---------- **CREATING A DATABASE** ----------

```
/*1. Creating a database in MySQL.*/
create database IF NOT EXISTS `company`;

/*2. Display existing database in MySQL.*/
SHOW databases;

/*3. Using the created database.*/
USE `company`;

/*4. Drop the database.*/
DROP database IF EXISTS `company`;
```

---------- **CREATING A TABLE** ----------

```
/*5. Creating a table.*/
CREATE table `employee`(
`emp_no` int PRIMARY KEY,
`emp_id` varchar(20),
`emp_name` varchar(40),
`emp_sal` float,
`dept_no` int
);

/* To delete table.*/
DROP table `employee`;

CREATE table `Dept`(
`dept_no` int,
`dept_name` varchar(20),
`location` varchar(20)
);

DROP table `Dept`;
```

```
INSERT INTO `employee` (emp_no,emp_id, emp_name, emp_sal, dept_no)
VALUES
('1','100', 'Gianina Milleton', 50000, 1),
('2','101', 'Jewel Cobon', 25000, 1),
('3','102', 'Kevyn Mabson', 5000, 2),
('4','103', 'Michele Winsiowiecki', 55000, 1),
('5','104', 'Simonette Craigie', 87000, 1),
('6','105', 'Jacques Hearl', 35000, 2),
('7','106', 'Melinde Milverton', 96000, 3),
('8','107', 'Sherm Elderidge', 20000, 1),
('9','108', 'Lib Gladbach', 10000, 5),
('10','109', 'Madelon Bercevelo', 24000, 4),
('11','110', 'Paulina Cromblehome', 550000, 1),
('12','111', 'Shel Dillestone', 52000, 2);

/*To show tables in database.*/
SHOW tables;

/*Show all fields of table.*/
DESCRIBE `company`.`employee`;
DESCRIBE `company`.`dept`;

/*To view values entered in table.*/
SELECT * FROM `employee`;

/*To empty the table.*/
TRUNCATE `dept`;

INSERT INTO `dept`(dept_no, dept_name, location)          /*To insert data in table.*/
VALUES  ('1', 'Software development', 'Banglore'),
('2', 'Finance', 'Mumbai'),
('3', 'HR', 'Delhi'),
('4', 'Marketing', 'Pune'),
('5', 'Transport', 'Nashik');
```

```
/*To see values entered in table.*/
SELECT * FROM `dept`;
```

