

## Questions

- 1) Cosa si intende per database?
- 2) Cos'è un DBMS?
- 3) Indica le principali clausole di uno statement SELECT in ordine di esecuzione logica. Descrivi per ciascuna delle clausole indicate la logica di funzionamento.
- 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).
- 5) Descrivi la differenza tra uno schema OLTP e uno schema OLAP.
- 6) Dato un medesimo scenario di analisi, qual è la differenza in termini di risultato ottenibile tra una join e una subquery?
- 7) Cosa si intende per DML e DDL?
- 8) Quali istruzioni possono utilizzare per estrarre l'anno da un campo data? Proponi degli esempi.
- 9) Qual è la differenza tra gli operatori logici AND e OR?
- 10) È possibile innestare una query nella clausola SELECT?
- 11) Qual è la differenza tra l'operatore logico OR e l'operatore logico IN?
- 12) L'operatore logico BETWEEN include anche gli estremi del range specificato?

1. Un database (o anche base dati) è una raccolta, un contenitore di dati organizzati in modo logico e coerente al fine di gestirli, manipolarli e interrogarli in modo corretto. Un database può essere relazionale (i cui dati sono organizzati in tabelle collegate da relazioni) o non relazionale.
2. Un DBMS, ovvero database management system, è un software progettato per creare, manipolare e interrogare il database. I DBMS possono essere relazionali (RDBMS) come MySQL.
3. L'ordine logico di esecuzione delle clausole di uno statement SELECT è:
  - **FROM**: indica la tabella o le tabelle da cui vengono estratti i dati.
  - **JOIN**: consente di mettere in unione più tabelle tra loro. Può essere di più tipi: CROSS JOIN prende tutte le possibili combinazioni dei record delle tabelle, INNER JOIN prende solo i record delle tabelle che hanno corrispondenza, ovvero che verificano la condizione scritta in ON, che lega la chiave primaria e la chiave esterna delle tabelle considerate, LEFT OR RIGHT JOIN prendono rispettivamente tutti i record della tabella di sinistra o di destra, se non c'è corrispondenza viene scritto null.
  - **WHERE**: filtra i record in base a una o più condizioni di ricerca.
  - **GROUP BY**: raggruppa i record in base a uno o più campi, si utilizza in presenza di formule di aggregazione (sum, avg, count, min, max).
  - **HAVING**: filtra i dati aggregate restituiti dal group by in base a una o più condizioni di ricerca.
  - **SELECT**: è il primo campo che viene scritto nella query ma viene eseguito alla fine e il suo compito è quello di selezionare i campi da visualizzare nel result set.
  - **ORDER BY**: ordina il risultato in base a uno o più campi, in modo crescente (ASC) o decrescente (DESC).  
Ad ORDER BY può succedere **LIMIT** che si utilizza per ottenere dalla query un numero di righe specifico.

4. L'istruzione GROUP BY ci consente di creare dei gruppi di dati in base a uno o più campi della tabella considerata. Si utilizza in presenza di funzioni di aggregazione quali la somma, la media, il conta, il minimo e il massimo, che sono funzioni le quali dati un numero indefinito di record restituiscono un valore sintetico.

Prendiamo in considerazione la seguente tabella prodotti (il codice è presente nel file sql 'EsercitazioneFinale2') che contiene il codice ID del prodotto, il nome, la categoria, il prezzo e la quantità presente in negozio:

	id	nome	categoria	prezzo	quantita
►	1	Laptop	Elettronica	1200.99	10
	2	Smartphone	Elettronica	799.49	25
	3	Frigorifero	Elettrodomestici	499.99	5
	4	Tavolo	Arredamento	150.00	12
	5	Sedia	Arredamento	75.50	30
	6	Libro	Cultura	19.99	50

Supponiamo di voler determinare la quantità totale di prodotti presenti per categoria.

```
SELECT categoria, SUM(quantita) AS quantita_totale
FROM prodotti
GROUP BY categoria;
```

Il risultato che otteniamo con questo codice è il seguente:

	categoria	quantita_totale
►	Elettronica	35
	Elettrodomestici	5
	Arredamento	42
	Cultura	50

Per ogni categoria abbiamo la somma della quantità dei prodotti afferenti a quella categoria.

