```
create database employees_id;

show tables;


select * from tbl_emp_detail_data;


create table emp_id as select * from tbl_emp_detail_data;


select * from emp_id;


select concat_ws(",",emp_id,first_name,last_name,salary,joining_date,department,gender,job_title)
from emp_id;


select * from emp_id

order by  emp_id desc limit 1;


select * from emp_id

order by  emp_id  asc limit 1;


select rand() from emp_id;


select * from emp_id

order by rand() limit 1;


select upper(first_name) as first_name_upper  from emp_id;


select lower(first_name) as first_name_upper  from emp_id;


select concat_ws(" ",first_name,last_name) from emp_id;
```

```
select concat_ws(" ",first_name,"hello") from emp_id;

select * from emp_id
where first_name="malli";

select * from emp_id
where first_name in ("malli", "meena" ,"anjali");

select * from emp_id
where first_name not in ("malli", "meena" ,"anjali");

select * from emp_id
where position( 'o' in first_name);

select * from emp_id
where first_name like "v%";

select * from emp_id
where first_name like "%i";

select * from emp_id
where substring(first_name,1,1) between 'm' and 'v';

select * from emp_id
where substring(first_name,1,1)not  between 'm' and 'v';
```

```
select * from emp_id
where first_name like 'm%' and length(first_name)=5;

select distinct department as distinct_department from emp_id;

select count(*) from emp_id;

select concat(lower(substring(first_name,1,1)),upper(substring(first_name ,2))) from emp_id;
select concat(upper(substring(first_name,1,1)),lower(substring(first_name ,2))) from emp_id;

select group_concat(first_name, ",") as employee_name
from emp_id;

select min(salary) as min_salary,max(salary) as max_salary,avg(salary) as avg_salary from emp_id;

select max(salary)-min(salary) as diff_salary from emp_id;

SELECT
   ROUND(((MAX(salary) - MIN(salary)) * 100.0 / MAX(salary)), 1) || '%' AS percent_difference
FROM
   emp_id;

select * from emp_id
where first_name <> ltrim(first_name);

select * from emp_id
where first_name <> rtrim(first_name);
```

```
select * from emp_id
where first_name <> trim(first_name);

select * from emp_id
where salary ='50000';

select *
from emp_id
where joining_date = (
   select max(joining_date)
   from emp_id
   where department = department
);

select first_name ,
CASE
WHEn  gender='male' then 'm'
when gender='female' then 'f'
end as gender
from emp_id;

select first_name,salary,
case
when salary < 50000 then 'low'
when salary between 50000 and 60000 then 'medium'
```

```
else
'high'
end as 'salary_end'
from emp_id;

select first_name,department,
case
when department='it' then 'technical'
when department='hr' then 'human resources'
when department='finance' then 'accounting'
else
'others'
end as 'department_classification'
from emp_id;

select first_name,salary,
case
when salary <=50000 then 'eligible for salary rise'
else
'not eligibile'
end as salary_eligability
from emp_id;

select first_name,joining_date,
case
when joining_date < '2022-01-01' then 'expreinced '
else
```

```
'new hire'
end as employement_status
from emp_id;

select first_name,salary,
case
when salary > 60000 then salary*0.10
when salary between 50000 and 60000 then salary*0.07
else
salary*0.05
end as employee_bonus
from emp_id;

select * from emp_id;

select first_name,salary,
case
when salary < 50000 then 'junior employee'
when salary between 50000 and 60000 then 'intermediate level employee'
else
'senior employee'
end as 'senior employee'
from emp_id;

select first_name,department,
case
when department='it' and salary > 55000 then 'senior it employee'
```

```
else
'junior it employee'
end as 'job for it employee'
from emp_id
where department='it';

select first_name,joining_date,
case
when joining_date > '2024-01-01' then 'junior'
else
'long-term employee'
end as 'recent joiner status'
