LAB 1

Q.N.1. Create database as per your name.

Solution:

Query: create database kyurisha_karki;

Result:

```
mysql> create database Kyurisha_karki;
Query OK, 1 row affected (0.05 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| kyurisha           |
| kyurisha_karki     |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
8 rows in set (0.01 sec)
```

Q.N.2. Create Table Student with primary key sid.

Solution:

Query: create table Student(

-> sid int primary key,

-> name varchar(255),

-> age int

->);

Result:

```
mysql> use Kyurisha_karki
Database changed
mysql> create table students(
    -> sid int primary key,
    -> name varchar(255),
    -> age int
    -> );
Query OK, 0 rows affected (0.14 sec)
```

Q.N.3. Create table Course with foreign key sid.

Solution:

Query: create table courses(

-> cid int primary key,

-> course_name varchar(50),

-> sid int,

-> foreign key (sid) references students(sid)

->);

Result:

```
mysql> create table courses(
    -> cid int primary key,
    -> course_name varchar(50),
    -> sid int,
    -> foreign key (sid) references students(sid)
    -> );
Query OK, 0 rows affected (0.12 sec)
```

Q.N.4. Describe structure of table student.

Solution:

Query: describe students;

Result:

```
mysql> use Kyurisha_karki
Database changed
mysql> describe students;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| sid   | int          | NO   | PRI | NULL    |       |
| name  | varchar(255) | YES  |     | NULL    |       |
| age   | int          | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
3 rows in set (0.03 sec)
```

Q.N.5. Rename table employee with new name employee_details.

Solution:

Query: alter table employee rename to employee_details.;

Result:

```
mysql> use employee;
Database changed
mysql> alter table employee rename to employee_detail;
Query OK, 0 rows affected (0.08 sec)
```

LAB 2

Q.N.1 Write a query to display all employee who are from Bhaktapur.

Solution:

Query: select * from employee where address like ‘Bhaktapur’

Result:

```
mysql> select * from employee where address like 'Bhaktapur';
+-----+---------+------+-----------+-----------+
| eid | e_name  | age  | address   | salary    |
+-----+---------+------+-----------+-----------+
| 101 | Jharana |   20 | Bhaktapur | 100000.00 |
| 103 | Nikita  |   20 | Bhaktapur |  80000.00 |
| 105 | Sunita  |   20 | Bhaktapur |  85000.00 |
+-----+---------+------+-----------+-----------+
3 rows in set (0.01 sec)
```

Q.N.2 Create table employee with minimum 5 attributes and insert minimum 10 records.

Solution:

```
Query: create table employees(
   -> eid int primary key,
   -> e_name varchar(100),
   -> age int,
   -> address varchar(100),
   -> salary decimal(10,2)
   -> );
insert into employees(eid,e_name,age,address,salary)values
   -> (1,'Kyurisha Karki',20,'Kathmandu',30000),
   -> (2,'Jharana Oli',19,'Bhaktapur',40000),
   -> (3,'Sunita Rai',17,'Kathmandu',20000),
   -> (4,'Nikita Bhujel',18,'Lalitpur',45000),
   -> (5,'Abantika Lama',20,'Kathmandu',40000),
   -> (6,'Kusum Dangol',20,'Lalitpur',50000),
   -> (7,'Neelam Nakarmi',19,'Lalitpur',50000),
   -> (8,'Pemba Gole',19,'Bhaktapur',40000),
   -> (9,'Orisha Shakya',20,'Lalitpur',45000),
   -> (10,'Kriyasha Karki',17,'Kathmandu',42000);
```

Result:

```
mysql> select * from employees;
+-----+----------------+------+-----------+----------+
| eid | e_name         | age  | address   | salary   |
+-----+----------------+------+-----------+----------+
|   1 | Kyurisha Karki |   20 | Kathmandu | 30000.00 |
|   2 | Jharana Oli    |   19 | Bhaktapur | 40000.00 |
|   3 | Sunita Rai     |   17 | Kathmandu | 20000.00 |
|   4 | Nikita Bhujel  |   18 | Lalitpur  | 45000.00 |
|   5 | Abantika Lama  |   20 | Kathmandu | 40000.00 |
|   6 | Kusum Dangol   |   20 | Lalitpur  | 50000.00 |
|   7 | Neelam Nakarmi |   19 | Lalitpur  | 50000.00 |
|   8 | Pemba Gole     |   19 | Bhaktapur | 40000.00 |
|   9 | Orisha Shakya  |   20 | Lalitpur  | 45000.00 |
|  10 | Kriyasha Karki |   17 | Kathmandu | 42000.00 |
+-----+----------------+------+-----------+----------+
10 rows in set (0.00 sec)
```

Q.N.3. Update table employee set new address Bhaktapur whose id is 1.

Solution:

Query: update employees set address= ‘Bhaktapur’ where eid=1;

Result:

```
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employees where eid like 1;
+-----+----------------+------+-----------+----------+
| eid | e_name         | age  | address   | salary   |
+-----+----------------+------+-----------+----------+
|   1 | Kyurisha Karki |   20 | Bhaktapur | 30000.00 |
+-----+----------------+------+-----------+----------+
1 row in set (0.03 sec)
```

Q.N.4. Create table department with eid as a foreign key.