## 5. Schema OLTP e Schema OLAP

- **Schema OLTP** (online transactional processing) rappresenta la gestione dei dati transazionali, ovvero è il processo con il quale si gestisce la lavorazione dei dati. Ha lo scopo di garantire consistenza, integrità e sicurezza delle transazioni. Questo schema è tipico dei database dove ci sono intense attività di scrittura, ovvero operazioni brevi ma frequenti.
- **Schema OLAP** (online analytical processing) rappresenta l'approccio all'analisi complessa di grandi volume di dati a supporto della business intelligence. È basato su data warehouse che rappresenta un magazzino dei dati ottimizzati per l'analisi dei dati e quindi a supporto della BI.

6. Una **JOIN** combina, unisce i dati da due o più tabelle in base a una condizione di corrispondenza, la condizione di ON, che appunto verifica la corrispondenza tra la chiave primaria e la chiave esterna delle tabelle considerate. Una **SUBQUERY** è una sottointerrogazione, ovvero una query annidata all'interno di un'altra query e può essere utilizzata come valore di confronto in una clausola WHERE o nella SELECT.

La JOIN restituisce un'unica tabella con i dati combinati delle tabelle coinvolte. È utile quando si vuole ottenere dati da più tabelle contemporaneamente. Invece la SUBQUERY generalmente restituisce un singolo valore o un insieme di valori, che sono poi l'input della query esterna.

La join è più efficiente, specialmente quando il numero di record è molto elevato.

La subquery può essere utile quando si vuole filtrare i risultati in modo dinamico, ad esempio utilizzando un calcolo complesso che non è direttamente connesso alla tabella principale della query.

Vediamo un esempio in cui join e subquery restituiscono lo stesso risultato. (lo script è nel file 'EsercitazioneFinale2').

```
64 -- vogliamo i dettagli degli ordini (nome prodotto, quantità)
65 -- per i prodotti della categoria "Cancelleria" e con prezzo maggiore di 5.
66
67 -- risoluzione con la join
68 • SELECT prodotti2.nome, ordini.quantita
69 FROM ordini
70 JOIN prodotti2 ON ordini.prodID = prodotti2.ID
71 WHERE prodotti2.categoria = 'Cancelleria' AND prodotti2.prezzo > 5;
72
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
nome	quantita			
▶ Cartella	4			

```
73 -- risoluzione con la subquery (ho bisogno di due subquery)
74 • SELECT (SELECT nome FROM prodotti2 WHERE prodotti2.ID = ordini.prodID
75 AND categoria = 'Cancelleria' AND prezzo > 5) AS nome_prodotto, quantita
76 FROM ordini
77 WHERE prodID IN ( SELECT ID FROM prodotti2
78 WHERE categoria = 'Cancelleria' AND prezzo > 5);
79
80
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
nome_prodotto	quantita			
▶ Cartella	4			

Il risultato che otteniamo è lo stesso con entrambi i codici.

7. Per **DML (Data Manipulation Language)** si intende la parte del linguaggio SQL che si occupa della manipolazione dei dati nel database. Le istruzioni DML includono:

**INSERT**: aggiungere nuovi record a una tabella.

**UPDATE**: modificare i dati esistenti nelle tabelle.

**DELETE**: rimuovere i dati da una tabella.

**SELECT**: recuperare dati dal database.

Per **DDL (Data Definition Language)** si intende la parte di SQL che si occupa della definizione e gestione della struttura del database. Le istruzioni DDL includono:

**CREATE**: crea un nuovo database, una nuova tabella o altro oggetto (es. view) del database.

**ALTER**: modifica la struttura di una tabella esistente.

**DROP**: elimina una tabella o un oggetto del database.

**TRUNCATE**: elimina tutti i dati da una tabella, ma senza rimuoverla.

8. Per estrarre il campo anno da una data si utilizza la funzione YEAR() che ha come argomento una data.

```
SELECT YEAR('2025-02-21') AS anno;  
SELECT YEAR(CURDATE()) AS anno_data_corrente;  
SELECT YEAR(NOW()) AS anno_data_odierna;
```

Nella prima query ho inserito una specificata data, nella seconda query la funzione CURDATE() restituisce la data corrente nel momento della esecuzione della query, nella terza query la funzione NOW() restituisce la data corrente nel formato datetime che presenta sia la parte della data che del tempo.

