

## Module 4 – Introduction to DBMS

### Introduction to SQL

#### Theory Questions:

1. What is SQL, and why is it essential in database management?
2. Explain the difference between DBMS and RDBMS.
3. Describe the role of SQL in managing relational databases.
4. What are the key features of SQL?

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

### 2. SQL Syntax

#### Theory Questions:

1. What are the basic components of SQL syntax?
2. Write the general structure of an SQL `SELECT` statement.
3. Explain the role of clauses in SQL statements.

#### LAB EXERCISES:

- **Lab 1:** Write SQL queries to retrieve specific columns (`student_name` and `age`) from the `students` table.
  - **Lab 2:** Write SQL queries to retrieve all students whose age is greater than 10.
- 

### 3. SQL Constraints

#### Theory Questions:

1. What are constraints in SQL? List and explain the different types of constraints.
2. How do `PRIMARY KEY` and `FOREIGN KEY` constraints differ?
3. What is the role of `NOT NULL` and `UNIQUE` constraints?

## 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).
  - **Lab 2:** Implement a `FOREIGN KEY` constraint to relate the `teacher_id` from the `teachers` table with the `students` table.
- 

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

### Theory Questions:

1. Define the SQL Data Definition Language (DDL).
2. Explain the `CREATE` command and its syntax.
3. What is the purpose of specifying data types and constraints during table creation?

## 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.
  - **Lab 2:** Use the `CREATE` command to create a database `university_db`.
- 

## 5. ALTER Command

### Theory Questions:

1. What is the use of the `ALTER` command in SQL?
2. How can you add, modify, and drop columns from a table using `ALTER`?

## LAB EXERCISES:

- **Lab 1:** Modify the `courses` table by adding a column `course_duration` using the `ALTER` command.
  - **Lab 2:** Drop the `course_credits` column from the `courses` table.
- 

## 6. DROP Command

### Theory Questions:

1. What is the function of the `DROP` command in SQL?
2. What are the implications of dropping a table from a database?

## LAB EXERCISES:

- **Lab 1:** Drop the `teachers` table from the `school_db` database.
  - **Lab 2:** Drop the `students` table from the `school_db` database and verify that the table has been removed.
- 

## 7. Data Manipulation Language (DML)

### Theory Questions:

1. Define the `INSERT`, `UPDATE`, and `DELETE` commands in SQL.
2. What is the importance of the `WHERE` clause in `UPDATE` and `DELETE` operations?

## LAB EXERCISES:

- **Lab 1:** Insert three records into the `courses` table using the `INSERT` command.
  - **Lab 2:** Update the course duration of a specific course using the `UPDATE` command.
  - **Lab 3:** Delete a course with a specific `course_id` from the `courses` table using the `DELETE` command.
- 

## 8. Data Query Language (DQL)

### Theory Questions:

1. What is the `SELECT` statement, and how is it used to query data?
2. Explain the use of the `ORDER BY` and `WHERE` clauses in SQL queries.

## LAB EXERCISES:

- **Lab 1:** Retrieve all courses from the `courses` table using the `SELECT` statement.
  - **Lab 2:** Sort the courses based on `course_duration` in descending order using `ORDER BY`.
  - **Lab 3:** Limit the results of the `SELECT` query to show only the top two courses using `LIMIT`.
- 

## 9. Data Control Language (DCL)

### Theory Questions:

1. What is the purpose of `GRANT` and `REVOKE` in SQL?
2. How do you manage privileges using these commands?

## LAB EXERCISES:

- **Lab 1:** Create two new users `user1` and `user2` and grant `user1` permission to `SELECT` from the `courses` table.
  - **Lab 2:** Revoke the `INSERT` permission from `user1` and give it to `user2`.
- 

## 10. Transaction Control Language (TCL)

### Theory Questions:

1. What is the purpose of the `COMMIT` and `ROLLBACK` commands in SQL?
2. Explain how transactions are managed in SQL databases.

## LAB EXERCISES:

- **Lab 1:** Insert a few rows into the `courses` table and use `COMMIT` to save the changes.
  - **Lab 2:** Insert additional rows, then use `ROLLBACK` to undo the last insert operation.
  - **Lab 3:** Create a `SAVEPOINT` before updating the `courses` table, and use it to roll back specific changes.
- 

## 11. SQL Joins

### Theory Questions:

1. Explain the concept of `JOIN` in SQL. What is the difference between `INNER JOIN`, `LEFT JOIN`, `RIGHT JOIN`, and `FULL OUTER JOIN`?
2. How are joins used to combine data from multiple tables?

## LAB EXERCISES:

- **Lab 1:** Create two tables: `departments` and `employees`. Perform an `INNER JOIN` to display employees along with their respective departments.
  - **Lab 2:** Use a `LEFT JOIN` to show all departments, even those without employees.
- 

## 12. SQL Group By

### Theory Questions:

1. What is the `GROUP BY` clause in SQL? How is it used with aggregate functions?
2. Explain the difference between `GROUP BY` and `ORDER BY`.

