**TEST TEORICO MYSQL – EMANUELE BUEMI**

1. **Qual è la differenza tra INNER JOIN, LEFT JOIN, RIGHT JOIN e FULL JOIN in SQL?**

Una JOIN è un'operazione che combina righe da due o più tabelle in base a un campo comune tra le due tabelle relazionate. In particolare permette di trovare una corrispondenza sulla base delle chiavi primaria e esterna apparteneti alle due tabelle. Questo permette di creare una tabella temporanea contenente i dati correlati. In SQL, in particolare è possibile utilizzare diversi tipi di join per combinare i dati provenienti da due o più tabelle, tuttavia con differenze l’uno con l’altro.

* **INNER JOIN**: Restituisce solo le righe in cui c'è almeno una corrispondenza in entrambe le tabelle in base alla condizione di join. Se non viene trovata una riga corrispondente in una delle tabelle, quelle righe non vengono incluse nel set di risultati.
* **LEFT JOIN**: Restituisce tutte le righe dalla tabella sinistra (tabella1) e le righe corrispondenti dalla tabella destra (tabella2). Se non c'è corrispondenza per una riga dalla tabella sinistra nella tabella destra, vengono utilizzati valori NULL per le colonne dalla tabella destra.
* **RIGHT JOIN**: Rappresenta l’operazione opposta al LEFT JOIN. Restituisce tutte le righe dalla tabella destra (tabella2) e le righe corrispondenti dalla tabella sinistra (tabella1). Se non c'è corrispondenza per una riga dalla tabella destra nella tabella sinistra, vengono utilizzati anche in questo caso valori NULL per le colonne dalla tabella sinistra.
* **FULL JOIN**: Restituisce tutte le righe quando c'è una corrispondenza sia nella tabella sinistra che in quella destra. Se non c'è corrispondenza per una riga da una tabella nell'altra tabella, vengono utilizzati valori NULL per le colonne dall'altra tabella. È importante dire che MySQL non supporta l’operazione di FULL JOIN.

1. **Cosa fa l'istruzione ORDER BY in SQL e come viene utilizzata?**

L'istruzione ORDER BY in SQL viene utilizzata per ordinare i risultati di una query in base ai valori di una o più colonne specificate. Questo permette di visualizzare i risultati in un ordine specifico, che può essere crescente (ASC) o decrescente (DESC), a seconda delle esigenze. In molti casi per l’ordinamento viene anche inclusa la clausola LIMIT per selezionare un tot di record da estrarre.

1. **Differenza tra un database relazionale e non relazionale.**

Le differenze principali tra un database relazionale e uno non relazionale (o database NoSQL) riguardano principalmente la struttura dei dati, il modello di archiviazione e l'approccio alla scalabilità. I database relazionali seguono uno schema tabellare rigido e sono basati su SQL e relazioni tra tabelle, i database non relazionali offrono una maggiore flessibilità nella struttura dei dati e nell'approccio alla scalabilità, adattandosi meglio a determinati casi d'uso, come l'archiviazione di grandi volumi di dati non strutturati o semistrutturati e la gestione di carichi di lavoro distribuiti su larga scala.

1. **Differenza tra primary key e foreign key.**

La differenza principale tra una primary key (chiave primaria) e una foreign key (chiave esterna) risiede nel loro ruolo e nella relazione che stabiliscono tra le tabelle di un database relazionale. La primary key è un attributo o un insieme di attributi che identifica univocamente ogni riga o record in una tabella. Ogni tabella può avere una sola primary key, e ogni valore nella primary key deve essere unico all'interno della tabella. La foreign key è un attributo o un insieme di attributi che stabilisce una relazione tra due tabelle in un database relazionale. Una foreign key fa riferimento alla primary key di un'altra tabella, consentendo di stabilire una relazione padre-figlio tra le due tabelle.

Una cosa importante sulla primary key è che i suoi valori non possono essere null, poiché devono essere unici e identificare in modo univoco ciascuna riga, mentre la foreign key impone un vincolo di riferimento che assicura che i valori presenti nella foreign key esistano come valori nella primary key della tabella di riferimento o siano null, se consentito.

1. **Descrivi le operazioni di CRUD successivamente spiega il funzionamento della keyword WHERE.**

Le operazioni CRUD (Create, Read, Update, Delete) rappresentano le operazioni fondamentali eseguite su dati in un sistema di gestione di database relazionali (DBMS).

Descrizione delle operazioni:

* **Create (Creazione):** Questa operazione consiste nell'inserire nuovi dati nel database. Viene utilizzata per aggiungere nuove righe o record a una tabella esistente.
* **Read (Lettura):** Questa operazione consiste nel recuperare i dati esistenti dal database. Viene utilizzata per visualizzare i dati presenti nelle tabelle, eseguendo query che restituiscono risultati basati su determinati criteri di ricerca.
* **Update (Aggiornamento):** Questa operazione consiste nell'aggiornare i dati esistenti nel database. Viene utilizzata per modificare o aggiornare i valori delle colonne di una o più righe in base a determinati criteri.
* **Delete (Cancellazione):** Questa operazione consiste nell'eliminare dati esistenti dal database. Viene utilizzata per rimuovere righe o record da una tabella in base a determinati criteri.

La keyword WHERE è utilizzata per filtrare i dati durante le operazioni di lettura e cancellazione, consentendo di specificare criteri che devono essere soddisfatti affinché una riga o un record sia incluso o eliminato dalla query.