Lab 2: Fetching, Filtering, Sorting Database (DBMS)

1. Write a SQL statement to create a simple table countries including columns country_id, country_name and region_id.
   → CREATETABLEcountries(
   COUNTRY_ID varchar(2),
   COUNTRY_NAME varchar(40),
   REGION_ID decimal(10,0));
2. Write a SQL statement to create a table named jobs including columns job_id, job_title, min_salary, max_salary and check whether the max_salary amount exceeding the upper limit 25000.
   → CREATE TABLE jobs (
   JOB_ID varchar(10)NOT NULL,
   JOB_TITLE varchar(35)NOT NULL,
   MIN_SALARY decimal(6,0),
   MAX_SALARY decimal(6,0)
   CHECK(MAX_SALARY<=25000));
3. Write a SQL statement to create a table named job_history including columns employee_id, start_date, end_date, job_id and department_id and make sure that the value against column end_date will be entered at the time of insertion to the format like '--/--/----
   → CREATE TABLE job_history(
   EMPLOYEE_ID decimal(6,0) NOT NULL,
   START_DATE date NOT NULL,
   END_DATE date NOT NULL,
   CHECK(END_DATE LIKE'--/--/----'),
   JOB_ID varchar(10) NOT NULL,
   DEPARTMENT_ID decimal(4,0) NOT NULL);
4. Write a SQL statement to insert a record with your own value into the table countries against each columns.
   → INSERT INTO countries VALUES('5','Nepal',1001);
5. Write a SQL statement to insert one row into the table countries against the column country_id and country_name.
   → INSERT INTO countries (country_id,country_name) VALUES ('C1','Nepal');
6. Write a SQL statement to insert 3 rows by a single insert statement.
   → INSERT INTO countries VALUES('C8','Vietnam',1001),
   ('C2','USA',1007),('C3','UK',1003);
7. Write a SQL statement to rename the table countries to country_new.
   → ALTER TABLE countries RENAME country_new; (MYSQL case)
   → EXEC sp_rename 'countries','country_table'; (SQL Server) → *(Stored Procedure)*

→ *SELECT SUM(salary) FROM employee;*

8. Write a query to get the average salary and number of employees in the employees table.
   → *SELECT AVG(salary), COUNT(*) FROM employee;*
9. Write a query to get the number of employees working with the company
   → *SELECT COUNT(*) FROM employee;*
10. Write a query to get the number of designations available in the employees table.
    → *SELECT COUNT(DISTINCT job_id) FROM employee;*
11. Write a query get all first name from employees table in upper case.
    → *SELECT UPPER(first_name) FROM employees;*
12. Write a query to get the first 3 characters of first name from employees table.
    → *SELECT SUBSTRING(first_name,1,3) FROM employees;*
13. Write a query to calculate 171*214+625
    → *SELECT 171*214 + 625 Result;*
14. Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range 10,000 through 15,000.
    → *SELECT first_name, last_name,salary, department_id FROM employee WHERE salary NOT BETWEEN 10000 AND 15000;*
15. Write a query to display the name (first_name, last_name) and hire date for all employees who were hired in 2016
    → *SELECT first_name, last_name, hire_date FROM employee WHERE YEAR (hire_date) LIKE '2016%';*
16. Write a query to display the first_name of all employees who have both "n" and "k" in their first name
    → *SELECT first_name FROM employee WHERE first_name LIKE '%n%' AND first_name LIKE '%k%';*
17. Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or analyst, and whose salary is not equal to 4,500, 10,000, or 15,000.
    → *SELECT last_name, job_id, salary FROM employee WHERE job_id IN ('IT_PROG', 'analyst') AND salary NOT IN (4500,10000, 15000);*
18. Write a query to display the last name of employees whose names have exactly 6 characters

➔ *SELECT last_name FROM employee WHERE last_name LIKE ' ';*

19. Write a query to display the last name of employees having 'k' as the third character

➔ *SELECT last_name FROM employee WHERE last_name LIKE '__d%';*

20. Write a query to display the jobs/designations available in the employees table.

➔ *SELECT DISTINCT job_id FROM employees;*

I. **Write SQL statement for the following queries in reference to relation Emp_time provided:**

Eid#	Name	Start_time	End_time
E101	Mangale	10:30	18:30
E102	Malati	8:30	14:30
E103	Fulmaya	9:00	18:00

i. Create the table and place Eid# as primary key and insert the values provided.

ii. Display the name of the employees whose name starts from letter 'M' and who work for more than seven hours.

iii. Delete the entire contents of the table so that new data can be insterted as (E107, Kamale, 10:00, 17:00).

II. **Create a employee table with following attributes (employee_id, first_name, last_name, email, phone_number, hire_date, department_id, job_id, salary)**

Note- Data type of hire_date must be in year(int), JOB_ID is like IT_PROG, analyst, lecturer, designer, db_admin. Atleast insert 10-20 data

1. Write a query to display the names (first_name, last_name)
   ➔ *SELECT first_name, last_name FROM employee;*
2. Write a query to get unique department ID from employee table
   ➔ *SELECT DISTINCT department_id FROM employee;*
3. Write a query to get all employee details from the employee table
   ➔ *SELECT * FROM employee;*
4. Write a query to get the names (first_name, last_name), salary, PF of all the employees (PF is calculated as 15% of salary)
   ➔ *SELECT first_name, last_name, salary, salary*.15 PF FROM employees;*
5. Count the number of unique hire dates
   ➔ *Select COUNT(distinct hire_date) from employee;*
6. Write a query to get the employee ID, names (first_name, last_name), salary in ascending order of salary.
   ➔ *SELECT employee_id, first_name, last_name, salary FROM employee ORDER BY salary;*
7. Write a query to get the total salaries payable to employees.