**Todays CT**

**DDL**

CREATE TABLE `classroom`(

`building` varchar(20) NOT NULL,#primary

`room\_num` varchar(20) NOT NULL,

`capacity` INT,

PRIMARY KEY(`building`,`room\_num`)

);

CREATE TABLE `department`(

`dept\_name` varchar(20) NOT NULL,#primary

`building` varchar(20) NOT NULL,

`budget` varchar(20) ,

PRIMARY KEY(`dept\_name`),

FOREIGN KEY(`building`) REFERENCES classroom(`building`)

);

CREATE TABLE `course`(

`course\_id` varchar(20) NOT NULL,#primary

`title` varchar(20) ,

`dept\_name` varchar(20) NOT NULL,#foreign

`credits` float(2,1),

PRIMARY KEY(`course\_id`),

FOREIGN KEY(`dept\_name`) REFERENCES depertment(`dept\_name`)

);

CREATE TABLE `instructor`(

`ID` varchar(20) NOT NULL,#primary

`name` varchar(20) ,

`dept\_name` varchar(20) NOT NULL,#foreign

`salary` float(2,1),

PRIMARY KEY(`ID`)

FOREIGN KEY(`dept\_name`) REFERENCES depertment(`dept\_name`)

);

**DML**

insert into classroom(building,room\_num,capacity)

VALUES('Padma','011A',45)

insert into depertment(dept\_name,building,budget)

VALUES('EEE','Meghna','870000.00')

insert into depertment(dept\_name,building,budget)

VALUES('EEE','Meghna','870000.00')

**– Update table name(DDL)**

**Rename TABLE takes to takings**

**– Update column name(DDL)**

**ALTER TABLE advisor CHANGE COLUMN s\_ID student\_ID Varchar(20);**

**ALTER TABLE advisor CHANGE COLUMN i\_ID Teacher\_ID Varchar(20);**

**– ADD and Drop table**

**CREATE TABLE `prerequisite` (**

**`course\_id` VARCHAR(20) NOT NULL,**

**`title` VARCHAR(20) NOT NULL,**

**PRIMARY KEY (`course\_id`),**

**FOREIGN KEY (`course\_id`) REFERENCES course (`course\_id`)**

**);**

**DROP table prerequisite;**

**– Update Column data**

**UPDATE advisor SET Teacher\_ID = 'NaS' WHERE Teacher\_ID = 'Nas';**

**Primary key and foreign key delete and update**

ALTER TABLE table\_name DROP CONSTRAINT constraint\_name;

ALTER TABLE students DROP CONSTRAINT pk\_students\_id;

**Searching(ripas)**

1. Show all data from countries table.

SELECT \*

FROM countries;

2. Show all data from employees table.

SELECT\*

FROM employees;

3. Show all data from departments table.

4. Show all data from job\_history table.

LIKE / NOT LIKE

1. Show those employee details whose first name starts with the letter s.

SELECT \*

FROM employees

WHERE first\_name LIKE 's%';

2. Show those employee details whose first name doesn’t start with the letter s.

SELECT \*

FROM employees

WHERE first\_name NOT LIKE 's%';

3. Show those employee details whose first name ends with the letter a.

SELECT\*

FROM employees

WHERE first\_name LIKE '%a';

4. Show those employee details whose first name contains da as substring.

SELECT\*

FROM employees

WHERE first\_name LIKE 'da%';

5. Show those employee details whose first name starts with s and ends with a.

SELECT\*

FROM employees

WHERE first\_name LIKE 's%' AND first\_name LIKE'%a';

6. Show those employee details whose first name either starts with s or starts with m.

SELECT\*

FROM employees

WHERE first\_name LIKE 's%' OR first\_name LIKE 'm%';

7. Show those employee details whose first name contains the letter o and a.

SELECT\*

FROM employees

WHERE first\_name LIKE '%o%' AND first\_name LIKE'%a%';

8. Show those employee details whose first name contains the letter o followed by the letter a.

SELECT\*

FROM employees

WHERE first\_name LIKE '%o\_a%';

9. Show those employee details whose first name consists of exactly 3 characters.

SELECT\*

FROM employees

WHERE first\_name LIKE '\_\_\_';

10. Show those employee details whose first name consists of minimum 3 characters.

SELECT\*

FROM employees

WHERE first\_name LIKE '%\_\_\_%';

11. Show those employee details whose first name contains the letter a from the second last position.

SELECT\*

FROM employees

WHERE first\_name LIKE '\_a%';

