Assignment

Database

**Introduction to SQL**

**Theory Questions:**

1. What is SQL, and why is it essential in database management?

Ans. Structured Query Language (SQL) refers to a standard programming language utilized to extract, organize, manage, and manipulate data stored in relational databases.

SQL plays a essential role in retrieving relevant data from databases, which can later be used by various platforms such as Python or R for analysis purposes. SQL can manage several data transactions simultaneously where large volumes of data are written concurrently.

2. Explain the difference between DBMS and RDBMS.

Ans.

|  |  |
| --- | --- |
| DBMS | RDBMS |
| DBMS stores data as file. | RDBMS stores data in tabular form. |
| Data elements need to access individually. | Multiple data elements can be accessed at the same time. |
| No relationship between data. | Data is stored in the form of tables which are related to each other. |
| Normalization is not present. | Normalization is present. |
| DBMS does not support distributed database. | RDBMS supports distributed database. |

3. Describe the role of SQL in managing relational databases.

Ans. SQL (Structured Query Language) plays a fundamental role in managing relational databases. It is the standard programming language specifically designed for interacting with and manipulating relational database systems.

Functions of SQL:

1. Data Definition (DDL)

SQL provides commands to define the structure of a database, including creating, altering, and deleting tables, schemas, and other database objects. Examples:

CREATE TABLE for creating new tables.

ALTER TABLE for modifying existing table structures.

DROP TABLE for deleting tables.

2. Data Manipulation (DML)

SQL allows users to insert, update, delete, and retrieve data within the database. Examples:

INSERT INTO to add new records.

UPDATE to modify existing records.

DELETE to remove records.

SELECT to retrieve data, often combined with filters (WHERE), grouping (GROUP BY), sorting (ORDER BY), and aggregation functions.

3. Data Control (DCL)

SQL includes commands to manage user access and permissions:

GRANT to provide specific privileges to users or roles.

REVOKE to remove those privileges.

4. Transaction Control (TCL)

SQL helps ensure data integrity and consistency through transaction management:

BEGIN TRANSACTION, COMMIT, and ROLLBACK are used to control transactions, ensuring atomicity and recoverability.

4. What are the key features of SQL?

Ans.

1. Easy to Learn and Use: SQL is a declarative language, meaning you specify what you want to do, and the database system handles how to do it.Its syntax is simple and intuitive, making it accessible even for non-programmers.

2. Data Definition Language (DDL):SQL provides commands to define the structure of a database:

CREATE to create tables, views, and other database objects.

ALTER to modify existing objects.

DROP to delete database objects.

3. Data Manipulation Language (DML): It allows users to perform operations on the data stored in the database:

INSERT to add records.

UPDATE to modify existing records.

DELETE to remove records.

SELECT to retrieve data.

LAB EXERCISES:

Lab 1: Create a new database named school\_db and a table called students with the following columns: student\_id, student\_name, age, class, and address.

Ans. Database: CREATE DATABASE school\_db;

Table: CREATE TABLE students(

student\_id int PRIMARY KEY AUTO\_INCREMENT,

student\_name varchar (80),

student\_agebigint UNIQUE KEY,

student\_classbigint UNIQUE KEY,

student\_address varchar (80)

);

Lab 2: Insert five records into the students table and retrieve all records using the SELECT statement.

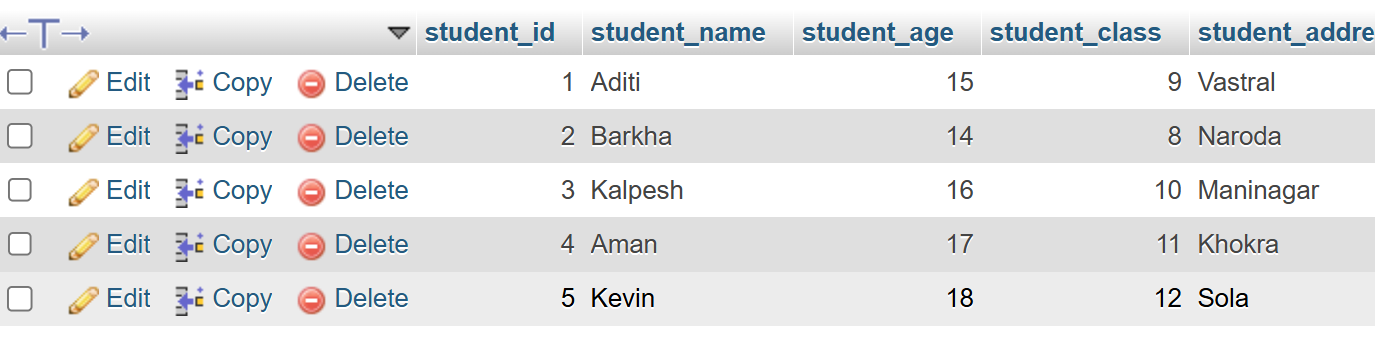
Ans. INSERT INTO students(student\_id,student\_name,student\_age,student\_class,student\_address) VALUES (1,"Aditi",15,9,"Vastral"),

