1. Che cosa si intende per un database?

Un database è un sistema che accoglie e organizza dati in maniera strutturata, permettendo di gestirli in modo efficace. È costituito da un insieme di tabelle, dove ogni tabella rappresenta un’entità con i suoi attributi (o campi). Le tabelle all’interno di un database sono collegate tra loro attraverso delle relazioni che possono essere di diversi tipi, come uno-a-uno, uno-a-molti o molti-a-molti.

L’obiettivo di un database è rendere i dati organizzati in modo coerente e logico per facilitarne l’accesso, la gestione e l’aggiornamento delle informazioni.

1. Cos’è un DBMS?

Un DBMS (Database Management system) è un software che consente la gestione di un database permettendo di inserire, modificare, eliminare e consultare i dati. Supporta le operazioni CRUD🡪

* CREATE: creazione di dati o strutture;
* READER: lettura o consultazione dei dati;
* UPGRADE: modifica dei dati esistenti;
* DELETE: eliminazione dei dati.

I principali linguaggi utilizzati sono:

* DDL (Data Definition Language);
* DML (Data Manipulation Language);
* DCL (Data Control Language).

1. Indica le principali clausole di uno statement SELECT in ordine di esecuzione logica. Descrivi per ciascuna delle clausole indicate la logica di funzionamento.

* FROM🡪 rappresenta l’origine dei dati. Qui si possono includere operazioni come i JOIN, query innestate, UNION o l’uso di view per combinare e selezionare i dati da più tabelle.
* WHERE🡪 applica i filtri e le condizioni di ricerca sui dati selezionati dalla clausola FROM. Viene utilizzata per limitare il numero di righe restituite.
* GROUP BY🡪 consente di creare raggruppamenti o categorie sui dati, permettendo anche l’uso di funzioni di aggregazione (come COUNT, SUM, AVG) per operare su gruppi di righe.
* HAVING🡪 può essere simile a WHERE ma si applica sui risultati della GROUP BY. Quindi permette di filtrare i gruppi di dati limitando quelli restituiti sulla base di una condizione.
* SELECT🡪 può anche includere il costrutto CASE WHEN che permette di ripristiare i valori diversi in base a condizioni logiche.

CASE

WHEN <condizione 1> THEN <risultato 1>

WHEN <condizione 2> THEN <risultato 2>

...

WHEN <condizione n> THEN <risultato n>

ELSE <risultato di default>

END <nome colonna>

ESEMPIO:

Immaginiamo di avere a che fare con una tabella di vendite e vogliamo classificare ogni vendita in base all’importo. L’importo è classificato in basso (< 1000), medio (compreso tra 1000 e 5000) e alto (> 5000):

SELECT

nome\_cliente,

importo,

CASE

WHEN importo < 1000 THEN 'Basso'

WHEN importo BETWEEN 1000 AND 5000 THEN 'Medio'

ELSE 'Alto'

END AS categoria\_importo

FROM vendite;

* ORDER BY🡪 consente di ordinare i risultati in base a una o più colonne, in ordine crescente o decrescente.

1. Descrivi, immaginando uno scenario a te familiare, il concetto di group by. Utilizza l’approccio che ritieni più efficiente per trasmettere il concetto.

Vendita settimanale di vari tipi di frutta in un negozio.

|  |  |  |
| --- | --- | --- |
| TIPO DI FRUTTA | QUANTITA’ VENDUTA | PREZZO UNITARIO |
| Mela | 10 | 0.5 |
| Banana | 5 | 0.3 |
| Mela | 15 | 0.5 |
| Arancia | 8 | 0.4 |
| Banana | 7 | 0.3 |
| Mela | 12 | 0.5 |
| Arancia | 10 | 0.4 |

Vogliamo sapere quante mele/banane/arance sono state vendute in totale. Per ottenere questo risultato utilizzerò una query SQL con GROUP BY:

SELECT Tipo\_di\_Frutta, SUM(Quntita\_Venduta) AS Totale\_Venduta

FROM Vendite

Group by Tipo\_di\_Frutta

La query ci restituirà una visione complessiva delle vendite per ciascun tipo di frutta, consentendoci di analizzare quali prodotti stanno performando meglio.