IN() / NOT IN( ) / OR, ||

1. Show those country details whose country\_id is AU/BR/CN/JP.

SELECT\*

FROM countries

WHERE country\_id IN('AU','BR','CN','JP');

2. Show those department details whose manager\_id is not 204/100/145.

SELECT\*

FROM departments

WHERE manager\_id NOT IN('204','100','145');

3. Show those employee details whose job\_id is ST\_MAN/IT\_PROG.

SELECT \*

FROM employees

WHERE job\_id IN('ST\_MAN','IT\_PROG')

4. Show those employee details who does not work in the department\_id 100/30/90.

SELECT \*

FROM employees

WHERE department\_id NOT IN(100,30,90);

5. Show those location details where the postal\_code is either 2901/50090.

SELECT \*

FROM locations

WHERE postal\_code IN(2901,50090);

6. Show those location details where the city name is either Roma/Venice/Tokyo.

SELECT \*

FROM locations

WHERE city IN('Roma','Venice','Tokyo');

BETWEEN … AND… / NOT BETWEEN … AND … / AND, &&

1. Show those department details whose location\_id is within the range 1000 to 2000 inclusive.

SELECT \*

FROM departments

WHERE location\_id BETWEEN 1000 AND 2000;

2. Show those employee details whose salary is within the range 10000 to 20000 inclusive.

SELECT \*

FROM employees

WHERE salary BETWEEN 10000 AND 20000;

3. Show those employee details whose hire\_date is within the range ‘1987-01-01’ to ‘1987-06-30’ inclusive.

SELECT \*

FROM employees

WHERE hire\_date BETWEEN '1987-01-01' AND '1987-06-30';

4. Show those employee details whose department\_id is not within the range 50 to 60 inclusive.

SELECT \*

FROM employees

WHERE department\_id NOT BETWEEN 50 AND 60;

5. Show those job details where the difference between max\_salary and min \_salary is within the range 5000 to

10000 inclusive.

SELECT\*

FROM jobs

WHERE max\_salary-min\_salary BETWEEN 5000 AND 10000;

6. Show those job\_history details where the end\_date is within the range ‘1998-12-01’ to ‘1998-12-31’ inclusive.

SELECT\*

FROM job\_history

WHERE end\_date BETWEEN '1998-12-01' AND '1998-12-31';

CASE WHEN … WHEN … ELSE … END statement

1. Show all the past employees employee id, start date, job id and his group name from the job\_history table:

Determine the group name as below:

Start Date Group Name

on or before 1989-12-31 C

between 1994-12-31 to 1900-01-01 inclusive B

after 1995-01-01 inclusive A

ANS:

SELECT employee\_id,

start\_date,

job\_id,

CASE

WHEN end\_date<='1989-12-31' THEN 'C'

WHEN end\_date BETWEEN '1994-12-31' AND '1900-01-01' THEN 'B'

WHEN end\_date>'1995-01-01' THEN 'A'

END AS group\_name

FROM job\_history ;

2. Show the department id, department name, location id and location group name from the departments table:

Determine the location group name as below:

Location Id Location Group Name

less than 1200 C

between 2000 to 1200 inclusive B

greater than 2000 A

ANS:

Numerical and String Functions

1. Show all the employees employee id and their short name in lowercase format.

Short name format: first 3 letters from the first name followed by an underscore and then followed by the first 3

letters of the last name.

SELECT employee\_id,

CONCAT(LEFT(LOWER(first\_name),3), LEFT(LOWER(last\_name),3))

FROM employees;

2. Show all those employee details whose name is a palindrome.

SELECT \*

FROM employees

WHERE first\_name= REVERSE(first\_name);

3. Show all the employees employee id and email (i.e. add ‘@gmail.com’ at the end of each email).

SELECT employee\_id,

CONCAT(email, '@gmail.com')

FROM employees;

4. Show all the employees first name and phone number.

Phone number format: 515.xxx.xxx7 i.e. only show the first 4 characters and the last character and hide all the

intermediate characters with xxx.xxx

SELECT first\_name,

CONCAT(LEFT(phone\_number,3), '.xxx.xxx',RIGHT(phone\_number,1))

FROM employees;

5. Show all the employees employee id, email and full name.

Full name format: first\_name<SPACE>last\_name

Also show the full name in 20 characters if necessary right pad with necessary no of spaces.

SELECT employee\_id,

email,

CONCAT(first\_name,' ',last\_name) AS full\_name

FROM employees;

6. Show those location details from locations table whose postal code consists at most 5 characters and the first

two digits of its postal code is between 50 to 99 inclusive.