(2,"Barkha",14,8,"Naroda"),

(3,"Kalpesh",16,10,"Maninagar"),

(4,"Aman",17,11,"Khokra"),

(5,"Kevin",18,12,"Sola");



2. SQL Syntax

Theory Questions:

1. What are the basic components of SQL syntax?

Ans. The basic syntax is as follows: SELECT columnName FROM yourTable WHERE CONTAINS ( columnName, 'yourSubstring' );

From the example above, you should note that the column name isn't enclosed in quotes, but the arguments for the CONTAINS SQL function are enclosed in parentheses.

2. Write the general structure of an SQL SELECT statement.

Ans. SELECT column1, column2, ...

FROM table\_name;

3. Explain the role of clauses in SQL statements.

Ans.

1. SELECT – Specifies the columns to retrieve from a table.

Eg: SELECT name, age FROM employees;

2. FROM – Identifies the table(s) from which data is fetched.

Eg: SELECT \* FROM employees;

3. WHERE – Filters records based on specific conditions.

Eg: SELECT \* FROM employees WHERE age > 30;

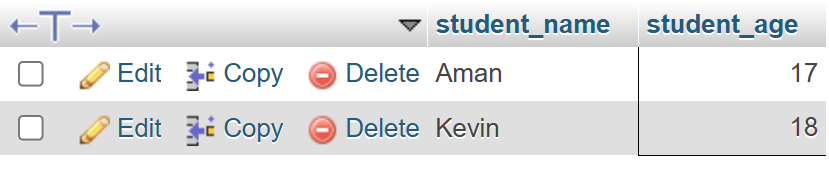
4. INSERT INTO – Adds new records into a table.

Eg: INSERT INTO employees (name, age, department) VALUES ('Alice', 28, 'HR');

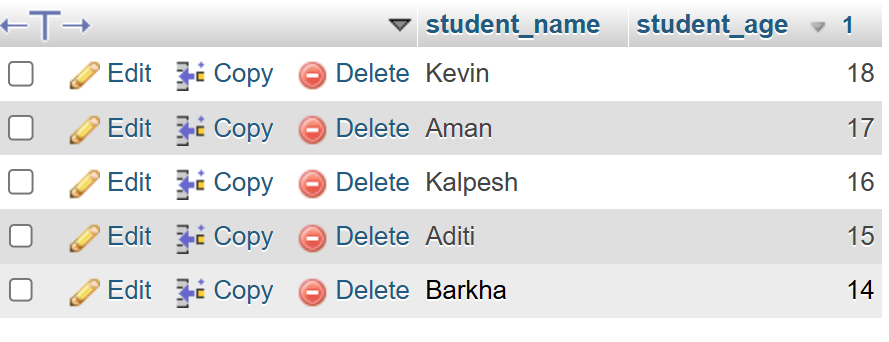
LAB EXERCISES:

Lab 1: Write SQL queries to retrieve specific columns (student\_name and age) from the students table.

Ans. SELECT student\_name, student\_age FROM students WHERE student\_age> 16;

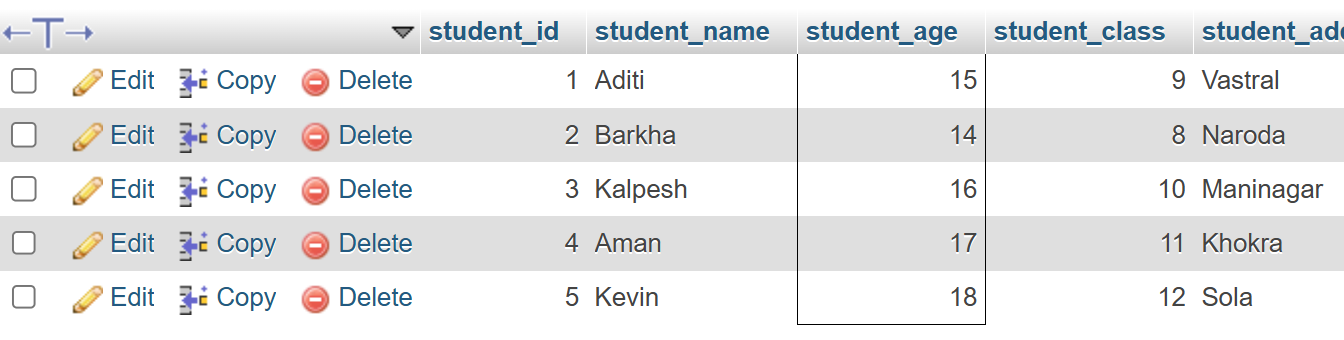


SELECT student\_name, student\_age FROM students ORDER BY student\_age DESC;



Lab 2: Write SQL queries to retrieve all students whose age is greater than 10.