Solution:

```
Query: create table department(
   -> dept_id int primary key,
   -> dept_name varchar(100),
   -> eid int,
   -> foreign key (eid) references employees(eid)
   -> );
```

Result:

```
mysql> create table department(
    -> dept_id int primary key,
    -> dept_name varchar(100),
    -> eid int,
    -> foreign key (eid) references employees(eid)
    -> );
Query OK, 0 rows affected (0.14 sec)

mysql> desc department;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| dept_id   | int         | NO   | PRI | NULL    |       |
| dept_name | varchar(100) | YES  |     | NULL    |       |
| eid       | int         | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.06 sec)
```

Q.N.5. Write a query which will increase the salary of each department by Rs 1000.
Solution:

Query: update employee set salary = salary +1000;

Result:

```
mysql> update employees set salary=salary+1000;
Query OK, 10 rows affected (0.07 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from employees;
+-----+----------------+------+-----------+----------+
| eid | e_name         | age  | address   | salary   |
+-----+----------------+------+-----------+----------+
|   1 | Kyurisha Karki |   20 | Bhaktapur | 31000.00 |
|   2 | Jharana Oli    |   19 | Bhaktapur | 41000.00 |
|   3 | Sunita Rai     |   17 | Kathmandu | 21000.00 |
|   4 | Nikita Bhujel  |   18 | Lalitpur  | 46000.00 |
|   5 | Abantika Lama  |   20 | Kathmandu | 41000.00 |
|   6 | Kusum Dangol   |   20 | Lalitpur  | 51000.00 |
|   7 | Neelam Nakarmi |   19 | Lalitpur  | 51000.00 |
|   8 | Pemba Gole     |   19 | Bhaktapur | 41000.00 |
|   9 | Orisha Shakya  |   20 | Lalitpur  | 46000.00 |
|  10 | Kriyasha Karki |   17 | Kathmandu | 43000.00 |
+-----+----------------+------+-----------+----------+
10 rows in set (0.00 sec)
```

LAB 3

Q.N.1. Write a query to increase salary of employees by 10% whose salary is more than 5000.

Solution:

Query: update employee set salary = salary *1.10 where salary > 5000;

Result:

```
mysql> update employees set salary=salary*1.10 where salary>5000;
Query OK, 10 rows affected (0.03 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from employees;
+-----+----------------+------+-----------+----------+
| eid | e_name         | age  | address   | salary   |
+-----+----------------+------+-----------+----------+
|   1 | Kyurisha Karki |   20 | Bhaktapur | 34100.00 |
|   2 | Jharana Oli    |   19 | Bhaktapur | 45100.00 |
|   3 | Sunita Rai     |   17 | Kathmandu | 23100.00 |
|   4 | Nikita Bhujel  |   18 | Lalitpur  | 50600.00 |
|   5 | Abantika Lama  |   20 | Kathmandu | 45100.00 |
|   6 | Kusum Dangol   |   20 | Lalitpur  | 56100.00 |
|   7 | Neelam Nakarmi |   19 | Lalitpur  | 56100.00 |
|   8 | Pemba Gole     |   19 | Bhaktapur | 45100.00 |
|   9 | Orisha Shakya  |   20 | Lalitpur  | 50600.00 |
|  10 | Kriyasha Karki |   17 | Kathmandu | 47300.00 |
+-----+----------------+------+-----------+----------+
10 rows in set (0.00 sec)
```

Q.N.2. Write a query to change the department id of employees whose old department id is 201. The new department id should be 501.

Solution:

Query: update departments set dept_id = 501 where dept_id = 201;

Result:

```
mysql> update department set dept_id = 501 where dept_id = 201;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from department;
+---------+------------+------+
| dept_id | dept_name  | eid  |
+---------+------------+------+
|     202 | Psychology |    2 |
|     501 | IT         |    1 |
+---------+------------+------+
2 rows in set (0.00 sec)
```

Q.N.3. Write a query to select name, address and salary of all employees who are from Kathmandu, Lalitpur and Bhaktapur.

Solution:

Query: select name,address,salary from employee

-> where address in('Kathmandu','Lalitpur','Bhaktapur');

Result:

```
mysql> select e_name,address,salary from employees
    -> where address in('Kathmandu','Lalitpur','Bhaktapur');
+----------------+-----------+----------+
| e_name         | address   | salary   |
+----------------+-----------+----------+
| Kyurisha Karki | Bhaktapur | 34100.00 |
| Jharana Oli    | Bhaktapur | 45100.00 |
| Sunita Rai     | Kathmandu | 23100.00 |
| Nikita Bhujel  | Lalitpur  | 50600.00 |
| Abantika Lama  | Kathmandu | 45100.00 |
| Kusum Dangol   | Lalitpur  | 56100.00 |
| Neelam Nakarmi | Lalitpur  | 56100.00 |
| Pemba Gole     | Bhaktapur | 45100.00 |
| Orisha Shakya  | Lalitpur  | 50600.00 |
| Kriyasha Karki | Kathmandu | 47300.00 |
+----------------+-----------+----------+
10 rows in set (0.04 sec)
```

Q.N.4. Write a query to select name, department name and print employee details who are working in IT department.