SELECT \*

FROM locations

WHERE (LENGTH(postal\_code)<=5 )

AND (LEFT(postal\_code,2) BETWEEN 50 AND 99);

7. Show all the employees employee id, first name and his salary in “10 thousand 5 hundred and 12 taka only”

format.

SELECT first\_name

employee\_id,

CONCAT(floor(salary/1000), 'thousand',

floor((salary%1000)/100) , 'hundred and',

floor((salary%100)) ,'taka only') AS 'new salary'

FROM employees;

8. For each job, show the job id, job title and how much greater the max\_salary from its min\_salary in percentage

format.

Note: Show the output in 2 decimal points

%greater=(max\_salary-min\_salary)\*100/min\_salary

SELECT job\_id,

job\_title,

CONCAT(floor(((max\_salary-min\_salary)\*100)/min\_salary),'%') AS 'percentage of diff'

FROM jobs;

9. Show all those job details from jobs table whose salary range (i.e. max\_salary-min\_salary) is greater than 8000

and the job title contains the word ‘Manager’.

SELECT \*

FROM jobs

WHERE (max\_salary- min\_salary)>8000 AND

job\_title LIKE '%manager%';

10. Show all the employees employee id, and his yearly total gross salary.

Note: Show the floor value of the total salary

yearly total salary = salary \* 12\* ( 1+(commission\_pct/100)) ]

SELECT employee\_id,

floor(salary \* 12\* ( 1+(commission\_pct/100))) AS 'Yearly\_salary'

FROM employees;

11. Show those department details from departments table whose tens digits of location id is within the range 5 to 9

inclusive.

SELECT\*

FROM departments

WHERE LENGTH(location\_id) BETWEEN '9' AND '5';

ORDER BY clause

1. Show all the employees first name, last name, email, hire date, salary in descending order of salary. If multiple

employees receive the same salary then also sort them based on the alphabetical order of their first name.

SELECT first\_name,

last\_name,

email,

hire\_date

FROM employees

ORDER BY salary DESC, first\_name ASC;

2. Show all the employees employee id and their join date in such a way that the senior most employee comes first.

If multiple employees have the same join date then also sort them based on the descending order of their

department id.

SELECT employee\_id,

hire\_date

FROM employees

ORDER BY hire\_date ASC, employee\_id DESC;

3. Show all the employees first name, email and phone number. Order the output based on the descending order of

first 3 digits of their phone number.

SELECT first\_name,

email,

phone\_number

FROM employees

ORDER BY LEFT(phone\_number,3)DESC;

4. Show all the employees employee id, email, hire year (only the year portion) and hire month (show the full

month name). Show the output from most recent hired employee to old employees.

SELECT

employee\_id,

email,

DATE\_FORMAT((hire\_date),'%Y'),

DATE\_FORMAT((hire\_date),'%M')

FROM employees

ORDER BY hire\_date DESC;

5. Show all the job\_history details in such a way that senior most employee data comes first and if multiple

employees have the same start date then also sort them based on the descending order of their end date.

SELECT \*

FROM job\_history

ORDER BY start\_date ASC, end\_date DESC;

6. Show all the jobs from jobs table where the highest salary range (i.e. max\_salary-min\_salary) job data comes

first.

SELECT \*

FROM jobs

ORDER BY max\_salary-min\_salary DESC;

DISTINCT clause

1. Show all the distinct manager\_ids from employees table.

SELECT DISTINCT manager\_id

FROM employees;

2. Show all the distinct job\_ids from the employees table.

3. Show all the distinct country\_ids from the locations table.

4. Show all the distinct job\_ids and department\_ids from employees table.

LIMIT clause

1. Show the highest salary holder employee details from the employees table.

SELECT\*

FROM employees

ORDER BY salary DESC

LIMIT 0,1;

2. Show the top 10 most experienced employee details from the employees table.

SELECT\*

FROM employees

ORDER BY hire\_date ASC

LIMIT 0,10;

3. Show the 2nd lowest salary range (i.e. max\_salary-min\_salary) job details from the jobs table.

SELECT\*

FROM jobs

ORDER BY max\_salary-min\_salary ASC

LIMIT 1,1;

4. Show the top 3 lowest salary holder employee details from department number 60.

SELECT\*

FROM employees

WHERE department\_id = 60

ORDER BY salary ASC

LIMIT 0,3;

5. Among the employees supervised by manager id 108, find out the 2nd highest salary holder employee details.

SELECT\*

FROM employees

WHERE manager\_id = 108

ORDER BY salary DESC

LIMIT 1,1;

6. Among the employees whose job type is ‘ST\_CLERK’, show the highest experienced employee id from the

job\_history table.