Il risultato in tutti i casi è:

	anno
▶	2025

9. L'operatore logico AND restituisce true se entrambe le condizioni sono true.  
L'operatore logico OR restituisce true se almeno una delle condizioni è true.  
Vedi esempi in FILE SQL 'EsercitazioneFinale2'
10. Sì, è possibile inserire una query nella select, ad esempio quando si vuole ottenere un valore che deriva da un'altra tabella per ogni riga.  
Vedi esempi nel FILE SQL 'EsercitazioneFinale2'.

11. L'operatore logico OR restituisce true se almeno una delle condizioni è true.

L'operatore IN restituisce true se l'operando è presente nell'elenco considerato.

La differenza è che l'operatore OR confronta più condizioni singolarmente, mentre l'operatore IN controlla la presenza di un valore in un insieme.

Vediamo un esempio sempre riferito alla tabella prodotti precedente: si vuole visualizzare i prodotti che hanno come categoria abbigliamento o elettronica.

Possiamo scrivere i seguenti codici:

```
SELECT * FROM prodotti WHERE categoria = 'Elettronica' OR categoria = 'Abbigliamento';  
SELECT * FROM prodotti WHERE categoria IN ('Elettronica', 'Abbigliamento');
```

Il result set per entrambe le query è lo stesso:

	id	nome	categoria	prezzo	quantita
▶	1	Laptop	Elettronica	1200.99	10
	2	Smartphone	Elettronica	799.49	25
•	NULL	NULL	NULL	NULL	NULL

12. L'operatore logico BETWEEN include anche gli estremi del range specificato.

Ad esempio, considerata la tabella prodotti precedente:

```
SELECT * FROM prodotti  
WHERE prezzo BETWEEN 150 AND 1000;
```

Con questo codice otteniamo tutti i prodotti che hanno un prezzo compreso tra 150 e 1000, inclusi.

Il risultato della query è:

	id	nome	categoria	prezzo	quantita
▶	2	Smartphone	Elettronica	799.49	25
	3	Frigorifero	Elettrodomestici	499.99	5
	4	Tavolo	Arredamento	150.00	12
•	NULL	NULL	NULL	NULL	NULL

## Case Study

ToysGroup è un'azienda che distribuisce articoli (giocattoli) in diverse aree geografiche del mondo.

I prodotti sono classificati in categorie e i mercati di riferimento dell'azienda sono classificati in regioni di vendita.

In particolare:

1) Le entità individuabili in questo scenario sono le seguenti:

- Product
- Region
- Sales

2) Le relazioni tra le entità possono essere descritte nel modo seguente:

- Product e Sales
  - Un prodotto può essere venduto tante volte (o nessuna) per cui è contenuto in una o più transazioni di vendita.
  - Ciascuna transazione di vendita è riferita ad uno solo prodotto
- Region e Sales
  - Possono esserci molte o nessuna transazione per ciascuna regione
  - Ciascuna transazione di vendita è riferita ad una sola regione

3) Le entità Product e Region presentano delle gerarchie:

- L'entità prodotto contiene, oltre alle informazioni del singolo prodotto, anche la descrizione della categoria di appartenenza. L'entità prodotto contiene quindi una gerarchia: un prodotto può appartenere ad una sola categoria mentre la stessa categoria può essere associata a molti prodotti diversi.  
*Esempio: gli articoli 'Bikes-100' e 'Bikes-200' appartengono alla categoria Bikes; gli articoli 'Bike Glove M' e 'Bike Gloves L' sono classificati come Clothing.*
- L'entità regione contiene una gerarchia: più stati sono classificati in una stessa regione di vendita e una stessa regione di vendita include molti stati.  
*Esempio: gli stati 'France' e 'Germany' sono classificati nella region WestEurope; gli stati 'Italy' e 'Greece' sono classificati nel mercato SouthEurope.*