from emp_id;

select first_name,joining_date,
case
when joining_date > '2021-01-01' then'give 10 days'
when joining_date between '2021-01-01' and '2023-01-01' then 'give 20 days leave'
else
'give  25 days leave'
end as 'leave entitlement'
from emp_id;

select first_name,salary,department,
case
when salary > 60000 and department = 'it' then 'eligibale for promotion'
else
```

```
'not eligibale for promotion'
end as 'promotion eligability'
from emp_id;

select first_name,salary,department,
case
when salary < 50000  then 'eligibale for overtime'
else
'not eligibale for ivertime'
end as 'overtime eligability'
from emp_id;

select first_name,salary,department,
case
when salary > 60000 and department='hr' then 'hr executive'
when salary > 55000 and department='finance' then 'finance manager'
else
'regular employee'
end as 'job title'
from emp_id;

select
   first_name,
   salary,
   case
      when salary > (select avg(salary) from emp_id) then 'above average'
      else 'below average'
```

```
    end as salary_comparison
from emp_id;

select * from emp_id;

select department,sum(salary) as total_salary
from emp_id
group by department
order by total_salary asc;

select department,sum(salary) as total_salary
from emp_id
group by department
order by total_salary desc;

select * from tbl_project_deatils;

select project_name
from tbl_project_deatils
group by project_name
having count(emp_id_no)>1;

select department,count(distinct(department)) as total_department,sum(salary) as total_salary
from emp_id
group by department;

select department,avg(salary) as avg_salary,min(salary) as min_salary
```

```
from emp_id
group by department
order by avg_salary asc;

select department,count(*) as total_count,
case
when count(*) > then 'large'
when  count(*) between 3 and 5 then 'medium'
else
'small'
end as separtment_size
from emp_id
group by department;

SELECT
   department,
   COUNT(*) AS employee_count,
   CASE
      WHEN COUNT(*) > 5 THEN 'Large'
      WHEN COUNT(*) BETWEEN 3 AND 5 THEN 'Medium'
      ELSE 'Small'
   END AS department_size
FROM emp_id
GROUP BY department;

select department,count(*) as total_count,avg(salary)as avg_sal,
case
```

```
when salary >60000 then 'high pay'
when salary between 50000 and 40000 then 'medium pay'
else
'small pay'
end as avg_pay
from emp_id
group by department;

select
   department,
   count(*) as total_count,
   avg(salary) as avg_sal,
   case
      when avg(salary) > 60000 then 'high pay'
      when avg(salary) between 40000 and 60000 then 'medium pay'
      else 'low pay'
   end as avg_pay
from emp_id
group by department;

select * from emp_id;

select department,gender,count(emp_id)
from emp_id
group by department,gender;

SELECT
```

```
    joining_date,
    COUNT(*) AS employee_count,
    CASE
        WHEN joining_date > 5 THEN 'high hering'
        WHEN joining_date BETWEEN 3 AND 5 THEN 'heiring'
        ELSE 'Small'
    END AS hireing
FROM emp_id
GROUP BY joining_date;

select department,max(salary) as max_salary,
case
when max(salary) > 70000 then 'senior leader'
else
'medium leader'
end as 'department_wise salary'
from emp_id
group by department;

select
    department,
    count(case when salary > 60000 then 1 end) as emp_above_60k,
    case
        when count(case when salary > 60000 then 1 end) > 2
            then 'high-paying team'
        else 'normal team'
    end as team_classification
```

```
from emp_id
group by department;

select joining_date,
year(joining_date) as join_year ,
month(joining_date) as join_month,
day(joining_date) as join_date,
current_date() as today_date
from emp_id;

select
   joining_date,
   current_date() as today_date,
   timestampdiff(day, joining_date, current_date()) as diff_in_days,
   timestampdiff(month, joining_date, current_date()) as diff_in_months
from emp_id;

select *
from emp_id
