MySQL Subjective Question Paper

**Time: 60 minutes**  
**Total Marks: 50**

Section A: (30 Marks)

1. **Explain the concept of a database. Why is it important to use SQL in database management?**  
   \*[5 Marks]\*
2. **Describe the differences between DDL, DCL, DML, and DQL in SQL. Provide examples of commands used in each category.**  
   \*[5 Marks]\*
3. **Discuss the process of installing mySQL and mySqlWorkbench.** \*[5 Marks]\*
4. **Define the following terms and explain their significance in relational databases: Primary key, Foreign key, Unique key, and Null values.**  
   \*[5 Marks]\*
5. **Outline the steps involved in installing MySQL and creating a new database. Include details on setting up the initial configuration and creating a user.**  
   \*[5 Marks]\*
6. **Explain the different constraints that can be applied in MySQL. Illustrate your answer with examples of how each constraint is used when creating a table.**  
   \*[5 Marks]\*

Section B: (20 Marks)

1. **Describe the process of loading an existing database (such as the ‘employees’ database) into MySQL. What are the key considerations to ensure data integrity during this process?**  
   \*[5 Marks]\*
2. **Write an SQL query to retrieve the names of employees who have a salary greater than the average salary in the company. Use appropriate SQL functions and clauses in your query. Explain your query step-by-step.**  
   \*[5 Marks]\*
3. **Discuss the different types of joins in SQL. Provide examples of scenarios where each type of join would be most appropriate. Include SQL queries for Left Join, Right Join, Inner join.**  
   \*[5 Marks]\*
4. **Write a query to create a table named `employees` with columns `id`, `name`, `position`, `salary`, and `hire\_date`. And insert a 1 employees details.** \*[5 Marks]\*

Answer Key

Section A: Basics of Database, SQL, and MySQL

1. **Explain the concept of a database. Why is it important to use SQL in database management?**  
   A database is a structured collection of data that is stored and accessed electronically. Databases allow for efficient retrieval, insertion, and deletion of data. SQL (Structured Query Language) is important because it provides a standardized way to interact with relational databases, enabling users to perform operations such as querying, updating, and managing the data.
2. **Describe the differences between DDL, DCL, DML, and TCL in SQL. Provide examples of commands used in each category.**
   * DDL (Data Definition Language): Commands that define the structure of the database, such as CREATE, ALTER, DROP.
   * DCL (Data Control Language): Commands that control access to data, such as GRANT, REVOKE.
   * DML (Data Manipulation Language): Commands that manipulate data within the database, such as SELECT, INSERT, UPDATE, DELETE.
   * TCL (Transaction Control Language): Commands that manage transactions, such as COMMIT, ROLLBACK, SAVEPOINT.
3. **Discuss the process of installing PostgreSQL and pgAdmin4. Why might you choose PostgreSQL over MySQL for certain projects?**
   * Installation involves downloading PostgreSQL from the official website, running the installer, and configuring the initial settings. PgAdmin4 can be downloaded and installed separately to provide a graphical interface for managing PostgreSQL databases.
   * PostgreSQL might be chosen over MySQL for projects requiring advanced features like complex queries, full ACID compliance, extensibility, and support for advanced data types.

Section B: Relational Database Essentials and SQL Basics

1. **Define the following terms and explain their significance in relational databases: Primary key, Foreign key, Unique key, and Null values.**
   * Primary Key: A column or set of columns that uniquely identifies each row in a table.
   * Foreign Key: A column that creates a relationship between two tables by referencing the primary key in another table.
   * Unique Key: A constraint that ensures all values in a column are unique.
   * Null Values: Represent missing or unknown data in a column.
2. **Outline the steps involved in installing MySQL and creating a new database. Include details on setting up the initial configuration and creating a user.**
   * Download MySQL from the official website and run the installer.
   * Configure the server by setting the root password and default settings.
   * Start the MySQL service and log in using the root account.
   * Create a new database using the CREATE DATABASE command.
   * Create a new user with appropriate privileges using the CREATE USER and GRANT commands.
3. **Explain the different constraints that can be applied in MySQL. Illustrate your answer with examples of how each constraint is used when creating a table.**
   * Primary Key: PRIMARY KEY (id)
   * Foreign Key: FOREIGN KEY (dept\_id) REFERENCES departments(id)
   * Unique: UNIQUE (email)
   * Default: DEFAULT 'N/A'
   * Not Null: NOT NULL

Section C: Advanced SQL Queries and Operations

1. **Describe the process of loading an existing database (such as the ‘employees’ database) into MySQL. What are the key considerations to ensure data integrity during this process?**
   * Import the database using the mysql command-line tool or a graphical tool like MySQL Workbench.
   * Ensure the data types and constraints match those defined in the source database.
   * Verify the integrity of relationships between tables, especially foreign key constraints.
   * Check for any errors during the import process and resolve them before using the database.
2. **Write an SQL query to retrieve the names of employees who have a salary greater than the average salary in the company. Use appropriate SQL functions and clauses in your query. Explain your query step-by-step.**

SELECT name  
FROM employees  
WHERE salary > (SELECT AVG(salary) FROM employees);

* + The subquery calculates the average salary from the employees table.
  + The main query selects the names of employees whose salary is greater than the average salary.

1. **Discuss the different types of joins in SQL. Provide examples of scenarios where each type of join would be most appropriate. Include SQL queries for Left Join, Right Join, Cross Join, and Union.**
   * Left Join: Returns all rows from the left table and matching rows from the right table.

SELECT a.name, b.department  
FROM employees a  
LEFT JOIN departments b ON a.dept\_id = b.id;

* + Right Join: Returns all rows from the right table and matching rows from the left table.

SELECT a.name, b.department  
FROM employees a  
RIGHT JOIN departments b ON a.dept\_id = b.id;

* + Cross Join: Returns the Cartesian product of both tables.

SELECT a.name, b.department  
FROM employees a  
CROSS JOIN departments b;

* + Union: Combines the results of two or more SELECT statements.

SELECT name FROM employees1  
UNION  
SELECT name FROM employees2;