---------- **USING ALL TYPES OF SELECT** ----------

```
/*1. Basic select statement. To show the data inserted in tables.*/
SELECT * FROM `employee`;
SELECT * FROM `dept`;

/*2.Select using distinct, it eliminates duplicate rows from table.*/
SELECT DISTINCT `dept_no` FROM `employee`;

/*3. Select by column name.*/
SELECT `emp_name` FROM `employee`;

/*4. Select using where.*/
SELECT `emp_name` FROM `employee`
WHERE `emp_no` = 5;

/*5. All select statement with like. Like is a wildcard.*/

    SELECT * FROM `employee` WHERE `emp_name` LIKE "%cra%";

    SELECT * FROM `employee` WHERE `emp_name` LIKE "Gia%";

    SELECT `emp_name` FROM `employee` WHERE `emp_name` LIKE "%dge";

    SELECT * FROM `employee` WHERE `emp_name` LIKE "%Je__l%";

/*6 Select with case.*/

  SELECT CASE
  WHEN `emp_no` = 1 THEN 'ONE'
  WHEN `emp_no` = 2 THEN 'TWO'
```

```
  WHEN `emp_no` = 3 THEN 'THREE'
  WHEN `emp_no` = 4 THEN 'FOUR'
  ELSE 'other'
  END 'row_no'
  FROM `employee`;

/*7. Select with limit clause, limits the number of rows.*/
SELECT * FROM `employee`
ORDER BY `emp_no`
LIMIT 3;

/*8. All select with between.*/
   SELECT * FROM `employee`
   WHERE `emp_sal` >= 20000 AND `emp_sal` <= 70000;

   SELECT * FROM `employee`
   WHERE `emp_sal` BETWEEN 20000 AND 50000;

   SELECT * FROM `employee`
   WHERE `emp_sal` NOT BETWEEN 20000 AND 50000;

/*9. LIMIT or OFFSET.*/

SELECT * FROM `employee` ORDER BY `emp_sal` ASC LIMIT 5;

/*10. Extra.*/

SELECT * FROM `employee`
WHERE `dept_no` = 1 OR `dept_no` = 2 OR `dept_no` = 3;

SELECT * FROM `employee`
WHERE `dept_no` in (1,2,3);
```

```
---------------------------------------------- UPDATE AND DELETE ----------------------------------------------------------------

/* Updating a row.*/

UPDATE `employee` SET `emp_sal` = 100000

WHERE `emp_sal` = 5000;


UPDATE `employee` SET `emp_sal` = 100000*12

WHERE `emp_sal` = 5000;


/*Deleting a row.*/

DELETE FROM `employee` WHERE `emp_no` = 12;


---------------------------------------------- ARITHMETIC OPERATORS ----------------------------------------------------------------

/* PI */

SELECT pi();


/*Trignometry.*/

SELECT SIN(pi());

SELECT COS(pi());

SELECT TAN(pi());


/*Round a decimal number to integer value.*/

SELECT ROUND(4.51);

SELECT ROUND(4.49) ;

SELECT ROUND(-4.51) ;


/*Round Up number.*/

SELECT CEIL(1.23);

SELECT CEILING(4.83);


/*Round down number.*/

SELECT FLOOR(1.99);


/*Round a decimal number to a specified number of decimal places.*/

SELECT ROUND(1234.987, 1);
```

```
SELECT ROUND(1234.987, -2);

/*Raise a number to power.*/
SELECT POW(2,2);
SELECT POW(4,2);

/*Square root.*/
SELECT SQRT(16);
SELECT SQRT(-16);

/*Absolute.*/
SELECT ABS(2);
SELECT ABS(-46);

SELECT `emp_name`,`emp_sal`,`emp_sal`*12 as "Annual"
FROM `employee`;

SELECT `emp_name`,`emp_sal`, `emp_sal`*12*0.4 as "HRA"
FROM `employee`;
```

---------- **ORDER BY** ----------

```
/*ORDER BY.*/

SELECT `emp_no` FROM `employee`
ORDER BY `emp_no` desc;

SELECT * FROM `employee`
ORDER BY `emp_sal` desc, `emp_no`;
```