Solution:

Query: select employee.e_name, departments.dept_name

-> from employee join departments

-> on employee.id = departments.eid

-> where departments.dept_name = 'IT';

Result:

```
mysql> select employees.e_name,department.dept_name
    -> from employees join department
    -> on employees.eid=department.eid
    -> where department.dept_name='IT';
+----------------+-----------+
| e_name         | dept_name |
+----------------+-----------+
| Kyurisha Karki | IT        |
+----------------+-----------+
1 row in set (0.04 sec)
```

Q.N.5. Create table exam marks and subject should be unique.

Solution:

Query: create table exam_marks (

-> id int auto_increment primary key,

-> subject varchar(100) UNIQUE,

-> marks int

->);

Result:

```
mysql> desc exam_marks;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int          | NO   | PRI | NULL    | auto_increment |
| subject | varchar(100) | YES  | UNI | NULL    |                |
| marks   | int          | YES  |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
3 rows in set (0.04 sec)
```

Q.N.6. Create table customer and Orders and orders table should have foreign key.

Solution:

Query:

Create table customer (

-> cid int auto increment primary key,

-> name varchar(50),

-> address varchar(100)

->);

Create table orders (

-> oi dint primary key auto_increment,

-> o_time time,

-> cid int,

-> freign key(cid) references customer(cid)

->);

Result:

```
mysql> desc orders;
+--------+------+------+-----+---------+----------------+
| Field  | Type | Null | Key | Default | Extra          |
+--------+------+------+-----+---------+----------------+
| oid    | int  | NO   | PRI | NULL    | auto_increment |
| o_time | time | YES  |     | NULL    |                |
| cid    | int  | YES  | MUL | NULL    |                |
+--------+------+------+-----+---------+----------------+
3 rows in set (0.02 sec)
```

Q.N.7. Create table atm where atmid should be greater than 100.

Solution:

Query: create table atm(

-> atmid int primary key check (atmid > 100),

-> location varchar(50)

-> bankname varchar (20) DEFAULT ‘Kumari’

->);

Result:

```
mysql> desc atm;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| atmid    | int         | NO   | PRI | NULL    |       |
| location | varchar(50) | YES  |     | NULL    |       |
| bankname | varchar(20) | YES  |     | Kumari  |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.04 sec)
```

Q.N.8. Create table bank where default balance should be 1000.

Solution:

Query: create table bank (

-> acc_id int primary key auto_increment,

-> acc_holder varchar(100),

-> acc_balance decimal(10, 2) default 1000

->);

Result:

```
mysql> desc bank;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| acc_id      | int          | NO   | PRI | NULL    | auto_increment |
| acc_holder  | varchar(100) | YES  |     | NULL    |                |
| acc_balance | decimal(10,2)| YES  |     | 1000.00 |                |
+-------------+--------------+------+-----+---------+----------------+
3 rows in set (0.04 sec)
```

LAB 4

Q.N.1. Write a SQL query to retrieve empno, ename, job,salary of all employees in descending order of their salary.

Solution:

Query: select empno,ename,job,salary from employee order by salary desc;

Result:

```
mysql> select empno,ename,job,salary from employee order by salary desc;
+-------+----------+------------------+-----------+
| empno | ename    | job              | salary    |
+-------+----------+------------------+-----------+
|    11 | Jharana  | Web Developer    | 100000.00 |
|    13 | Sunita   | DBA              |  85000.00 |
|    12 | Nikita   | Data Analyst     |  80000.00 |
|    14 | Kyurisha | Graphic Designer |  80000.00 |
+-------+----------+------------------+-----------+
4 rows in set (0.00 sec)
```

Q.N.2. Write a SQL query to retrieve all information of employee that belongs to department number 10 or 20.

Solution:

Query: select employees.* from employees join department

-> on employees.eid = department.eid

-> where dept_no in (10,20);

Result:

```
mysql> select employees.* from employees join department
    -> on employees.eid = department.eid
    -> where dept_no in (10,20);
+-----+----------------+------+-----------+----------+
| eid | e_name         | age  | address   | salary   |
+-----+----------------+------+-----------+----------+
|   2 | Jharana Oli    |   19 | Bhaktapur | 45100.00 |
|   1 | Kyurisha Karki |   20 | Bhaktapur | 34100.00 |
+-----+----------------+------+-----------+----------+
2 rows in set (0.00 sec)
```

Q.N.3. Write a query in SQL to list the employee who does not belongs department no 10.

Solution:

Query: SELECT employees.* FROM employees

-> JOIN department ON employees.eid = department.eid

-> where department.dept_no <> 10;

Result:

```
mysql>  SELECT employees.* FROM employees
    -> JOIN department ON employees.eid = department.eid
    ->  where department.dept_no <> 10;
+-----+----------------+------+-----------+----------+
| eid | e_name         | age  | address   | salary   |
+-----+----------------+------+-----------+----------+
|   1 | Kyurisha Karki |   20 | Bhaktapur | 34100.00 |
|   3 | Sunita Rai     |   17 | Kathmandu | 23100.00 |
+-----+----------------+------+-----------+----------+
2 rows in set (0.00 sec)
```

Q.N.4. Write a SQL query to retrieve employee information whose salary is greater than average salary of all employee.