SELECT employee\_id

FROM job\_history

WHERE job\_id LIKE 'ST\_CLERK'

ORDER BY start\_date ASC

LIMIT 1,1;

Aggregate Operations (GROUP BY, HAVING clauses)

1. Show the total no of employees, their total salary, their average salary, their maximum salary, their minimum

salary from employees table.

SELECT count(employee\_id),sum(salary),avg(salary),max(salary),min(salary)

FROM employees;

2. Show the maximum and minimum experienced employees hire dates from employees table

SELECT MAX(CURDATE()-hire\_date),MIN(CURDATE()-hire\_date)

FROM employees;

3. Show the maximum experienced employee hire date working in department number 50 from employees table.

SELECT MAX(hire\_date)

FROM employees

WHERE department\_id = 50;

4. Show the number of departments located in location id 1700 from departments table.

SELECT COUNT(department\_id)

FROM departments

WHERE location\_id=1700;

5. Show the most recent retired employee’s end date working in department number 80 from job history table.

SELECT MAX(end\_date)

FROM job\_history

WHERE department\_id= 80;

6. Show the maximum and minimum value of min\_salary column, maximum and minimum value of max\_salary

column from jobs table.

SELECT

MAX(min\_salary) AS max\_min\_salary,

MIN(min\_salary) AS min\_min\_salary,

MAX(max\_salary) AS max\_max\_salary,

MIN(max\_salary) AS min\_max\_salary

FROM

jobs;

7. Count the number of employees managed by manager id 114 from employees table.

SELECT

COUNT(\*) AS num\_employees\_managed

FROM

employees

WHERE

manager\_id = 114;

8. Count the total number of distinct job\_ids from employees table

SELECT COUNT(DISTINCT job\_id)

FROM employees ;

9. Count the distinct number of countries from locations table.

10. Count the total number of locations located in ‘US’ from locations table.

SELECT COUNT(country\_id)

FROM locations

WHERE country\_id='US';

11. Show the maximum and minimum salary range value (i.e. salary range = max\_salary - min\_salary) from jobs table.

SELECT

MAX(max\_salary - min\_salary) AS max\_salary\_range,

MIN(max\_salary - min\_salary) AS min\_salary\_range

FROM

jobs;

12. Count the number of employees whose employee id is greater than his manager id.

SELECT

COUNT(\*) AS num\_employees

FROM

employees

WHERE

employee\_id > manager\_id;

1. Show each region\_id and corresponding no of countries in that region from countries table.

SELECT region\_id, COUNT(country\_id)

FROM countries

GROUP BY region\_id;

2. Show the location\_id and corresponding no of departments in that location from departments table.

3. For each department\_id, show the no of employees in that department from employees table.

4. For each manager\_id, show the no of employees under his supervision from employees table.

5. For each job\_id, show the no of employees in that job type from employees table.

6. For each department\_id, show the no of managers from that department using employees table.

7. Count the total number of employees joined in the even month and total number of employees joined in the odd

number months from the employees table.

----------------

2. Show the location\_id and corresponding number of departments in that location from the departments table:

SELECT

location\_id,

COUNT(\*) AS num\_departments

FROM

departments

GROUP BY

location\_id;

3. For each department\_id, show the number of employees in that department from the employees table:

SELECT

department\_id,

COUNT(\*) AS num\_employees

FROM

employees

GROUP BY

department\_id;

4. For each manager\_id, show the number of employees under his supervision from the employees table:

SELECT

manager\_id,

COUNT(\*) AS num\_employees

FROM

employees

GROUP BY

manager\_id;

5. For each job\_id, show the number of employees in that job type from the employees table:

SELECT

job\_id,

COUNT(\*) AS num\_employees

FROM

employees

GROUP BY

job\_id;

6. For each department\_id, show the number of managers from that department using the employees table:

SELECT

department\_id,

COUNT(\*) AS num\_managers

FROM

employees

WHERE

job\_id = 'MANAGER'

GROUP BY

department\_id;

7. Count the total number of employees joined in the even month and the total number of employees joined in the odd number months from the employees table:

SELECT

CASE WHEN MONTH(hire\_date) % 2 = 0 THEN 'Even' ELSE 'Odd' END AS month\_type,

COUNT(\*) AS num\_employees

FROM

employees

GROUP BY

month\_type;

----------------

8. Show the department wise total no of employees, maximum and minimum salary in that department, average

and total salary provided by that department from the employees table.