Ans. SELECT \* FROM students WHERE student\_age> 10;



3. SQL Constraints

Theory Questions:

1. What are constraints in SQL? List and explain the different types of constraints.

Ans. Constraints in SQL are rules applied to columns in a database table to enforce data integrity, consistency, and accuracy. Constraints limit the type of data that can be inserted, updated, or deleted in a table, ensuring the validity of the data.

1. NOT NULL Constraint: Ensures that a column cannot have a NULL value. A value must be provided for the column.

2. UNIQUE Constraint: Ensures that all values in a column are unique, meaning no duplicate values are allowed. A column with a unique constraint can still contain NULL values (unless combined with NOT NULL).

3. PRIMARY KEY Constraint: Uniquely identifies each record in a table. It combines both NOT NULL and UNIQUE constraints. A table can have only one primary key, and it may consist of one or multiple columns (composite key).

2. How do PRIMARY KEY and FOREIGN KEY constraints differ?

Ans. Keys are one of the most important elements in a relational database to maintain the relationship between the tables and it also helps in uniquely identifying the data from a table. The primary key is a key that helps uniquely identify the tuple of the database. In contrast, the Foreign Key is a key used to determine the relationship between the tables through the primary key of one table that is the primary key of one table acts as a foreign key to another table.

3. What is the role of NOT NULL and UNIQUE constraints?

Ans. NOT NULL Constraint:

Ensures that a column cannot have NULL values.

Every row must have a valid (non-null) value for this column.

Helps maintain data consistency by preventing missing or undefined values.

UNIQUE Constraint:

Ensures that all values in a column are distinct (no duplicates allowed).

Can be applied to one or multiple columns (when combined, values must be unique together).

Unlike PRIMARY KEY, a table can have multiple UNIQUE columns.

LAB EXERCISES:

Lab 1: Create a table teachers with the following columns: teacher\_id (Primary Key), teacher\_name (NOT NULL), subject (NOT NULL), and email (UNIQUE).

Ans. CREATE TABLE teachers (

teacher\_id INT PRIMARY KEY,

teacher\_nameVARCHAR(255) NOT NULL,

teacher\_subjectVARCHAR(255) NOT NULL,

teacher\_emailVARCHAR(255) UNIQUE

);

Lab 2: Implement a FOREIGN KEY constraint to relate the teacher\_id from the teachers table with the students table.

Ans.

ALTER TABLE students

ADD COLUMN teacher\_id INT;

ALTER TABLE students

ADD CONSTRAINT fk\_teacher

FOREIGN KEY (teacher\_id) REFERENCES teachers(teacher\_id);

4. Main SQL Commands and Sub-commands (DDL)

Theory Questions:

1. Define the SQL Data Definition Language (DDL).

Ans. SQL Data Definition Language (DDL) is a subset of SQL (Structured Query Language) used to define and manage the structure of a database. DDL statements are responsible for creating, altering, and deleting database objects such as tables, indexes, views, and schemas.

2. Explain the CREATE command and its syntax.

Ans. The CREATE command in SQL is a Data Definition Language (DDL) statement used to create new database objects such as tables, indexes, views, and schemas. It is fundamental for establishing the structure of a database and defining how data will be stored and organized.

Syntax: CREATE TABLE table\_name (

column1\_name data\_type [constraints],

column2\_name data\_type [constraints],

...

[table\_constraints]

);

3. What is the purpose of specifying data types and constraints during table creation?

Ans. Specifying data types and constraints during table creation in a relational database serves several important purposes. These elements are crucial for ensuring data integrity, optimizing performance, and defining the structure of the database.

LAB EXERCISES:

Lab 1: Create a table courses with columns: course\_id, course\_name, and course\_credits. Set the course\_id as the primary key.

Ans. CREATE TABLE courses (

course\_id INT PRIMARY KEY,

course\_nameVARCHAR(255) NOT NULL,

course\_credits INT NOT NULL

);

Lab 2: Use the CREATE command to create a database university\_db.

Ans. CREATE DATABASE university\_db;

5. ALTER Command

Theory Questions:

1. What is the use of the ALTER command in SQL?

Ans. The ALTER command in SQL is used to make changes to a table, view, or the entire database. We can add, modify, and drop constraints, columns, and indexes using the ALTER command in SQL.

2. How can you add, modify, and drop columns from a table using ALTER?

Ans. 1. Adding Columns: To add a new column to an existing table, you use the ADD clause.