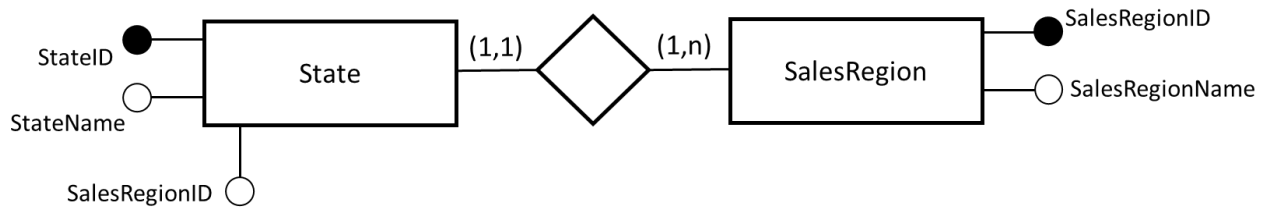
È necessario progettare e implementare fisicamente un database che modelli lo scenario garantendo l'**integrità referenziale** e la **minimizzazione della ridondanza dei dati**.

In altre parole, progetta opportunamente un numero di tabelle e di relazioni tra queste sufficiente a garantire la **consistenza del dato**.

### Task 1: Proponi una progettazione concettuale e logica della base dati

La progettazione concettuale deve includere tutte le entità coinvolte e le relazioni tra queste. Per ciascuna entità indica l'attributo chiave e i principali attributi descrittivi (non è necessario indicare tutti gli attributi).

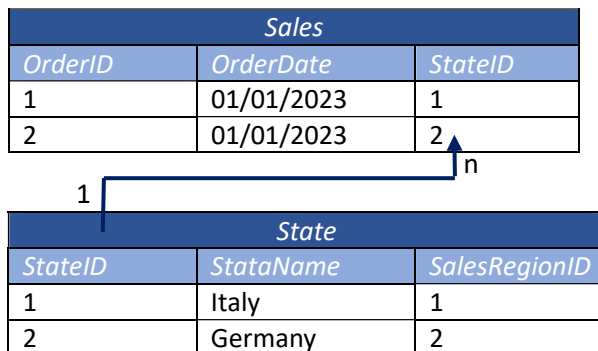
*Esempio di schema E/R*



*Lo schema proposto è puramente esemplificativo e non esaustivo o completo per la soluzione!*

La progettazione logica deve includere, per ciascuna tabella, tutte le colonne che poi verranno implementate fisicamente e deve esplicitare la cardinalità dei campi utilizzati per definire la relazione.

*Esempio di schema grafico delle tabelle e delle relazioni tra le stesse.*



*Il diagramma è puramente esemplificativo e non esaustivo o completo per la soluzione!*

**Task 2: Descrivi la struttura delle tabelle che reputeri utili e sufficienti a modellare lo scenario proposto tramite la sintassi DDL. Implementa fisicamente le tabelle utilizzando il DBMS SQL Server (o altro).**

```
CREATE TABLE table_name (  
    column1 datatype option,  
    column2 datatype option,  
    column3 datatype,  
    ....  
);
```

**Task 3: Popola le tabelle utilizzando dati a tua discrezione (sono sufficienti pochi record per tabella; riporta le query utilizzate)**

**Task 4: Dopo aver popolate le tabelle, scrivi delle query utili a:**

- 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).
- 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 ( $>180 \rightarrow \text{True}$ ,  $\leq 180 \rightarrow \text{False}$ )
- 3) Esporre l'elenco dei prodotti che hanno venduto, in totale, una quantità maggiore della media delle vendite realizzate nell'ultimo anno censito. (ogni valore della condizione deve risultare da una query e non deve essere inserito a mano). Nel result set devono comparire solo il codice prodotto e il totale venduto.
- 4) Esporre l'elenco dei soli prodotti venduti e per ognuno di questi il fatturato totale per anno.
- 5) Esporre il fatturato totale per stato per anno. Ordina il risultato per data e per fatturato decrescente.
- 6) Rispondere alla seguente domanda: qual è la categoria di articoli maggiormente richiesta dal mercato?
- 7) Rispondere alla seguente domanda: quali sono i prodotti invenduti? Proponi due approcci risolutivi differenti.
- 8) Creare una vista sui prodotti in modo tale da esporre una "versione denormalizzata" delle informazioni utili (codice prodotto, nome prodotto, nome categoria)

Creare una vista per le informazioni geografiche