Solution:

Query: select * from employees where salary > (select avg(salary) from employees);

Result:

```
mysql>  select * from employees where salary > (select avg(salary) from employees);
+-----+----------------+------+-----------+----------+
| eid | e_name         | age  | address   | salary   |
+-----+----------------+------+-----------+----------+
|   4 | Nikita Bhujel  |   18 | Lalitpur  | 50600.00 |
|   6 | Kusum Dangol   |   20 | Lalitpur  | 56100.00 |
|   7 | Neelam Nakarmi |   19 | Lalitpur  | 56100.00 |
|   9 | Orisha Shakya  |   20 | Lalitpur  | 50600.00 |
|  10 | Kriyasha Karki |   17 | Kathmandu | 47300.00 |
+-----+----------------+------+-----------+----------+
5 rows in set (0.00 sec)
```

Q.N.5. Write a SQL query to display all employee who does not have any commission.

Solution:

Query: select * from employees where commission is null;

Result:

```
mysql>  select * from employees where commission is null;
+-----+----------------+------+-----------+----------+------------+
| eid | e_name         | age  | address   | salary   | commission |
+-----+----------------+------+-----------+----------+------------+
|   1 | Kyurisha Karki |   20 | Bhaktapur | 34100.00 |       NULL |
|   2 | Jharana Oli    |   19 | Bhaktapur | 45100.00 |       NULL |
|   3 | Sunita Rai     |   17 | Kathmandu | 23100.00 |       NULL |
|   4 | Nikita Bhujel  |   18 | Lalitpur  | 50600.00 |       NULL |
|   5 | Abantika Lama  |   20 | Kathmandu | 45100.00 |       NULL |
|   6 | Kusum Dangol   |   20 | Lalitpur  | 56100.00 |       NULL |
|   7 | Neelam Nakarmi |   19 | Lalitpur  | 56100.00 |       NULL |
|   8 | Pemba Gole     |   19 | Bhaktapur | 45100.00 |       NULL |
|   9 | Orisha Shakya  |   20 | Lalitpur  | 50600.00 |       NULL |
|  10 | Kriyasha Karki |   17 | Kathmandu | 47300.00 |       NULL |
+-----+----------------+------+-----------+----------+------------+
10 rows in set (0.00 sec)
```

Q.N.6. Display empname and annual total salary of individual's employee from employee table.

Solution:

Query: SELECT e_name, salary * 12 AS annual_salary FROM employees;

Result:

```
mysql> SELECT e_name, salary * 12 AS annual_salary FROM employees;
+---------------+---------------+
| e_name        | annual_salary |
+---------------+---------------+
| Kyurisha Karki |     409200.00 |
| Jharana Oli    |     541200.00 |
| Sunita Rai     |     277200.00 |
| Nikita Bhujel  |     607200.00 |
| Abantika Lama  |     541200.00 |
| Kusum Dangol   |     673200.00 |
| Neelam Nakarmi |     673200.00 |
| Pemba Gole     |     541200.00 |
| Orisha Shakya  |     607200.00 |
| Kriyasha Karki |     567600.00 |
+---------------+---------------+
10 rows in set (0.00 sec)
```

Q.N.8. Write a SQL query to display information of employee whose name starts with A

Solution:

Query: select * from employee where e_name like 'A%';

Result:

```
mysql> select * from employees where e_name like 'A%';
+-----+---------------+------+-----------+----------+------------+
| eid | e_name        | age  | address   | salary   | commission |
+-----+---------------+------+-----------+----------+------------+
|   5 | Abantika Lama |   20 | Kathmandu | 45100.00 |       NULL |
+-----+---------------+------+-----------+----------+------------+
1 row in set (0.02 sec)
```

Q.N.9. Write a SQL query to find out total number of department in the given employee table.

Solution:

Query: SELECT COUNT(DISTINCT dept_no) AS total_department

-> FROM employees join department

-> on employees.eid = department.eid;

Result:

```
+------------------+
| total_department |
+------------------+
|                3 |
+------------------+
1 row in set (0.01 sec)
```

Q.N.10. Display all the information of employee whose salary is between 3000 AND 5000.

Solution:

Query: SELECT * FROM employee WHERE salary BETWEEN 3000 AND 5000;

Result:

```
mysql> SELECT * FROM employees WHERE salary BETWEEN 3000 AND 5000;
Empty set (0.00 sec)
```

Q.N.11. Write a SQL query to display information of employee whose name ends with A.

Solution:

Query: select * from employee where name like '%A';

Result:

```
mysql> select * from employees where e_name like '%A';
+-----+--------------+------+-----------+----------+------------+
| eid | e_name       | age  | address   | salary   | commission |
+-----+--------------+------+-----------+----------+------------+
|   5 | Abantika Lama |   20 | Kathmandu | 45100.00 |       NULL |
|   9 | Orisha Shakya |   20 | Lalitpur  | 50600.00 |       NULL |
+-----+--------------+------+-----------+----------+------------+
2 rows in set (0.00 sec)
```

Q.N.12. Write a SQL query to display information of employee whose name starts with A and ends with N.

Solution:

Query: select * from employee where name like 'A%N';

Result:

```
mysql> select * from employees where e_name like 'A%N';
Empty set (0.00 sec)
```

Q.N.13. Display all the information of employee whose salary is not between 3000 AND 5000.

