**Questions**

1. Cosa si intende per database?
   1. Una serie di informazioni strutturate codificate in tabelle
2. Cos’è un DBMS?
   1. Una Database management system, cioè un sistema di gestione del database
3. Indica le principali clausole di uno statement SELECT in ordine di esecuzione logica. Descrivi per ciascuna delle clausole indicate la logica di funzionamento.
   1. SELECT – definisco cosa vada selezionato
   2. FROM – definisco dove sia l’informazione da selezionare
   3. WHERE – definisco le condizioni secondo cui devono essere selezionate le informazioni
   4. GROUP BY – raggruppo le righe secondo un criterio
   5. HAVING – filtro i raggruppamenti
   6. ORDER BY – ordino secondo i valori di determinate colonne
4. Descrivi, immaginando uno scenario a te familiare, il concetto di group by. Utilizza l’approccio che ritieni più efficiente per trasmettere il concetto (suggerimento: disegna anche una sola tabella in Excel o in word con poche colonne e pochi record e descrivi, basandosi sulla tabella stessa, un esempio di group by).
   1. La group by raggruppa le informazioni contenute nelle colonne di una tabella come farebbe una Pivot di Excel, allo scopo di restituire informazioni utili sull’aggregazione (ad. Esempio il conteggio dei valori).

Immagine che contiene testo, Carattere, numero, software

Descrizione generata automaticamente

1. Descrivi la differenza tra uno schema OLTP e uno schema OLAP.
   1. OLTP è una base dati progettata per ottimizzare le CRUD. I dati al suo interno sono normalizzati
   2. OLAP è uno schema che integra i dati derivanti da più fonti. Denormalizzando facilita l’analisi anche se può creare ridondanza
2. Dato un medesimo scenario di analisi, qual è la differenza in termini di risultato ottenibile tra una join e una subquery?
   1. Nessuna, se lo scenario è lo stesso posso ottenere lo stesso risultato con una join o una subquery
3. Cosa si intende per DML e DDL?
   1. DML è il data management language, e si riferisce ai comandi che lavorano sui dati contenuti nelle tabelle
   2. DDL è il data definition language e si riferisce ai comandi che lavorano sulla struttura degli elementi che contengono i dati (es. ALTER TABLE)
4. Quali istruzioni possono utilizzare per estrarre l’anno da un campo data? Proponi degli esempi.
   1. SELECT YEAR(Data)
   2. FROM a
   3. SELECT DATENAME(year, Data)
   4. FROM a
   5. SELECT DATEPART (year, Data)
   6. FROM a
5. Qual è la differenza tra gli operatori logici AND e OR?
   1. AND restituisce TRUE quando si presentano entrambe le condizioni
   2. OR restituisce TRUE quando si presenta anche una sola delle due condizioni
6. È possibile innestare una query nella clausola SELECT?
   1. Si, quando voglio esporre i valori della subquery
7. Qual è la differenza tra l’operatore logico OR e l’operatore logico IN?
   1. OR restituisce TRUE quando si presenta anche una sola delle due condizioni
   2. IN restituisce TRUE quando il valore è uguale ad uno dei valori dell’oggetto di IN
8. L’operatore logico BETWEEN include anche gli estremi del range specificato?
   1. Si
9. Che cos’è una SELF JOIN? Proponi un esempio.

Situazione in cui, ho una colonna parent e una colonna child in una tabella.