where year(joining_year) = 2020;

select *
from emp_id
where month(joining_date) = 2;

select *
from emp_id
```

```
where joining_date between '2021-01-01' and '2021-12-01';


select
e.first_name,
p.project_name
from emp_id
join tbl_project_deatils p
on e.emp_id =p.emp_id
order by first_name;


SELECT
    e.first_name,
    p.project_name
FROM emp_id e
JOIN tbl_project_details p
    ON e.emp_id = p.emp_id
ORDER BY e.first_name;



select * from tbl_project_deatils;


SELECT
    e.first_name AS employee_name,
    p.project_name
FROM emp_id e
INNER JOIN tbl_project_deatils p
    ON e.emp_id = p.emp_id
```

```
ORDER BY e.first_name;


SELECT
   e.first_name AS employee_name,
   p.project_name
FROM emp_id e
INNER JOIN tbl_project_deatils p
   ON e.emp_id_no = p.emp_id_no
ORDER BY e.first_name;


SELECT
   e.first_name AS employee_name,
   p.project_name
FROM emp_id e
INNER JOIN tbl_project_deatils p
   ON e.emp_id = p.emp_id_no
ORDER BY e.first_name;


select
e.first_name as employee_name,
p.project_name
from emp_id e
left join tbl_project_deatils p
on e.emp_id=p.emp_id_no
order by e.first_name;


SELECT
```

```
    e.first_name AS employee_name,
    COALESCE(p.project_name, '-No Project Assigned') AS project_name
FROM emp_id e
LEFT JOIN tbl_project_deatils p
    ON e.emp_id = p.emp_id_no
ORDER BY e.first_name;


select
e.first_name as employee_name ,
p.project_name
from emp_id e
left join tbl_project_deatils p
on e.emp_id= p.emp_id_no
union
select
e.first_name as employee_name ,
p.project_name
from emp_id e
right join tbl_project_deatils p
on e.emp_id=p.emp_id_no
order by employee_name;


select
e.first_name as employee_name,
p.project_name
from emp_id e
left join tbl_project_deatils p
```

```
on e.emp_id=p.emp_id_no
where e.emp_id is not null;

select
e.first_name as employee_name,
p.project_name
from emp_id e
inner join tbl_project_deatils p
on e.emp_id=p.emp_id_no
where e.emp_id in(
select emp_id
from tbl_project_deatils
group by emp_id
having count(project_id)>1
)
order by e.first_name;

select
p.project_name,
e.first_name as  employee_name
from tbl_project_deatils p
inner join emp_id e
on p.emp_id_no = e.emp_id
where p.project_id in(
select project_id
from tbl_project_deatils
group by project_id
```

```
having count(emp_id is not null) > 1

)

order by p.project_name,e.first_name;


SELECT

   p.*

FROM tbl_project_deatils p

LEFT JOIN emp_id e

   ON p.emp_id_no = e.emp_id

WHERE e.emp_id IS NULL;


select

e.first_name as employee_name,

p.project_name

from emp_id

left join tbl_project_deatils p

on e.emp_id=p.emp_id_no

union

select

e.first_name as employee_name,

p.project_name

from emp_id e

right join tbl_project_deatils p

on e.emp_id=p.emp_id_no

order by employee_name;


SELECT
```

```
    e.first_name AS employee_name,
    p.project_name
FROM emp_id e
LEFT JOIN tbl_project_deatils p
    ON e.emp_id = p.emp_id_no

UNION

SELECT
    e.first_name AS employee_name,
    p.project_name
FROM emp_id e
RIGHT JOIN tbl_project_deatils p
    ON e.emp_id = p.emp_id_no

ORDER BY employee_name;

SELECT
    p.project_name
FROM tbl_project_deatils p
LEFT JOIN emp_id e
    ON p.emp_id_no = e.emp_id
WHERE e.emp_id IS NULL;

SELECT
    e.first_name AS employee_name,
    p.project_name
```

```
FROM emp_id e
INNER JOIN tbl_project_deatils p
   ON e.emp_id = p.emp_id_no
WHERE e.emp_id IN (
   SELECT emp_id
   FROM tbl_project_deatils
   GROUP BY emp_id
   HAVING COUNT(project_id) > 1
)
ORDER BY e.first_name, p.project_name;

SELECT
   p.project_name,
   e.first_name AS employee_name
FROM emp_id e
JOIN tbl_project_deatils p
   ON e.emp_id = p.emp_id_no
WHERE p.project_name IN (
   SELECT project_name
   FROM tbl_project_deatils
   GROUP BY project_name
   HAVING COUNT(emp_id_no) > 1
)
ORDER BY p.project_name, e.first_name;

SELECT p.*
FROM tbl_project_deatils p
```

```