## LAB EXERCISES:

- **Lab 1:** Group employees by department and count the number of employees in each department using `GROUP BY`.
  - **Lab 2:** Use the `AVG` aggregate function to find the average salary of employees in each department.
- 

## 13. SQL Stored Procedure

### Theory Questions:

1. What is a stored procedure in SQL, and how does it differ from a standard SQL query?
2. Explain the advantages of using stored procedures.

## LAB EXERCISES:

- **Lab 1:** Write a stored procedure to retrieve all employees from the `employees` table based on department.
  - **Lab 2:** Write a stored procedure that accepts `course_id` as input and returns the course details.
- 

## 14. SQL View

### Theory Questions:

1. What is a view in SQL, and how is it different from a table?
2. Explain the advantages of using views in SQL databases.

## LAB EXERCISES:

- **Lab 1:** Create a view to show all employees along with their department names.
  - **Lab 2:** Modify the view to exclude employees whose salaries are below \$50,000.
- 

## 15. SQL Triggers

### Theory Questions:

1. What is a trigger in SQL? Describe its types and when they are used.
2. Explain the difference between `INSERT`, `UPDATE`, and `DELETE` triggers.

## LAB EXERCISES:

- **Lab 1:** Create a trigger to automatically log changes to the `employees` table when a new employee is added.
  - **Lab 2:** Create a trigger to update the `last_modified` timestamp whenever an employee record is updated.
- 

## 16. Introduction to PL/SQL

### Theory Questions:

1. What is PL/SQL, and how does it extend SQL's capabilities?
2. List and explain the benefits of using PL/SQL.

## LAB EXERCISES:

- **Lab 1:** Write a PL/SQL block to print the total number of employees from the `employees` table.
  - **Lab 2:** Create a PL/SQL block that calculates the total sales from an `orders` table.
- 

## 17. PL/SQL Control Structures

### Theory Questions:

1. What are control structures in PL/SQL? Explain the `IF-THEN` and `LOOP` control structures.
2. How do control structures in PL/SQL help in writing complex queries?

## LAB EXERCISES:

- **Lab 1:** Write a PL/SQL block using an `IF-THEN` condition to check the department of an employee.
  - **Lab 2:** Use a `FOR LOOP` to iterate through employee records and display their names.
- 

## 18. SQL Cursors

### Theory Questions:

1. What is a cursor in PL/SQL? Explain the difference between implicit and explicit cursors.
2. When would you use an explicit cursor over an implicit one?

## LAB EXERCISES:

- **Lab 1:** Write a PL/SQL block using an explicit cursor to retrieve and display employee details.
  - **Lab 2:** Create a cursor to retrieve all courses and display them one by one.
- 

## 19. Rollback and Commit Savepoint

### Theory Questions:

1. Explain the concept of `SAVEPOINT` in transaction management. How do `ROLLBACK` and `COMMIT` interact with savepoints?
2. When is it useful to use savepoints in a database transaction?

## LAB EXERCISES:

- **Lab 1:** Perform a transaction where you create a savepoint, insert records, then rollback to the savepoint.
- **Lab 2:** Commit part of a transaction after using a savepoint and then rollback the remaining changes.

## EXTRA LAB PRACTISE FOR DATABASE CONCEPTS

### 1. Introduction to SQL

## LAB EXERCISES:

- **Lab 3:** Create a database called `library_db` and a table `books` with columns: `book_id`, `title`, `author`, `publisher`, `year_of_publication`, and `price`. Insert five records into the table.
  - **Lab 4:** Create a table `members` in `library_db` with columns: `member_id`, `member_name`, `date_of_membership`, and `email`. Insert five records into this table.
- 

### 2. SQL Syntax

## LAB EXERCISES:

- **Lab 3:** Retrieve all `members` who joined the library before 2022. Use appropriate SQL syntax with `WHERE` and `ORDER BY`.
  - **Lab 4:** Write SQL queries to display the titles of books published by a specific author. Sort the results by `year_of_publication` in descending order.
-

### 3. SQL Constraints

#### LAB EXERCISES:

- **Lab 3:** Add a `CHECK` constraint to ensure that the `price` of books in the `books` table is greater than 0.
  - **Lab 4:** Modify the `members` table to add a `UNIQUE` constraint on the `email` column, ensuring that each member has a unique email address.
- 

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

#### LAB EXERCISES:

- **Lab 3:** Create a table `authors` with the following columns: `author_id`, `first_name`, `last_name`, and `country`. Set `author_id` as the primary key.
  - **Lab 4:** Create a table `publishers` with columns: `publisher_id`, `publisher_name`, `contact_number`, and `address`. Set `publisher_id` as the primary key and `contact_number` as unique.
- 

### 5. ALTER Command

#### LAB EXERCISES:

- **Lab 3:** Add a new column `genre` to the `books` table. Update the `genre` for all existing records.
  - **Lab 4:** Modify the `members` table to increase the length of the `email` column to 100 characters.
- 