Syntax: ALTER TABLE table\_name

ADD column\_namedata\_type [constraints];

2. Modifying Columns: To modify an existing column, you use the MODIFY (or ALTER COLUMN in some SQL dialects) clause. This can include changing the data type or altering constraints.

Syntax: ALTER TABLE table\_name

MODIFY column\_namenew\_data\_type [new\_constraints];

3. Dropping Columns: To remove an existing column from a table, you use the DROP COLUMN clause.

Syntax: ALTER TABLE table\_name

DROP COLUMN column\_name;

LAB EXERCISES:

Lab 1: Modify the courses table by adding a column course\_duration using the ALTER command.

Ans. ALTER TABLE courses

ADD course\_duration INT;

Lab 2: Drop the course\_credits column from the courses table.

Ans. ALTER TABLE courses

DROP COLUMN course\_credits;

6. DROP Command

Theory Questions:

1. What is the function of the DROP command in SQL?

Ans. In SQL, the DROP command is used to permanently remove an object from a database, such as a table, database, index, or view. When we DROP a table, both the data and the structure of the object are permanently removed from the database leaving no trace of the object.

Syntax: DROP object object\_name ;

2. What are the implications of dropping a table from a database?

Ans. Dropping a table invalidates dependent objects and removes object privileges on the table. If you want to re-create the table, then you must regrant object privileges on the table, re-create the indexes, integrity constraints, and triggers for the table, and respecify its storage parameters.

LAB EXERCISES:

Lab 1: Drop the teachers table from the school\_db database.

Ans.

ALTER TABLE students

DROP FOREIGN KEY fk\_teacher\_id;

DROP TABLE school\_db.teachers;

Lab 2: Drop the students table from the school\_db database and verify that the table has been removed.

Ans. DROP TABLE school\_db.students;

SHOW TABLES IN school\_db;

7. Data Manipulation Language (DML)

Theory Questions:

1. Define the INSERT, UPDATE, and DELETE commands in SQL.

Ans. 1. INSERT – Adds new records to a table.

Syntax: INSERT INTO table\_name (column1, column2, column3)

VALUES (value1, value2, value3);

2. UPDATE – Modifies existing records in a table.

Syntax: UPDATE table\_name

SET column1 = value1, column2 = value2

WHERE condition;

3. DELETE – Removes records from a table.

Syntax: DELETE FROM table\_name

WHERE condition;

2. What is the importance of the WHERE clause in UPDATE and DELETE operations?

Ans. The WHERE clause is a critical component of SQL UPDATE and DELETE operations. It specifies the conditions that must be met for the operation to be applied to a particular row or set of rows in a table.

Syntax:UPDATE employees SET salary = salary \* 1.1 WHERE department = 'Sales'; -- Only increases salary for employees in the Sales department

Syntax: DELETE FROM customers WHERE last\_purchase< '2022-01-01'; -- Only deletes customers who haven't made a purchase since 2022

LAB EXERCISES:

Lab 1: Insert three records into the courses table using the INSERT command.

Ans. INSERT INTO courses (course\_id, course\_name, course\_duration) VALUES

(1, "IS", 10),

(2, "DBMS", 15),

(3, "WT", 12);

Lab 2: Update the course duration of a specific course using the UPDATE command.

Ans. UPDATE courses

SET course\_duration = 20

WHERE course\_id = 1;

Lab 3: Delete a course with a specific course\_id from the courses table using the DELETE command.

Ans. DELETE FROM courses

WHERE course\_id = 1;

8. Data Query Language (DQL)

Theory Questions:

1. What is the SELECT statement, and how is it used to query data?

Ans. The SELECT statement in SQL is used to retrieve data from one or more tables in a database. It is the most commonly used SQL command for querying information.

Syntax: SELECT column1, column2, ...

FROM table\_name

WHERE condition;

2. Explain the use of the ORDER BY and WHERE clauses in SQL queries.

Ans. WHERE and ORDER BY Clauses in SQL Queries

Both WHERE and ORDER BY clauses are used to refine and organize query results in SQL.

1. WHERE Clause – Filtering Data

The WHERE clause is used to filter records by specifying a condition. Only rows that meet the condition are included in the result.

2. ORDER BY Clause – Sorting Data

The ORDER BY clause sorts query results based on one or more columns in ascending (ASC) or descending (DESC) order.

LAB EXERCISES:

Lab 1: Retrieve all courses from the courses table using the SELECT statement.

Ans. SELECT \* FROM courses;

Lab 2: Sort the courses based on course\_duration in descending order using ORDER BY.

Ans. SELECT \* FROM courses

ORDER BY course\_duration DESC;

Lab 3: Limit the results of the SELECT query to show only the top two courses using LIMIT.

Ans. SELECT \* FROM courses

ORDER BY course\_duration DESC

LIMIT 2;