![Immagine che contiene testo, schermata, linea, numero

Descrizione generata automaticamente]()

In questo caso mi serve una self join per estrarre il nome del manager dalla colonna “employee”, quindi scriverò:

SELECT \*

FROM Employee AS e1

JOIN Employee AS e2 ON e1.ID = e2.ManagerID

CASE STUDY

CREATE TABLE Category (

ID\_Category INT

IDENTITY (1,1),

CategoryName VARCHAR (50),

CONSTRAINT PK\_Category PRIMARY KEY (ID\_Category)

);

INSERT INTO Category

SELECT EnglishProductCategoryName

FROM AdventureWorksDW2022.dbo.DimProductCategory

CREATE TABLE Product (

ID\_Product INT

IDENTITY (1,1),

ID\_Category INT,

ProductName VARCHAR (50),

Cost DEC (5,2),

CONSTRAINT PK\_Product PRIMARY KEY (ID\_Product),

CONSTRAINT FK\_Product\_Category FOREIGN KEY (ID\_Category)

REFERENCES Category (ID\_Category)

);

INSERT INTO Product

VALUES

(1, 'Adjustable Race', 34.2),

(2, 'Bearing Ball', 15.5),

(3, 'BB Ball Bearing', 9),

(4, 'LL Crankarm', 22.9)

CREATE TABLE State (

ID\_State INT

IDENTITY (1,1),

StateName VARCHAR (50)

CONSTRAINT PK\_State PRIMARY KEY (ID\_State)

);

INSERT INTO State

VALUES

('Italy'), ('France'), ('Spain'), ('Bulgaria'), ('Germany'), ('Ireland')

CREATE TABLE Region (

ID\_Region INT

IDENTITY (1,1),

ID\_State INT,

RegionName VARCHAR (50),

CONSTRAINT PK\_Region PRIMARY KEY (ID\_Region),

CONSTRAINT FK\_Region\_State FOREIGN KEY (ID\_State)

REFERENCES State (ID\_State)

);

INSERT INTO Region

VALUES

(1, 'South Europe'),

(2, 'Centre Europe'),

(3, 'West Europe'),

(4, 'East Europe'),

(5, 'Centre Europe'),

(6, 'North Europe')

CREATE TABLE Sales (

ID\_Sale INT

IDENTITY (1,1),

ID\_Product INT,

ID\_Region INT,

SalesQuantity INT,

SellerName VARCHAR (50)

CONSTRAINT PK\_Sales PRIMARY KEY (ID\_Sale)

CONSTRAINT FK\_Product\_Sales FOREIGN KEY (ID\_Product)

REFERENCES Product (ID\_Product),

CONSTRAINT FK\_Sales\_Region FOREIGN KEY (ID\_Region)

REFERENCES Region (ID\_Region)

);

--1) Verificare che i campi definiti come PK siano univoci. In altre parole, scrivi una query per determinare

INSERT INTO Sales

VALUES

(1, 6, 22, 'Andrea'),

(2, 5, 15,'Franco'),

(3, 4, 35, 'Luisa'),

(4, 3, 102, 'Gina'),

(2, 2, 4, 'Piero')

SELECT ID\_Product

, COUNT (\*) AS Conteggio

FROM Product

GROUP BY ID\_Product

HAVING COUNT (\*)>1

SELECT ID\_Category

, COUNT (\*) AS Conteggio

FROM Category

GROUP BY ID\_Category

HAVING COUNT (\*)>1

SELECT ID\_Region

, COUNT (\*) AS Conteggio

FROM Region

GROUP BY ID\_Region

HAVING COUNT (\*)>1

SELECT ID\_Sale

, COUNT (\*) AS Conteggio

FROM Sales

GROUP BY ID\_Sale

HAVING COUNT (\*)>1

SELECT ID\_State

, COUNT (\*) AS Conteggio

FROM State

GROUP BY ID\_State

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)

\*/

ALTER TABLE Sales

ADD SaleDate DATE

UPDATE Sales

SET SaleDate = '2023-02-06'

WHERE ID\_Sale = 1

UPDATE Sales

SET SaleDate = '2022-04-15'

WHERE ID\_Sale = 2

UPDATE Sales

SET SaleDate = '2023-09-8'

WHERE ID\_Sale = 3

UPDATE Sales

SET SaleDate = '2021-10-07'

WHERE ID\_Sale = 4

UPDATE Sales

SET SaleDate = '2022-12-01'