1. Descrivi la differenza tra uno schema OLTP e uno schema OLAP.

* **OLTP** (Online Transaction Processing) sono sistemi progettati per gestire operazioni quotidiane e transazioni in tempo reale ottimizzati per operazioni di tipo CRUD (CREATE, READ, UPDATE, DELETE). L’obiettivo è quello di garantire l’integrità e la consistenza dei dati, ridurre al minimo la ridondanza dei dati e gestire un numero elevato di transazioni simultanee in modo efficiente.
* **OLAP** (Online Analytical Processing)sono sistemi progettati per l’analisi dei dati e la generazione di report, ottimizzati per query complesse che richiedono aggregazioni e analisi multidimensionali. L’obiettivo è quello di supportare l’analisi dei dati e la visualizzazione, fornire informazioni e facilitarne l’analisi storica e la sintesi dei dati.

A differenza dell’OLTP utilizza una struttura dei dati de normalizzata spesso in modelli a stella o a fiocco di neve, per ottimizzare le prestazioni della query di lettura. Inoltre, a differenza sempre dell’OLTP i tipi di operazioni sono più complesse e su larga scala (es: query per report mensili, analisi delle vendite per regione, trend storici).

1. Dato un medesimo scenario di analisi, qual è la differenza in termini di risultato ottenibile tra una join e una subquery?

Una JOIN è un’operazione che combina righe da due o più tabelle in base a una condizione comune. Le JOIN possono essere di diversi tipi: INNER JOIN, LEFT JOIN E RIGHT JOIN.

Quando si utilizza una JOIN il risultato è una tabella che include colonne da entrambe le tabelle coinvolte. Si possono selezionare colonne specifiche da entrambe le tabelle nella query.

Invece una subquery è una query annidata all’interno di un’altra query che può restituire un valore singolo, una lista di valori o un insieme di righe che possono essere utilizzate nella query principale. La subquery serve generalmente per filtrare i dati dalla tabella principale A in base ai risultati ottenuti dalla tabella secondaria B.

1. Cosa si intende per DML e DDL?

* **DML (Data Manipulation Language):** insieme di comandi SQL utilizzati per manipolare i dati all’interno delle tabelle di un database. Comandi principali:
* **INSERT INTO**🡪 consente di aggiunger nuove righe a una tabella;
* **UPDATE**🡪 modifica i dati esistenti di una tabella
* **DELETE**🡪 rimuove righe di una tabella
* **SELECT**🡪 recupera i dati da una o più tabelle.
* **DDL (Data Definition Language):** insieme di comandi SQL utilizzati per definire e gestire la struttura del database. I comandi principali sono:
* **CREATE**🡪 permette di creare nuovi oggetti nel database (tabelle, indici...)
* **ALTER**🡪 modifica la struttura di oggetti esistenti nel database
* **DROP** 🡪 elimina oggetti nel database
* **TRUNCATE**🡪 rimuove tutte le righe di una tabella senza eliminare la tabella.

1. Quali istruzione possono essere utilizzate per estrare l’anno da un campo data? Proponi degli esempi.

Possiamo utilizzare la funzione YEAR().

SELECT YEAR(data\_colonna) AS ANNO

FROM nome\_tabella;

1. Qual è la differenza tra gli operatori logici AND e OR?

L’operatore logico **AND** richiede che tutte le condizioni siano vere per restituire vero; mentre l’operatore **OR** restituisce vero se almeno una delle condizioni è vera.

1. È possibile innestare una query nella clausola SELECT?

Si.

1. Qual è la differenza tra l’operatore logico OR e l’operatore logico IN?

L’operatore **OR** è utile quando si devono combinare più condizioni mentre **IN** è più efficiente per verificare se un valore specifico è presente all’interno di un insieme di valori.

La sintassi🡪

|  |  |
| --- | --- |
| OR | IN |
| condizione1 OR condizione2 | valore IN(valore1,valore2,valore3) |

1. L’operatore logico BETWEEN include anche gli estremi del range specificato?

SI. Supponiamo ad esempio di avere una tabella chiamata PRODOTTI con una colonna chiamata PREZZO. Vogliamo selezionare tutti i prodotti il cui tutti i prodotti il cui prezzo è compreso tra 10 e 20 euro:

SELECT \* FROM PRODOTTI

WHERE prezzo BETWEEN 10 AND 20;