99,'Collectible-Lego') ;

INSERT INTO REGION (TerritoryID,RegionName,StateName)

VALUES

(30,'Italia','Europa'),

(31,'Francia','Europa'),

(32,'India','Asia'),

(33,'United States','North America');

INSERT INTO Sales1 (ShopID,ProductID,OrderDate,TotalProductCost,SalesAmount,TerritoryID,SalesOrderNumber)

VALUES

(140,1,'2023/11/01',200,260,30,55),

(141,2,'2023/11/22',240,312,30,56),

(142,4,'2023/12/02',49,98,31,57),

(144,1,'2023/12/15' ,200,260,31,58),

(143,5,'2024/01/5',104.97,135.97,33,59),

(145,7,'2024/01/10',23.96,31,33,60),

(146,8,'2024/01/15',30,60,32,61),

(147,9,'2024/01/18',518,580,30,62),

(148,10,'2024/01/22',338,405,32,63);

/\*

1)Verificare che i campi definiti come PK siano univoci.

In altre parole, scrivi una query per determinare l’univocità dei valori di ciascuna PK (una query per tabella implementata).

\*/

SELECT COUNT(\*),ProductID

FROM Product

GROUP BY ProductID

HAVING COUNT (\*) >1

SELECT COUNT(\*),TerritoryID

FROM Region

GROUP BY TerritoryID

HAVING COUNT (\*) >1

SELECT COUNT(\*),ShopID

FROM Sales1

GROUP BY ShopID

HAVING COUNT (\*) >1

/\*2) Esporre l’elenco delle transazioni indicando nel result set il codice documento,

la data, il nome del prodotto, la categoria del prodotto, il nome dello stato,

il nome della regione di vendita e un campo booleano valorizzato in base alla condizione

che siano passati più di 180 giorni dalla data vendita o meno (>180 -> True, <= 180 -> False)

\*/

SELECT S.SalesAmount,S.SalesOrderNumber,S.OrderDate,P.ProductName,P.Category,R.StateName,R.RegionName,

CASE

WHEN DATEDIFF(DAY,GETDATE(),OrderDate) > 180 THEN 'TRUE'

ELSE 'FALSE'

END AS BOOLEANO

FROM Sales1 AS S

INNER JOIN Product AS P

ON S.ProductID=P.ProductID

INNER JOIN Region AS R

ON S.TerritoryID=R.TerritoryID

--3)Esporre l’elenco dei soli prodotti venduti e per ognuno di questi il fatturato totale per anno.

SELECT ProductID,

SUM (SalesAmount) AS FATTURATO,

YEAR (OrderDate) AS ANNO

FROM SALES1

GROUP BY ProductID, YEAR (OrderDate)

--4)Esporre il fatturato totale per stato per anno. Ordina il risultato per data e per fatturato decrescente

SELECT R.StateName,

SUM (S.SalesAmount) AS FATTURATO,

YEAR (OrderDate) AS ANNO

FROM Sales1 AS S

INNER JOIN Region AS R

ON S.TerritoryID= R.TerritoryID

GROUP BY R.StateName,YEAR (OrderDate)

ORDER BY YEAR (OrderDate),FATTURATO DESC

--5)Rispondere alla seguente domanda: qual è la categoria di articoli maggiormente richiesta dal mercato?

SELECT Category

FROM Product

WHERE OrderQuantity IN (

SELECT OrderQuantity

FROM Product

WHERE OrderQuantity > 4)

GROUP BY Category

/\*

6)Rispondere alla seguente domanda: quali sono, se ci sono, i prodotti invenduti?

Proponi due approcci risolutivi differenti.

\*/

SELECT P.ProductName,P.ProductID, S.SalesAmount

FROM Product AS P

LEFT OUTER JOIN Sales1 AS S

ON P.ProductID = S.ProductID

WHERE S.SalesAmount IS NULL

--7)Esporre l’elenco dei prodotti con la rispettiva ultima data di vendita (la data di vendita più recente).

SELECT P.ProductID, P.ProductName,

MAX (S.OrderDate) AS ULTIMAVENDITA

FROM Product AS P

INNER JOIN Sales1 AS S

ON P.ProductID=S.ProductID

GROUP BY P.ProductID,P.ProductName

ORDER BY ULTIMAVENDITA DESC

/\* 8)Creare una vista sui prodotti in modo tale da esporre una “versione denormalizzata” delle informazioni utili

(codice prodotto, nome prodotto, nome categoria)

\*/

CREATE VIEW VW\_LP\_INFOPRODOTTO AS

(SELECT ProductID,ProductName,Category

FROM Product );

SELECT \* FROM VW\_LP\_INFOPRODOTTO

--9)Creare una vista per restituire una versione “denormalizzata” delle informazioni geografiche

CREATE VIEW VW\_LP\_INFOGEO1 AS (

SELECT P.ProductID,P.ProductName,P.Category,R.TerritoryID,RegionName,StateName,S.SalesAmount,OrderDate

FROM Region AS R

INNER JOIN Sales1 AS S

ON R.TerritoryID =S.TerritoryID

INNER JOIN Product AS P

ON P.ProductID= S.ProductID );

SELECT\* FROM VW\_LP\_INFOGEO1