### 6. DROP Command

#### LAB EXERCISES:

- **Lab 3:** Drop the `publishers` table from the database after verifying its structure.
  - **Lab 4:** Create a backup of the `members` table and then drop the original `members` table.
-



## 7. Data Manipulation Language (DML)

### LAB EXERCISES:

- **Lab 4:** Insert three new authors into the `authors` table, then update the last name of one of the authors.
  - **Lab 5:** Delete a book from the `books` table where the `price` is higher than \$100.
- 

## 8. UPDATE Command

### LAB EXERCISES:

- **Lab 3:** Update the `year_of_publication` of a book with a specific `book_id`.
  - **Lab 4:** Increase the `price` of all books published before 2015 by 10%.
- 

## 9. DELETE Command

### LAB EXERCISES:

- **Lab 3:** Remove all members who joined before 2020 from the `members` table.
  - **Lab 4:** Delete all books that have a `NULL` value in the `author` column.
- 

## 10. Data Query Language (DQL)

### LAB EXERCISES:

- **Lab 4:** Write a query to retrieve all books with `price` between \$50 and \$100.
  - **Lab 5:** Retrieve the list of books sorted by `author` in ascending order and limit the results to the top 3 entries.
- 

## 11. Data Control Language (DCL)

### LAB EXERCISES:

- **Lab 3:** Grant `SELECT` permission to a user named `librarian` on the `books` table.
  - **Lab 4:** Grant `INSERT` and `UPDATE` permissions to the user `admin` on the `members` table.
-

## 12. REVOKE Command

### LAB EXERCISES:

- **Lab 3:** Revoke the `INSERT` privilege from the user `librarian` on the `books` table.
  - **Lab 4:** Revoke all permissions from user `admin` on the `members` table.
- 

## 13. Transaction Control Language (TCL)

### LAB EXERCISES:

- **Lab 3:** Use `COMMIT` after inserting multiple records into the `books` table, then make another insertion and perform a `ROLLBACK`.
  - **Lab 4:** Set a `SAVEPOINT` before making updates to the `members` table, perform some updates, and then roll back to the `SAVEPOINT`.
- 

## 14. SQL Joins

### LAB EXERCISES:

- **Lab 3:** Perform an `INNER JOIN` between `books` and `authors` tables to display the title of books and their respective authors' names.
  - **Lab 4:** Use a `FULL OUTER JOIN` to retrieve all records from the `books` and `authors` tables, including those with no matching entries in the other table.
- 

## 15. SQL Group By

### LAB EXERCISES:

- **Lab 3:** Group `books` by `genre` and display the total number of books in each genre.
  - **Lab 4:** Group `members` by the year they joined and find the number of members who joined each year.
- 

## 16. SQL Stored Procedure

### LAB EXERCISES:

- **Lab 3:** Write a stored procedure to retrieve all `books` by a particular `author`.
- **Lab 4:** Write a stored procedure that takes `book_id` as an argument and returns the price of the book.

---

## 17. SQL View

### LAB EXERCISES:

- **Lab 3:** Create a view to show only the `title`, `author`, and `price` of books from the `books` table.
- **Lab 4:** Create a view to display `members` who joined before 2020.

---

## 18. SQL Trigger

### LAB EXERCISES:

- **Lab 3:** Create a trigger to automatically update the `last_modified` timestamp of the `books` table whenever a record is updated.
- **Lab 4:** Create a trigger that inserts a log entry into a `log_changes` table whenever a `DELETE` operation is performed on the `books` table.

---

## 19. Introduction to PL/SQL

### LAB EXERCISES:

- **Lab 3:** Write a PL/SQL block to insert a new `book` into the `books` table and display a confirmation message.
- **Lab 4:** Write a PL/SQL block to display the total number of books in the `books` table.

---

## 20. PL/SQL Syntax

### LAB EXERCISES:

- **Lab 3:** Write a PL/SQL block to declare variables for `book_id` and `price`, assign values, and display the results.
  - **Lab 4:** Write a PL/SQL block using `constants` and perform arithmetic operations on book prices.
-

## 21. PL/SQL Control Structures

### LAB EXERCISES:

- **Lab 3:** Write a PL/SQL block using `IF-THEN-ELSE` to check if a book's price is above \$100 and print a message accordingly.
  - **Lab 4:** Use a `FOR LOOP` in PL/SQL to display the details of all books one by one.
- 

## 22. SQL Cursors

### LAB EXERCISES:

- **Lab 3:** Write a PL/SQL block using an explicit cursor to fetch and display all records from the `members` table.
  - **Lab 4:** Create a cursor to retrieve books by a particular author and display their titles.
- 

## 23. Rollback and Commit Savepoint

### LAB EXERCISES:

- **Lab 3:** Perform a transaction that includes inserting a new `member`, setting a `SAVEPOINT`, and rolling back to the savepoint after making updates.
- **Lab 4:** Use `COMMIT` after successfully inserting multiple books into the `books` table, then use `ROLLBACK` to undo a set of changes made after a savepoint.