**Introduction to 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.**

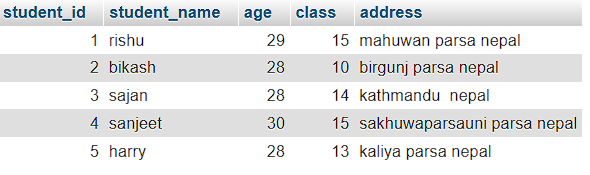
**Lab 2: Insert five records into the students table and retrieve all records using the SELECT statement.**

Command:

CREATE DATABASE school\_db;

CREATE TABLE students(student\_id int, student\_name varchar(30), age int , class int , address text);

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html)  \*FROM `student` ;

****

**2. SQL Syntax**

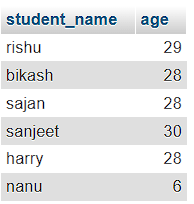
LAB EXERCISES:

**• 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**.

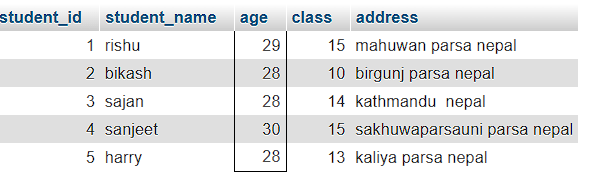
Command:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) student\_name,age FROM `student`;



Command:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html)  \*FROM `student` WHERE age>10;



**3. SQL 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.**

Command:

CREATE TABLE teachers

(

teacher\_id int (Primary Key),

teacher\_name varchar (NOT NULL),

subject varchar (NOT NULL),

email text (UNIQUE)

),



Command:

CREATE TABLE students(

student\_id int PRIMARY KEY,

student\_name varchar(30),

age int , class int ,

address text,

teacher\_id int,

FOREIGN KEY(teacher\_id) REFERENCES students(teacher\_id)

);



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

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.**

**Command:**

CREATE DATABASE university\_db;

CREATE TABLE courses(

course\_id int PRIMARY KEY,

course\_name varchar(30),

course\_credits int

);



**5. ALTER Command**

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.**

**Command:**

ALTER TABLE courses add course\_duration int;

****

**Command:**

ALTER TABLE courses DROP COLUMN course\_credits;

****

**6. DROP Command**

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.**

**Command:**

DROP TABLE students;

DROP TABLE TEACHERS;

**7. Data Manipulation Language (DML)**

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.**

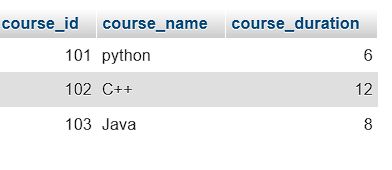
**Command:**

INSERT INTO courses VALUES(

101,'python','6 month'),

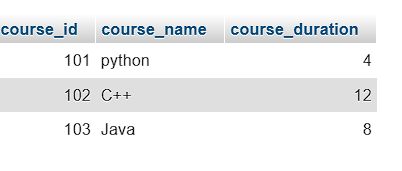
(102,'C++','12 month'),

(103,'Java','8 month');



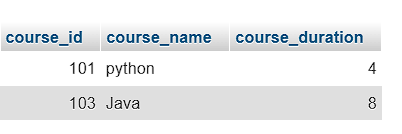
**Command:**

UPDATE courses SET course\_duration=4 WHERE course\_name=’python’;



**Command:**

DELETE FROM courses WHERE course\_id=102;



**8. Data Query Language (DQL)**

LAB EXERCISES:

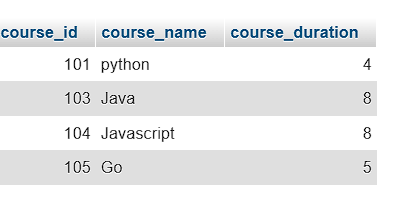
**• Lab 1: Retrieve all courses from the courses table using the SELECT statement.**

**• 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.**

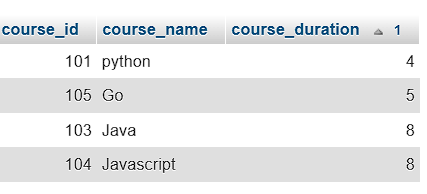
**Command:**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM courses;

****

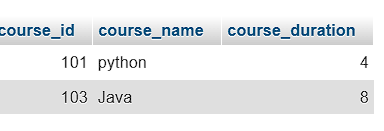
**Command:**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `courses` ORDER BY course\_duration;

****

**Command:**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `courses` LIMIT 2;

****

**11. SQL Joins**

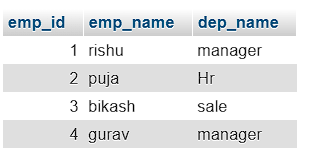
LAB EXERCISES:

**• 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.**

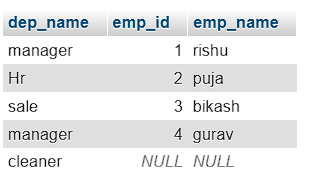
**Command:**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) emp\_id,emp\_name,dep\_name FROM employees INNER JOIN departments ON employees.dep\_id=departments.dep\_id;

****

**Command:**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html" \t "mysql_doc) dep\_name,emp\_id,emp\_name FROM departments [LEFT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_left) JOIN employees ON employees.dep\_id=departments.dep\_id;

****