Solution:

Query: SELECT * FROM employee WHERE salary NOT BETWEEN 3000 AND 5000;

Result:

```
mysql> SELECT * FROM employees WHERE salary NOT BETWEEN 3000 AND 5000;
+-----+----------------+------+-----------+----------+------------+
| eid | e_name         | age  | address   | salary   | commission |
+-----+----------------+------+-----------+----------+------------+
|   1 | Kyurisha Karki |   20 | Bhaktapur | 34100.00 |       NULL |
|   2 | Jharana Oli    |   19 | Bhaktapur | 45100.00 |       NULL |
|   3 | Sunita Rai     |   17 | Kathmandu | 23100.00 |       NULL |
|   4 | Nikita Bhujel  |   18 | Lalitpur  | 50600.00 |       NULL |
|   5 | Abantika Lama  |   20 | Kathmandu | 45100.00 |       NULL |
|   6 | Kusum Dangol   |   20 | Lalitpur  | 56100.00 |       NULL |
|   7 | Neelam Nakarmi |   19 | Lalitpur  | 56100.00 |       NULL |
|   8 | Pemba Gole     |   19 | Bhaktapur | 45100.00 |       NULL |
|   9 | Orisha Shakya  |   20 | Lalitpur  | 50600.00 |       NULL |
|  10 | Kriyasha Karki |   17 | Kathmandu | 47300.00 |       NULL |
+-----+----------------+------+-----------+----------+------------+
10 rows in set (0.00 sec)
```

Q.N.14. Display all the information of employee whose salary starts with 10.

Solution:

Query: SELECT * FROM employees WHERE salary like '10%';

Result:

```
mysql> SELECT * FROM employees WHERE salary like '10%';
Empty set (0.00 sec)
```

Q.N.15. Display all the information of employee whose salary exactly have 6 digit.

Solution:

Query: SELECT * FROM employee WHERE salary BETWEEN 100000 AND 999999;

Result:

```
mysql> SELECT * FROM employees WHERE salary BETWEEN 100000 AND 999999;
Empty set (0.00 sec)
```

Q.N.16. Write a query to display employee in which city name starts with ‘ka’ ends with ‘ti’ and contains multiple character between ‘ka’ and ‘ti’.

Solution:

Query: select * from employees where address like 'ka%ti';

Result:

```
mysql> select * from employees where address like 'ka%ti';
Empty set (0.00 sec)
```

LAB 5

Q.N.1. Display different job levels of employees in employee table.

Solution:

Query: select distinct job from employees;

Result:

```
mysql> select distinct job from employee;
+------------------+
| job              |
+------------------+
| Web Developer    |
| Data Analyst     |
| DBA              |
| Graphic Designer |
+------------------+
4 rows in set (0.04 sec)
```

Q.N.2. Display empname, job, annual salary of employee using order by clause.

Solution:

Query: SELECT name, job, salary * 12 AS annual_salary FROM employees

->ORDER BY annual_salary;

Result:

```
mysql> SELECT ename, job, salary * 12 AS annual_salary FROM employee
    ->  ORDER BY annual_salary;
+----------+------------------+---------------+
| ename    | job              | annual_salary |
+----------+------------------+---------------+
| Nikita   | Data Analyst     |     960000.00 |
| Kyurisha | Graphic Designer |     960000.00 |
| Sunita   | DBA              |    1020000.00 |
| Jharana  | Web Developer    |    1200000.00 |
+----------+------------------+---------------+
4 rows in set (0.01 sec)
```

Q.N.3. Display the maximum salary of employee using order by salary.

Solution:

Query: select max(salary) as max_salary from employees;

Result:

```
mysql> select max(salary) as max_salary from employees;
+------------+
| max_salary |
+------------+
|   56100.00 |
+------------+
1 row in set (0.03 sec)
```

Q.N.4. Print the minimum salary of employee.

Solution:

Query: SELECT MIN(salary) AS min_salary FROM employees;

Result:

```
mysql> SELECT MIN(salary) AS min_salary FROM employees;
+------------+
| min_salary |
+------------+
|   23100.00 |
+------------+
1 row in set (0.00 sec)
```

Q.N.5. Print the total salary of all employee.

Solution:

Query: SELECT SUM(salary) AS total_salary FROM employees;

Result:

```
mysql> SELECT SUM(salary) AS total_salary FROM employees;
+--------------+
| total_salary |
+--------------+
|    453200.00 |
+--------------+
1 row in set (0.00 sec)
```

Q.N.6. Display the average salary of employee.

Solution:

Query: SELECT avg(salary) AS avg_salary from employees;

Result:

```
mysql> SELECT avg(salary) AS avg_salary from employees;
+--------------+
| avg_salary   |
+--------------+
| 45320.000000 |
+--------------+
1 row in set (0.02 sec)
```

Q.N.7. Display branch and total marks of individual department from student table using group by.

Solution:

Query: SELECT branch, SUM(total_marks) AS total_marks FROM students

-> GROUP BY branch;

Result:

```
mysql>  SELECT branch, SUM(total_marks) AS total_marks FROM students
    ->  GROUP BY branch;
+---------+-------------+
| branch  | total_marks |
+---------+-------------+
| IT      |         150 |
| Arts    |         185 |
| Finance |         170 |
+---------+-------------+
3 rows in set (0.05 sec)
```

Q.N.8. Display branch and total marks of IT branch only from student table.

