**DBMS & MySQL Home Work:**

1. Practice (%, \_, [], ^,-, {}) with 10 example (like, wildcard )

|  |  |  |
| --- | --- | --- |
| % | SELECT \* FROM studentreport WHERE FirstName LIKE 'A%'; |  |
| \_ | SELECT \* FROM studentreport WHERE Gender LIKE '\_\_\_e'; |  |
| ^,[] | SELECT \* FROM studentreport WHERE FirstName REGEXP '^[ABCabc]'; |  |
| [],% | SELECT \* FROM Customers WHERE CustomerName like '[A-B]%'; |  |
| % | SELECT \* FROM Customers WHERE ContactName LIKE 'A%'; |  |
| \_ | SELECT \* FROM Customers WHERE Country LIKE 'Me\_\_\_o'; |  |
| {} | SELECT \* FROM Customers WHERE CustomerName LIKE '[A-Z{1,2,3}]%'; |  |
| % | SELECT \* FROM Customers WHERE CustomerName LIKE '%a'; |  |
| % | SELECT \* FROM studentreport WHERE Grade LIKE '%+'; |  |
| {},% | SELECT \* FROM Customers WHERE CustomerName LIKE '[l-o{1,2,3}]%'; |  |

2. Create a worker table with first\_name, last\_name, and salary, and write a query to show a list of workers whose salary is greater than the average salary. (Using in operator)

|  |  |
| --- | --- |
| CREATE TABLE worker (  id INT AUTO\_INCREMENT PRIMARY KEY,  first\_name VARCHAR(255) NOT NULL,  last\_name VARCHAR(255) NOT NULL,  salary DECIMAL(10, 2) NOT NULL  ); | SELECT \*  FROM worker  WHERE salary > (  SELECT AVG(salary)  FROM worker  ); |

3. write a query to show a list of workers whose first\_name starts with a vowel and last\_name ends with a vowel. (add 5 worker data whose first name starts with a vowel and ends with a vowel).

|  |  |
| --- | --- |
| INSERT INTO worker (first\_name, last\_name, salary) VALUES  ('Alice', 'Jone', 60000),  ('Emma', 'Tao', 55000),  ('Olivia', 'Doe', 62000),  ('Isabella', 'Tauli’, 59000),  (‘Ava', 'Anderso', 58000); | SELECT \*  FROM worker  WHERE first\_name REGEXP '^[AEIOUaeiou]' AND  last\_name REGEXP '[AEIOUaeiou]$'; |

4. write a query to show a list of workers whose salary is greater than the lowest salary and less than the highest salary. (Using between operations):

|  |  |
| --- | --- |
| SELECT \*  FROM worker  WHERE salary BETWEEN (  SELECT MIN(salary) FROM worker  ) AND (  SELECT MAX(salary) FROM worker  ); |  |

5. Create a salary history table with salary, worker\_id, and paid unpaid status and show the total paid amount until today of every worker with the worker's name.

|  |  |
| --- | --- |
| CREATE TABLE salary\_history (  id INT AUTO\_INCREMENT PRIMARY KEY,  salary DECIMAL(10, 2) NOT NULL,  worker\_id INT NOT NULL,  paid\_unpaid\_status VARCHAR(10) NOT NULL  ); | SELECT w.first\_name, w.last\_name, SUM(sh.salary) AS total\_paid\_amount  FROM worker w  JOIN salary\_history sh ON w.id = sh.worker\_id  WHERE sh.paid\_unpaid\_status = 'paid'  GROUP BY w.id; |

6. Create a menu table with name, parent\_id (parent id is menu table id) (show all menu and parent menu names using self join)

|  |  |
| --- | --- |
| CREATE TABLE menu (  id INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(255) NOT NULL,  parent\_id INT,  FOREIGN KEY (parent\_id) REFERENCES menu(id)  ); | SELECT m1.name AS menu\_name, m2.name AS parent\_menu\_name  FROM menu m1  LEFT JOIN menu m2 ON m1.parent\_id = m2.id; |