LEFT JOIN emp_id e
   ON p.emp_id_no = e.emp_id
WHERE e.emp_id IS NULL;


select
e.emp_id,
e.first_name,
e.department,
e.salary,
row_number() over
(
partition by e.department
order by e.salary desc
)
as row_num
from emp_id e;


select
e.emp_id,
e.first_name,
e.department,
e.salary,
rank() over(
partition by e.department
order by e.salary desc
)
as salary_rank
```

```
from emp_id e;


select
e.emp_id,
e.first_name,
e.department,
e.salary,
dense_rank() over (
partition by e.department
order by e.salary desc
)
as dense_salary_rank
from emp_id e;


select
emp_id,
first_name,
department,
salary
from (
select
e.emp_id,
e.first_name,
e.department,
e.salary,
rank() over(
partition by e.department
```

```
order by e.salary
) as salary_rank
from emp_id e
)ranked_employee
where salary_rank=1;

select
emp_id,
first_name,
department,
salary
from (
select
e.emp_id,
e.first_name,
e.department,
e.salary,
rank() over(
partition by e.department
order by e.salary
) as salary_rank
from emp_id e
)ranked_employee
where salary_rank=2;

select
e.emp_id,
```

```
e.first_name,
e.department,
timestampdiff(year,e.joining_date,curdate()) as years_of_experience,
rank() over(
partition by e.department
order by timestampdiff(year,e.joining_date,curdate()) desc
)
as exp_rank
from emp_id e;

select
emp_id,
first_name,
department,
joining_date
from(
select
e.emp_id,
e.first_name,
e.department,
e.joining_date,
rank() over(
partition by e.department
order by e.joining_date asc
)
as join_rank
from emp_id e
```

```
)
ranked_employee
where join_rank=1;

select
e.emp_id,
e.first_name,
e.department,
e.salary,
avg(e.salary) over (partition by e.department) as dept_avg_salary
from emp_id e
where e.salary  > avg(e.salary) over (partition by e.department);

select
e.emp_id,
e.first_name,
e.department,
e.salary
from emp_id e
where e.salary > (
select avg(salary)
from emp_id
where department =e.department
);

select
e.emp_id,
```

```
e.first_name,
e.department,
e.salary,
e.job_title,
rank() over(
partition by e.department ,e.job_title
order by e.salary desc
) as salary_rank
from emp_id e;

select
emp_id,
first_name,
department,
salary
from (
select
e.emp_id,
e.first_name,
e.department,
e.salary,
rank() over(
partition by e.department
order by e.salary desc
)
as salary_rank
from emp_id e
```

```
)
ranked_employee
where salary_rank <= 3
order by department,salary_rank,salary desc;


SELECT emp_id,
       first_name,
       department,
       salary,
       PERCENT_RANK() OVER (
          PARTITION BY department
          ORDER BY salary DESC
       ) AS salary_percent_rank
FROM emp_id
WHERE PERCENT_RANK() OVER (
          PARTITION BY department
          ORDER BY salary DESC
       ) <= 0.10;


SELECT
    e.emp_id,
    e.first_name,
    e.joining_date,
    YEAR(e.joining_date) AS join_year,
    ROW_NUMBER() OVER (
        PARTITION BY YEAR(e.joining_date)
        ORDER BY e.emp_id
```

```
    ) AS row_num
FROM emp_id e;


SELECT
    e.emp_id,
    e.first_name,
    e.department,
    COUNT(p.project_id) AS project_count,
    RANK() OVER (
        PARTITION BY e.department
        ORDER BY COUNT(p.project_id) DESC
    ) AS project_rank
FROM emp_id e
LEFT JOIN ProjectDetail p
      ON e.emp_id = p.emp_id_no
GROUP BY e.emp_id, e.first_name, e.department;


SELECT e.emp_id,
       e.first_name,
       e.department,
       e.salary
FROM emp_id e
WHERE e.department IN (
    SELECT department
    FROM emp_id
    GROUP BY department
    HAVING COUNT(emp_id) = 1
```

```
);


SELECT emp_id,
      first_name,
      department,
      join_date
FROM (
   SELECT e.emp_id,
         e.first_name,
         e.department,
         e.joining_date,
         RANK() OVER (
            PARTITION BY e.department
            ORDER BY e.joining_date ASC
         ) AS join_rank
   FROM emp_ide
) ranked_employees
WHERE join_rank = 1;
```