Solution:

Query: SELECT branch, SUM(marks) AS total_marks

-> FROM student

-> WHERE branch = 'IT'

-> group by branch;

Result:

```
mysql>  SELECT branch, SUM(total_marks) AS total_marks
    ->  FROM students
    ->  WHERE branch = 'IT'
    ->  GROUP BY branch;
+--------+-------------+
| branch | total_marks |
+--------+-------------+
| IT     |         150 |
+--------+-------------+
1 row in set (0.03 sec)
```

Q.N.9. Create view name as viewIT on the basis of IT branch.

Solution:

Query: create view viewIT as select * from student where branch = 'IT';

Result:

```
mysql> select * from viewIT;
+-----+----------+--------+-------------+
| sid | name     | branch | total_marks |
+-----+----------+--------+-------------+
|  20 | Kriyasha | IT     |         150 |
+-----+----------+--------+-------------+
1 row in set (0.01 sec)
```

LAB 6

Q.N.1. To find employee name containing exactly 5 characters use 5 instances of the _pattern character.

Solution:

Query: select * from employees where name like '_____';

Result:

```
mysql>  select * from employees where e_name like '_____';
Empty set (0.00 sec)
```

Q.N.2. Display all the records from left table with matched records from right table. There are two tables given to you as employee and department.

Solution:

Query: select * from employee as e

->left join departments as d

->on e.eid=d.eid;

Result:

```
mysql>  select * from employees as e
    -> left join department as d
    ->  on e.eid=d.eid;
+-----+----------------+------+-----------+----------+------------+---------+----------------+------+---------+
| eid | e_name         | age  | address   | salary   | commission | dept_id | dept_name      | eid  | dept_no |
+-----+----------------+------+-----------+----------+------------+---------+----------------+------+---------+
|   1 | Kyurisha Karki |   20 | Bhaktapur | 34100.00 |       NULL |     501 | IT             |    1 |      20 |
|   2 | Jharana Oli    |   19 | Bhaktapur | 45100.00 |       NULL |     202 | Psychology     |    2 |      10 |
|   3 | Sunita Rai     |   17 | Kathmandu | 23100.00 |       NULL |     502 | Human Resource |    3 |      25 |
|   4 | Nikita Bhujel  |   18 | Lalitpur  | 50600.00 |       NULL |    NULL | NULL           | NULL |    NULL |
|   5 | Abantika Lama  |   20 | Kathmandu | 45100.00 |       NULL |    NULL | NULL           | NULL |    NULL |
|   6 | Kusum Dangol   |   20 | Lalitpur  | 56100.00 |       NULL |    NULL | NULL           | NULL |    NULL |
|   7 | Neelam Nakarmi |   19 | Lalitpur  | 56100.00 |       NULL |    NULL | NULL           | NULL |    NULL |
|   8 | Pemba Gole     |   19 | Bhaktapur | 45100.00 |       NULL |    NULL | NULL           | NULL |    NULL |
|   9 | Orisha Shakya  |   20 | Lalitpur  | 50600.00 |       NULL |    NULL | NULL           | NULL |    NULL |
|  10 | Kriyasha Karki |   17 | Kathmandu | 47300.00 |       NULL |    NULL | NULL           | NULL |    NULL |
+-----+----------------+------+-----------+----------+------------+---------+----------------+------+---------+
10 rows in set (0.03 sec)
```

Q.N.3. Display employee name and department name from employee and department table using natural join.

Solution:

Query: select e.e_name, d.dept_name from employees e natural join department d;

Result:

```
mysql> select e.e_name, d.dept_name from employees e natural join department d;
+----------------+----------------+
| e_name         | dept_name      |
+----------------+----------------+
| Jharana Oli    | Psychology     |
| Kyurisha Karki | IT             |
| Sunita Rai     | Human Resource |
+----------------+----------------+
3 rows in set (0.00 sec)
```

Q.N.4. Replace the table employee with table emp.

Solution:

Query: rename table employees to emp;

Result:

```
mysql> desc emp;
+------------+---------------+------+-----+---------+-------+
| Field      | Type          | Null | Key | Default | Extra |
+------------+---------------+------+-----+---------+-------+
| eid        | int           | NO   | PRI | NULL    |       |
| e_name     | varchar(100)  | YES  |     | NULL    |       |
| age        | int           | YES  |     | NULL    |       |
| address    | varchar(100)  | YES  |     | NULL    |       |
| salary     | decimal(10,2) | YES  |     | NULL    |       |
| commission | decimal(10,2) | YES  |     | NULL    |       |
+------------+---------------+------+-----+---------+-------+
6 rows in set (0.07 sec)
```

Q.N.5. Write a query to display marks details in which marks starts with 4.

Solution:

Query: SELECT * from student where marks like '4%';

Result:

```
mysql> SELECT * from students where total_marks like '4%';
Empty set (0.00 sec)
```

Q.N.6. Write a query to display employee details in which name contains letter ‘ik’ in between.