---------- **STRING OPERATIONS** ----------

```
/*LENGTH*/
SELECT LENGTH('foobar');
```

```
/* CHAR_LENGTH()*/
SELECT  CHAR_LENGTH('Shrey@s');

/*HEX(str), Convert the argument to hexadecimal.*/
SELECT HEX('fööbar');

/*SUBSTRING()*/
SELECT SUBSTRING('foobarbaz', 4);
SELECT SUBSTRING('foobarbaz', -6);

SELECT SUBSTRING('foobarbaz', 4, 3);

/*Concatination Operation.*/

CREATE TABLE `Demo1`(
`Id` int,
`first_name` varchar(20),
`last_name` varchar(20)
);

INSERT INTO `Demo1`(Id,first_name, last_name) VALUES
(1,'Shreyas','Kulkarni');

DROP TABLE `Demo1`;

TRUNCATE `Demo1`;

SELECT * FROM `Demo1`;

SELECT concat(first_name, last_name) FROM `Demo1`;

/*UPPER()*/
SELECT UPPER('fOoBar');
```

```
SELECT UPPER(first_name) FROM `DEMO1`;

/*LOWER()*/
SELECT LOWER(last_name) FROM `Demo1`;

/*REPLACE()*/
SELECT REPLACE (first_name, 'yas', 'pad') FROM `Demo1`;

SELECT REPLACE (first_name, 'S', 'R') FROM `Demo1`;

/*FIND_IN_SET*/
SELECT FIND_IN_SET('b','a,b,c');

/*LPAD and RPAD*/
SELECT lpad(emp_name,25,'*') AS EmployeeName FROM `employee`
WHERE `emp_no` = 1;

SELECT rpad(emp_name,25,'*') AS EmployeeName FROM `employee`
WHERE `emp_no` = 1;

Select RPAD(LPAD(emp_name,23,'* '),30,'* ') AS EmployeeName FROM `employee`
WHERE `emp_no` = 1;

/*LTRIM and RTRIM*/
-- removes blank space from both the sides.

/*Substr*/
SELECT substr(emp_name, 3,2) FROM `employee`;
SELECT substr(emp_name, 3) FROM `employee`;

/*ASCII*/
SELECT ascii(emp_name) FROM `employee`;
```

```
/*Truncate*/
SELECT truncate(emp_sal, 0) FROM `employee`;

/*MOD*/
SELECT mod(9,5) FROM dual;

---------- DATE AND TIME ----------

/*Shows current date and time of sever.*/
SELECT sysdate() FROM dual;

/*Shows date and time when statement got exectue.*/
SELECT now() FROM dual;

SELECT sysdate(), now(), sleep(20), sysdate(),now() FROM dual;

/*Add date*/
SELECT adddate(sysdate(),1) FROM dual;
SELECT adddate(sysdate(),2) FROM dual;
SELECT adddate(sysdate(),7) FROM dual;
SELECT adddate(sysdate(),-2) FROM dual;

---------- LEAST AND GREATEST ----------

SELECT greatest(emp_sal,10000) FROM `employee`;
SELECT least(emp_sal, 50000) FROM `employee`;

---------- INSERT NEW COLUMN ----------

ALTER TABLE `employee`
ADD `hire_date` varchar(20) AFTER `dept_no`;
```

----------**JOINS**----------

-- EQUIJOIN

```
select dname,ename from dept, emp
where dept.deptno = emp.deptno;

select dname,ename from emp,dept
where dept.deptno = emp.deptno
group by dname
having dname like 'TRN';
```

-- INEQUIJOIN

```
select dname, ename from emp,dept
where  emp.deptno  != dept.deptno
order by 1;
```

-- RIGHT OUTERJOIN

```
select dname,ename from emp right outer join dept
on (dept.deptno = emp.deptno);
```

-- LEFT OUTERJOIN

```
select dname, ename from emp left outer join dept
on (dept.deptno = emp.deptno);
```

-- FULL JOIN

```
select dname,ename from emp right outer join dept
on (dept.deptno = emp.deptno)
union
select dname, ename from emp left outer join dept
on (dept.deptno = emp.deptno);
```

-- CARTESIAN JOIN (CROSS JOIN)

```
select dname, ename from emp, dept
order by 1;
```

-- SELF JOIN

```
select a.ename, b.ename from emp b, emp a

where a.mgr = b.empno;
```

SAMPLE JOIN QUESTIONS:

/*3. Write a query to find the name (first_name, last_name), job, department ID and name of the employees who works in London.*/ (EQUI JOIN SAMPLE)

```
select employee_id, first_name, last_name, job_title, employees.department_id, department_name, locations.city
from employees, departments, jobs, locations

where  employees.department_id = departments.department_id and employees.job_id = jobs.job_id and
locations.location_id = departments.location_id and city = 'London';
```

/*4. Write a query to find the employee id, name (last_name) along with their manager_id and name (last_name).*/ (SELF JOIN SAMPLE)

```
select a.employee_id , a.Last_name,b.employee_id as Manager_id, b.Last_name from employees b, employees a

where a.Manager_id=b.Employee_id;
```

/*9. Write a query to display the department name, manager name, and city.*/

```
select department_name, employees.first_name, employees.last_name, locations.city

from employees, departments, locations

where employees.employee_id = departments.manager_id and departments.location_id = locations.location_id;
```

