**Assignment Questions**

**Part 1: Theoretical Questions**

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

SQL (Structured Query Language) is a language used to store, retrieve, and manage data in relational databases. It is important because it allows efficient data manipulation, ensures data integrity, and supports complex queries for decision-making.

1. Explain the difference between a database and a DBMS.

A **database** is a collection of organized data, while a **DBMS (Database Management System)** is software that manages and interacts with the database.

1. What are the key features of an RDBMS?

Tables,relations,data integrity,sql transactions

4. Describe the purpose of the following SQL commands:

• CREATE TABLE 🡺 Used to define the structure of a table.

• ALTER TABLE 🡺 Used to modify an existing table’s structure.

• DROP TABLE 🡺 Used to delete an entire table and its data.

1. Explain the significance of constraints in SQL. Provide examples of different types of constraints.

Constraints enforce rules on data in tables.

• Primary Key: Uniquely identifies each row.

PRIMARY KEY (StudentID)

• Foreign Key: Ensures relationships between tables.

FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)

• Unique: Ensures values in a column are unique.

UNIQUE (Email)

• Check: Restricts the values in a column.

CHECK (Age >= 18)

• Default: Assigns a default value.

DEFAULT 'Not Assigned'

• Not Null: Ensures a column cannot have NULL values.

NOT NULL

1. Compare DELETE and TRUNCATE commands in SQL.

DELETE removes specific rows, while TRUNCATE removes all rows from a table.

1. What is the difference between PRIMARY KEY and FOREIGN KEY?

**primary key** uniquely identifies records in a table, while a **foreign key** links a record to a primary key in another table

1. Explain the use of the WHERE clause in SQL with an example.

Where is used to filter the rows based on specific condition

Example :

create table Employees(

empid int primary key,

name varchar(100) not null,

age int check(age > 18),

dept varchar(50),

joindate date default (current\_date)

);

insert into Employees values(1,"alice",30,"HR","2023-01-15");

insert into Employees values(2,"bob",25,"IT","2022-05-20");

insert into Employees values(3,"charlie",28,"Finance","2021-10-10");

select \* from Employees where name="alice";

Output:

+-------+-------+------+------+------------+

| empid | name | age | dept | joindate |

+-------+-------+------+------+------------+

| 1 | alice | 30 | HR | 2023-01-15 |

+-------+-------+------+------+------------+

1. What are the different types of SQL data types? Provide examples.

• INT: Integer numbers.( age int)

• VARCHAR: Variable-length text.(name varchar(50))

• DATE: Stores dates. (joindate date)

• FLOAT: Decimal numbers. (price float(5,2))

10.How can the ORDER BY clause be used to sort data in descending order?

SELECT \* FROM Employees ORDER BY age DESC;

Data is sorted in descending order by using desc

**Part 2: Coding Questions**

1. Creating a Database and Table Write SQL commands to:

• Create a database named CompanyDB.

• Create a table named Employees with the following structure:

Column Name Data Type Constraint

EmployeeID INT PRIMARY KEY

Name VARCHAR(100) NOT NULL

Age INT CHECK (Age > 18)

Department VARCHAR(50)

JoiningDate DATE DEFAULT CURDATE()

**COMMANDS**

CREATE DATABASE CompanyDB;

USE CompanyDB;

create table Employees(

empid int primary key,

name varchar(100) not null,

age int check(age > 18),

dept varchar(50),

joindate date default (current\_date)

);

2. Inserting Data into a Table Insert the following data into the Employees table:

EmployeeID Name Age Department JoiningDate

1 Alice 30 HR 2023-01-15

2 Bob 25 IT 2022-05-20

3 Charlie 28 Finance 2021-10-10

**COMMANDS**

insert into Employees values(1,"alice",30,"HR","2023-01-15");

insert into Employees values(2,"bob",25,"IT","2022-05-20");

insert into Employees values(3,"charlie",28,"Finance","2021-10-10");

3. Updating Data Write a query to update the Department of Bob to Operations.

**COMMANDS**

update Employees set dept="operations" where name="bob";

4. Deleting Data Write a query to delete the record of an employee whose EmployeeID is 3.

**COMMANDS**

delete from Employees where empid=3;

5. Retrieving Data Write SQL queries to:

• Retrieve all employees who belong to the IT department.

• Retrieve employees sorted by their Name in ascending order.

• Retrieve employees who joined after 2022-01-01.

**COMMANDS**

select \* from Employees where dept="IT";

select name from Employees ;

select \* from Employees where joindate > "2022-01-01";

6. Using Constraints

• Add a new column Salary to the Employees table with a NOT NULL constraint.

• Update the Salary of all employees to 50000.

**COMMANDS**

alter table Employees add COLUMN sal int not null;

update Employees set sal=50000;

7. Filtering Data

• Retrieve all employees whose Age is between 25 and 30.

• Retrieve employees whose Name starts with the letter 'A'.

**COMMANDS**

select \* from Employees where age between 25 and 30;

select name from Employees where name like 'a%';

8. Advanced Query Write a query to find the average Age of employees in each department.

**COMMANDS**

select dept,avg(age) from Employees group by dept;

**Part 3: Challenge Task**

1. Creating Relationships Create a new table Departments with the following structure:

Column Name Data Type Constraint

DepartmentID INT PRIMARY KEY

DepartmentName VARCHAR(50) UNIQUE

Establish a FOREIGN KEY relationship between the Department column in the

Employees table and the DepartmentID in the Departments

**COMMANDS**

create table Departments (

departmentid int primary key,

departmentname varchar(50) unique

);

ALTER TABLE Employees

ADD FOREIGN KEY (departmentid) REFERENCES Departments(departmentid);

2. Complex Query Write a query to display all employees along with their department names

using a JOIN.

**COMMANDS**

SELECT Employees.empid, Employees.name,Employees.age, Departments.Departmentname FROM Employees INNER JOIN Departments ON Employees.Departmentid = Departments.Departmentid;

3. Aggregations Write a query to count the number of employees in each department.

**COMMANDS**

select count(empid) from Employees group by dept;

OUTPUT:

+--------------+

| count(empid) |

+--------------+

| 1 |

| 1 |

| 1 |

+--------------+