Solution:

Query: select * from emp where name like '%ik%';

Result:

```
mysql> select * from employee where ename like '%ik%';
+-------+--------+--------------+----------+
| empno | ename  | job          | salary   |
+-------+--------+--------------+----------+
|    12 | Nikita | Data Analyst | 80000.00 |
+-------+--------+--------------+----------+
1 row in set (0.00 sec)
```

Q.N.7. Display all employees whose salary is either 3000, 4000 or 10000.

Solution:

Query: select * from employees where salary in (3000, 4000, 100000);

Result:

```
mysql> select * from emp where salary in (3000,4000,100000);
Empty set (0.03 sec)
```

Q.N.8. Display all employees with the salary except between 4000 to 5000.

Solution:

Query: select * from employee where salary not between 4000 and 5000;

Result:

```
mysql> select * from employee where salary not between 4000 and 5000;
+-------+----------+-----------------+-----------+
| empno | ename    | job             | salary    |
+-------+----------+-----------------+-----------+
|    11 | Jharana  | Web Developer   | 100000.00 |
|    12 | Nikita   | Data Analyst    |  80000.00 |
|    13 | Sunita   | DBA             |  85000.00 |
|    14 | Kyurisha | Graphic Designer |  80000.00 |
+-------+----------+-----------------+-----------+
4 rows in set (0.00 sec)
```

Q.N.9. Select all records from employee table where job is not manager.

Solution:

Query: select * from employee where job not like 'manager';

Result:

```
mysql> select * from employee where job not like 'manager';
+-------+----------+------------------+-----------+
| empno | ename    | job              | salary    |
+-------+----------+------------------+-----------+
|    11 | Jharana  | Web Developer    | 100000.00 |
|    12 | Nikita   | Data Analyst     |  80000.00 |
|    13 | Sunita   | DBA              |  85000.00 |
|    14 | Kyurisha | Graphic Designer |  80000.00 |
+-------+----------+------------------+-----------+
4 rows in set (0.03 sec)
```

LAB 7

Q.N.1. Create table customer with this attributes(ID,Name,Age,Address,Salary,Primary key(ID)) and insert minimum 7 records in the table.

Solution:

Query: create table customers(

-> ID int primary key,

-> Name varchar(100),

-> Age int,

-> Address varchar(255),

-> Salary decimal(10,2));

insert into customers (ID, Name, Age, Address, Salary)values

-> (1, 'Kumar Prasun', 38, 'Kathmandu', 70000.00),

-> (2, 'Sudip Raj Khadka', 39, 'Kathmandu', 75000.00),

-> (3, 'Ramesh Singh Saud', 37, 'Lalitpur', 60000.00),

-> (4, 'Mohan Singh Ayer', 39, 'Butwal', 70000.00),

-> (5, 'Bipin Timalsina', 31, 'Lalitpur', 80000.00),

-> (6, 'Sandhya Karki', 28, 'Lalitpur', 65000.00),

-> (7, 'Bhim Rawat', 35, 'Dharan', 72000.00);

Result:

```
mysql> select * from customers;
+----+-------------------+------+-----------+----------+
| ID | Name              | Age  | Address   | Salary   |
+----+-------------------+------+-----------+----------+
|  1 | Kumar Prasun      |   38 | Kathmandu | 70000.00 |
|  2 | Sudip Raj Khadka  |   39 | Kathmandu | 75000.00 |
|  3 | Ramesh Singh Saud |   37 | Lalitpur  | 60000.00 |
|  4 | Mohan Singh Ayer  |   39 | Butwal    | 70000.00 |
|  5 | Bipin Timalsina   |   31 | Lalitpur  | 80000.00 |
|  6 | Sandhya Karki     |   28 | Lalitpur  | 65000.00 |
|  7 | Bhim Rawat        |   35 | Dharan    | 72000.00 |
+----+-------------------+------+-----------+----------+
7 rows in set (0.00 sec)
```

Q.N.2. Write a SQL statement and display customer information where ID IN customer salary greater than 45000.

Solution:

Query: SELECT * FROM customer WHERE Salary > 45000;

Result:

```
mysql> SELECT * FROM customers WHERE Salary > 45000;
+----+------------------+------+-----------+----------+
| ID | Name             | Age  | Address   | Salary   |
+----+------------------+------+-----------+----------+
|  1 | Kumar Prasun     |   38 | Kathmandu | 70000.00 |
|  2 | Sudip Raj Khadka |   39 | Kathmandu | 75000.00 |
|  3 | Ramesh Singh Saud |   37 | Lalitpur  | 60000.00 |
|  4 | Mohan Singh Ayer |   39 | Butwal    | 70000.00 |
|  5 | Bipin Timalsina  |   31 | Lalitpur  | 80000.00 |
|  6 | Sandhya Karki    |   28 | Lalitpur  | 65000.00 |
|  7 | Bhim Rawat       |   35 | Dharan    | 72000.00 |
+----+------------------+------+-----------+----------+
7 rows in set (0.00 sec)
```

Q.N.3. To display NAME, LOCATION, PHONE_NUMBER of the students from student table whose section is A

Solution:

Query: select name, address, ph_number from student where section = 'A';

Result:

```
mysql> select name, address, ph_number from students where section = 'A';
+----------+------------+--------------+
| name     | address    | ph_number    |
+----------+------------+--------------+
| Kriyasha | 456 Elm St | 987-654-3210 |
| Jennie   | Pokhara    | 987-654-5210 |
+----------+------------+--------------+
2 rows in set (0.00 sec)
```

Q.N.4. Display employee's name whose salary is greater than 100000 and age is less than 20.

