**Lab Exercise 1: Understanding NULL in MySQL**

**Problem:**

1. Create a table employees with columns id, name, age, and salary (where age and salary can be NULL).
2. Insert the following values:
   * (1, 'Alice', 30, 50000)
   * (2, 'Bob', NULL, 45000)
   * (3, 'Charlie', 28, NULL)
   * (4, 'David', NULL, NULL)
3. Select all records from the table.

**Solution:**

CREATE TABLE employees (

id INT PRIMARY KEY,

name VARCHAR(50),

age INT NULL,

salary DECIMAL(10,2) NULL

);

INSERT INTO employees (id, name, age, salary) VALUES

(1, 'Alice', 30, 50000),

(2, 'Bob', NULL, 45000),

(3, 'Charlie', 28, NULL),

(4, 'David', NULL, NULL);

SELECT \* FROM employees;

**Lab Exercise 2: Checking for NULL Values**

**Problem:**

Find all employees whose age is NULL.

**Solution:**

SELECT \* FROM employees WHERE age IS NULL;

**Lab Exercise 3: Handling NULL using IS NOT NULL**

**Problem:**

Retrieve all employees who have a non-null salary.

**Solution:**

SELECT \* FROM employees WHERE salary IS NOT NULL;

**Lab Exercise 4: Using COALESCE()**

**Problem:**

Display the employee name and salary, but if salary is NULL, replace it with 0.

**Solution:**

SELECT name, COALESCE(salary, 0) AS salary FROM employees;

**Lab Exercise 5: Using IFNULL()**

**Problem:**

Show employee names and their age. If age is NULL, display "Unknown".

**Solution:**

SELECT name, IFNULL(age, 'Unknown') AS age FROM employees;

**Lab Exercise 6: Counting NULL Values**

**Problem:**

Count how many employees have NULL values in the salary column.

**Solution:**

SELECT COUNT(\*) AS null\_salaries FROM employees WHERE salary IS NULL;

**Lab Exercise 7: Using NULL in Aggregates**

**Problem:**

Find the average salary of employees. Ensure that NULL salaries are ignored.

**Solution:**

SELECT AVG(salary) AS avg\_salary FROM employees;

**Lab Exercise 8: Using CASE to Handle NULL**

**Problem:**

Show all employees' names and categorize them as:

* 'Has Age' if age is not NULL
* 'No Age' if age is NULL

**Solution:**

SELECT name,

CASE

WHEN age IS NULL THEN 'No Age'

ELSE 'Has Age'

END AS age\_status

FROM employees;

**Lab Exercise 9: Replacing NULL Values using UPDATE**

**Problem:**

Update all NULL salaries in the table to 40000.

**Solution:**

UPDATE employees

SET salary = 40000

WHERE salary IS NULL;

**Lab Exercise 10: Filtering Data with NULL Values in Joins**

**Problem:**

1. Create a departments table with columns dept\_id and dept\_name.
2. Insert department values:
   * (1, 'HR')
   * (2, 'IT')
   * (3, 'Finance')
3. Add a dept\_id column in the employees table (allow NULL).
4. Insert department values for employees:
   * Alice -> 1
   * Bob -> NULL
   * Charlie -> 2
   * David -> NULL
5. Write a query to show all employees with department names. If no department is assigned, display "Not Assigned".

**Solution:**

CREATE TABLE departments (

dept\_id INT PRIMARY KEY,

dept\_name VARCHAR(50)

);

INSERT INTO departments (dept\_id, dept\_name) VALUES

(1, 'HR'),

(2, 'IT'),

(3, 'Finance');

ALTER TABLE employees ADD COLUMN dept\_id INT NULL;

UPDATE employees SET dept\_id = 1 WHERE name = 'Alice';

UPDATE employees SET dept\_id = NULL WHERE name = 'Bob';

UPDATE employees SET dept\_id = 2 WHERE name = 'Charlie';

UPDATE employees SET dept\_id = NULL WHERE name = 'David';

SELECT e.name, COALESCE(d.dept\_name, 'Not Assigned') AS department

FROM employees e

LEFT JOIN departments d ON e.dept\_id = d.dept\_id;