WHERE ID\_Sale = 5

SELECT\*

FROM Sales

SELECT sal.ID\_Sale

, sal.SaleDate

, p.ProductName

, c.CategoryName

, st.StateName

, r.RegionName

, CASE

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

ELSE 'FALSE'

END AS SixMonth

FROM Sales AS sal

JOIN Product AS p ON sal.ID\_Product = p.ID\_Product

JOIN Region AS r ON sal.ID\_Region = r.ID\_Region

JOIN Category AS c ON p.ID\_Category = c.ID\_Category

JOIN State AS st ON r.ID\_State = st.ID\_State

/\*

3)Esporre l’elenco dei soli prodotti venduti

e per ognuno di questi il fatturato totale per anno

\*/

SELECT p.ProductName

, YEAR(s.SaleDate) AS SalesYear

, s.SalesQuantity \* p.Cost AS Fatturato

FROM Product AS p

JOIN Sales AS s ON p.ID\_Product = s.ID\_Product

GROUP BY p.ProductName, YEAR(s.SaleDate), s.SalesQuantity, p.Cost

/\*

4)Esporre il fatturato totale per stato per anno.

Ordina il risultato per data e per fatturato decrescente.

\*/

SELECT st.StateName

, YEAR(s.SaleDate) AS SaleYear

, SalesQuantity \* p.Cost AS Fatturato

FROM State AS st

LEFT JOIN Region AS r ON st.ID\_State = r.ID\_State

LEFT JOIN Sales AS s ON r.ID\_Region = s.ID\_Region

LEFT JOIN Product AS p ON s.ID\_Product = p.ID\_Product

GROUP BY st.StateName, YEAR(s.SaleDate), s.SalesQuantity,p.Cost

ORDER BY SaleYear, Fatturato DESC

/\*

5)Qual è la categoria di articoli maggiormente richiesta dal mercato?

1- tra le categorie

2- il prodotto che ha venduto di più

\*/

SELECT TOP 1 c.CategoryName,

MAX(s.SalesQuantity) AS TopOrder

FROM Category AS c

JOIN Product AS p ON c.ID\_Category = p.ID\_Category

JOIN Sales AS s ON p.ID\_Product = s.ID\_Product

GROUP BY c.CategoryName

/\*

6)quali sono, se ci sono, i prodotti invenduti?

Proponi due approcci risolutivi differenti.

\*/

SELECT p.ProductName

FROM Product AS p

LEFT JOIN Sales AS s ON p.ID\_Product = s.ID\_Sale

WHERE s.ID\_Sale IS NULL

SELECT p.ProductName

FROM Product AS p

WHERE ID\_Product = (

SELECT ID\_Product

FROM Sales AS s

WHERE p.ID\_Product = s.ID\_Product

AND

s.ID\_Product IS NULL

)

/\*

7)Esporre l’elenco dei prodotti con la rispettiva ultima data di vendita

(la data di vendita più recente)

\*/

SELECT p.ProductName, s1.SaleDate

FROM Product AS p

JOIN Sales AS s1 ON p.ID\_Product = s1.ID\_Product

WHERE s1.SaleDate = (

SELECT MAX(s2.SaleDate)

FROM Sales AS s2

WHERE s1.ID\_Sale = s2.ID\_Sale

)

/\*

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 ProdottoCategoria AS

SELECT DISTINCT Product.ID\_Product

, Product.ProductName

, Category.CategoryName

FROM Product

LEFT JOIN Category ON Product.ID\_Category = Category.ID\_Category

LEFT JOIN Sales ON Product.ID\_Product = Sales.ID\_Product

/\*

9) Creare una vista per restituire una versione “denormalizzata”

delle informazioni geografiche

\*/

CREATE VIEW Regione\_Stato AS

SELECT Region.ID\_Region

, Region.RegionName

, State.ID\_State

, State.StateName

, Sales.ID\_Sale

FROM Region

JOIN State ON Region.ID\_State = State.ID\_State

JOIN Sales ON Region.ID\_Region = Sales.ID\_Sale