Solution:

Query: select name from employee where salary > 100000 and age < 20;

Result:

```
mysql> select ename from employee where salary > 100000 and age < 20;
Empty set (0.00 sec)
```

Q.N.5. Update table customer and update salary by 10000, it means increase the salary of customer by 10000.

Solution:

Query: update customers set salary = salary + 10000;

Result:

```
mysql> update customers set salary = salary + 10000;
Query OK, 7 rows affected (0.05 sec)
Rows matched: 7  Changed: 7  Warnings: 0

mysql> select * from customers;
+----+------------------+------+-----------+----------+
| ID | Name             | Age  | Address   | Salary   |
+----+------------------+------+-----------+----------+
|  1 | Kumar Prasun     |   38 | Kathmandu | 80000.00 |
|  2 | Sudip Raj Khadka |   39 | Kathmandu | 85000.00 |
|  3 | Ramesh Singh Saud |   37 | Lalitpur  | 70000.00 |
|  4 | Mohan Singh Ayer |   39 | Butwal    | 80000.00 |
|  5 | Bipin Timalsina  |   31 | Lalitpur  | 90000.00 |
|  6 | Sandhya Karki    |   28 | Lalitpur  | 75000.00 |
|  7 | Bhim Rawat       |   35 | Dharan    | 82000.00 |
+----+------------------+------+-----------+----------+
7 rows in set (0.00 sec)
```

Q.N.6. Write a SQL statement and print employee_id and salary whose employee_id is 120.

Solution:

Query: select ID, salary from employee where ID = 120;

Result:

```
mysql> select ID, salary from employee where Id = 120;
+------+----------+
| ID   | salary   |
+------+----------+
|  120 | 85000.00 |
+------+----------+
1 row in set (0.00 sec)
```

Q.N.7. Write a SQL statement and print employee details whose salary is greater than that of employee_id is 120.

Solution:

Query: select * from employee

->where salary > (select salary from employee where id = 120);

Result:

```
mysql> select * from employee
    -> where salary > (select salary from employee where id = 120);
+-------+---------+--------------+-----------+------+------+
| empno | ename   | job          | salary    | age  | ID   |
+-------+---------+--------------+-----------+------+------+
|    11 | Jharana | Web Developer | 100000.00 |   19 |  118 |
+-------+---------+--------------+-----------+------+------+
1 row in set (0.04 sec)
```

Q.N.8. Write a SQL statement and print employee name and salary whose post is project manager.

Solution:

Query: select ename, salary from employee where job = 'Project Manager';

Result:

```
mysql> select ename,salary from employee where job = 'Project Manager';
Empty set (0.00 sec)
```

Q.N.9. Write a SQL statement to add joining_date of employee in employee table.

Solution:

Query: ALTER TABLE employees ADD COLUMN joining_date DATE;

Result:

```
mysql> desc employee;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| empno        | int           | NO   | PRI | NULL    |       |
| ename        | varchar(100)  | YES  |     | NULL    |       |
| job          | varchar(100)  | YES  |     | NULL    |       |
| salary       | decimal(10,2) | YES  |     | NULL    |       |
| age          | int           | YES  |     | NULL    |       |
| ID           | int           | YES  |     | NULL    |       |
| joining_date | date          | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
7 rows in set (0.05 sec)
```

Q.N.10. Write a SQL statement to add email address after address of employee in employee table.

Solution:

Query: ALTER TABLE employee ADD COLUMN email VARCHAR(255)

->AFTER address;

Result:

```
mysql> desc employee;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| empno        | int           | NO   | PRI | NULL    |       |
| ename        | varchar(100)  | YES  |     | NULL    |       |
| job          | varchar(100)  | YES  |     | NULL    |       |
| salary       | decimal(10,2) | YES  |     | NULL    |       |
| age          | int           | YES  |     | NULL    |       |
| ID           | int           | YES  |     | NULL    |       |
| joining_date | date          | YES  |     | NULL    |       |
| address      | varchar(255)  | YES  |     | NULL    |       |
| email        | varchar(255)  | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
9 rows in set (0.04 sec)
```

Q.N.11. Update employee table for adding employee salary with 10000 in employee table.

Solution:

Query: UPDATE employee SET salary = salary + 10000;

Result:

```
mysql> select * from employee;
+-------+----------+------------------+-----------+------+------+--------------+---------+-------+
| empno | ename    | job              | salary    | age  | ID   | joining_date | address | email |
+-------+----------+------------------+-----------+------+------+--------------+---------+-------+
|    11 | Jharana  | Web Developer    | 110000.00 |   19 |  118 | NULL         | NULL    | NULL  |
|    12 | Nikita   | Data Analyst     |  90000.00 |   19 |  119 | NULL         | NULL    | NULL  |
|    13 | Sunita   | DBA              |  95000.00 |   19 |  120 | NULL         | NULL    | NULL  |
|    14 | Kyurisha | Graphic Designer |  90000.00 |   20 |  121 | NULL         | NULL    | NULL  |
+-------+----------+------------------+-----------+------+------+--------------+---------+-------+
4 rows in set (0.00 sec)
```