/*13. Write a query to display department name, name (first_name, last_name), hire date, salary of the manager for all managers whose experience is more than 15 years.*/

```
select DEPARTMENT_NAME, first_name, last_name, HIRE_DATE, salary from departments, employees

where departments.DEPARTMENT_ID = employees.DEPARTMENT_ID

and employees.EMPLOYEE_ID = departments.MANAGER_ID

and (sysdate()-HIRE_DATE) > (15*365);
```

----------**SUB QUERY**----------

/* 1. Write a query to find the name (first_name, last_name) and the salary of the employees who have a higher salary than the employee whose last_name='Bull'. */

```
select first_name,last_name,salary from employees
where salary > (select salary from employees where last_name='BULL');
```

/* 2. Write a query to find the name (first_name, last_name) of all employees who works in the IT department. */

```
select first_name, last_name from employees
where department_id = (select department_id from departments where department_name = 'IT');
```

/* 3. Write a query to find the name (first_name, last_name) of the employees who have a manager and worked in a USA based department. */

```
select first_name, last_name from employees
where manager_id != 0 and manager_id in (select manager_id from departments
where location_id in (select location_id from locations
where country_id in (select country_id from countries
where country_name = 'United States of America')));
```

/* 7 Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the average salary and works in any of the IT departments. */

```
select first_name, last_name ,salary from employees
where job_id ='IT_PROG' && salary > (select avg(salary) from employees);
```

/* 8 Write a query to find the name (first_name, last_name), and salary of the employees who earns more than the earning of Mr. Bell. */

```
select  first_name, last_name, salary from employees
where salary >
(select salary from employees
where last_name = 'bell');
```

----------PROCEDURE SAMPLES ----------

/* 1. Write a program that computes the perimeter and the area of a rectangle. Define your own values for the length and width.

(Assuming that L and W are the length and width of the rectangle, Perimeter = 2*(L+W) and Area = L*W. */

```
create table rectangle
(
area float,
peri float
);

delimiter $
create procedure area(l int,w int)
begin
declare area float;
declare peri float;
set area = l*w;
set peri = 2*(l+w);
insert into rectangle values (area, peri);
end; $
delimiter ;

drop procedure area;
drop table rectangle;

-- set @x = 20;
-- set @y = 30;
call area(10, 20);

select * from rectangle;
```

```
/* 2.Write a program that declares an integer variable called num, assigns a value to it,
and computes and inserts into the tempp table the value of the variable itself, its
square, and its cube. */

create table sample
(
num int,
square int,
cuube int
);

delimiter $
create procedure operations(num int)
begin
declare square float;
declare cuube float;
set square = num*num;
set cuube = num*num*num;
insert into sample values(num, square, cuube);
end; $
delimiter ;

call operations(2);

select * from sample;
```

```
/* 3. Convert a temperature in Fahrenheit (F) to its equivalent in Celsius (C) and vice versa. The required formulae are:- C= (F-32)*5/9

F= 9/5*C + 32 */

create table sample2

(

temperature float,

degree varchar(300)

);

drop table sample2;


delimiter $

create procedure converter(num int, type varchar(15))

begin

declare Fahrenheit float;

declare Celsius float;

if (type = 'F' or 'Fahrenheit') then

 set Celsius = (num - 32) * 5/9;

 insert into sample2 values ( Celsius, 'Celsius');

 elseif (type = 'C' or 'Celsius') then

  set Fahrenheit = 9/5*num + 32;

  insert into sample2 values ( Fahrenheit, 'Fahrenheit');

 else

 insert into sample2 values (num, 'Incorrect Temp Formate');

 end if;

end; $

delimiter ;


drop procedure converter;

call converter(32, 'F');

select * from sample2;
```

----------**FUNCTIONS**----------

/* write a stored function by the name of Num_cube. The stored function should return the cube of a number ‘N’. The number ‘N’ should be passed to the stored function as a parameter. Calling program for the stored function need not be written. */

```
delimiter //
create function num_cube(N int)
returns int
deterministic
begin
return N*N*N;
end; //
delimiter ;

set @cube=num_cube(5);
select @cube from dual;
```