9. For each year, show the total no of employees who were hired during that year from the employees table.

SELECT count(employee\_id)

FROM employees

GROUP BY year(hire\_date);

10. Show the total no of jobs within 0k to 10k, 10k to 20k and so on salary ranges(max\_salary-min\_salary) groups

from the jobs table.

11. For each country\_id, show the total no of locations in that country from the locations table.

12. For each city, show the total no of locations in that city from the locations table.

13. Group and count employees based on the first letter of their names. (max 26 groups as 26 alphabets)

SELECT first\_name,count(employee\_id)

FROM employees

GROUP BY SUBSTR(first\_name,1,1) ;

14. For each job\_id and each department, show the total no of employees in that group from the employees table.

15. For each year and each month, show the total no of employees who have left their jobs from the job\_history

table.

---------

Show only those department\_ids where the total salary expense is more than $100,000.

sql

Copy code

SELECT

department\_id

FROM

employees

GROUP BY

department\_id

HAVING

SUM(salary) > 100000;

Count the total number of employees for each department, excluding employees with job\_id "AD\_PRESS" and considering only departments with more than 5 employees.

sql

Copy code

SELECT

department\_id,

COUNT(\*) AS total\_employees

FROM

employees

WHERE

job\_id != 'AD\_PRESS'

GROUP BY

department\_id

HAVING

total\_employees > 5;

Group employees based on the first 3 digits of their phone number, excluding employees from departments 10, 20, and 60, and excluding groups where the total salaries of employees are less than $50,000.

sql

Copy code

SELECT

SUBSTRING(phone\_number, 1, 3) AS phone\_prefix,

COUNT(\*) AS total\_employees,

SUM(salary) AS total\_salary

FROM

employees

WHERE

department\_id NOT IN (10, 20, 60)

GROUP BY

phone\_prefix

HAVING

total\_salary > 50000;

For each year and each month, count the total number of employees joined, excluding years and months where the total number of hired employees is less than 20.

sql

Copy code

SELECT

YEAR(hire\_date) AS hire\_year,

MONTH(hire\_date) AS hire\_month,

COUNT(\*) AS total\_employees\_joined

FROM

employees

GROUP BY

hire\_year,

hire\_month

HAVING

total\_employees\_joined >= 20;

For each country and each city, count the total number of locations, excluding locations from the city 'US' and countries and cities with less than 5 locations.

sql

Copy code

SELECT

country\_name,

city,

COUNT(\*) AS total\_locations

FROM

locations

WHERE

city != 'US'

GROUP BY

country\_name,

city

HAVING

total\_locations >= 5;

-----------

1. Show only those manager\_ids who handle more than 5 employees.

SELECT manager\_id

FROM employees

GROUP BY manager\_id

HAVING COUNT(employee\_id)>5;

2. Show only those department\_ids where in total salary expense is more than 100000 dollar.

3. Count the total no of employees for each department. Don’t consider employees of job\_id “AD\_PRESS” and also

consider only those departments where total no of employees is greater than 5.

4. Group employees based on the first 3 digit of their phone number. Avoid employees from department no

10/20/60 and also avoid those groups where total salaries of employees is less than 50000.

5. For each year and each month, count total number of employees joined from employees table. Don’t consider

those year and months where total number of hired employees are less than 20.

6. For each country and each city, count total number of locations from locations table. Don’t consider locations

from city ‘US’ and also don’t consider those country and city having less than 5 locations.

Table Join Operations (JOIN, LEFT JOIN clauses)

1. Show the region\_name and corresponding country\_name

SELECT c.country\_name,r.region\_name

FROM countries AS c

JOIN

regions AS r

ON c.region\_id=r.region\_id;

2. Show the department\_name and corresponding country\_name.

3. Show the employee\_name and his job place country\_name.

4. Show the employee\_name and his job\_title.

5. Show the employee\_name and his manager\_name

6. Show the department\_name and the manager\_name of corresponding department.

7. Show the employee\_id, his salary, his manager\_id, his manager\_name, his manager\_salary.

8. Show the employee\_id, his join\_date, his manager\_id, his manager\_name, his manager\_salary.

SELECT e.employee\_id,e.hire\_date,e.manager\_id,m.first\_name,m.salary

FROM employees AS e

JOIN employees AS m

ON e.employee\_id=m.manager\_id;

9. Show the manger\_name and his manager\_name (manager of manager).

SELECT m.first\_name,mm.first\_name

FROM employees AS e

JOIN employees AS m

ON e.employee\_id=m.employee\_id

JOIN employees AS mm

ON m.employee\_id= mm.employee\_id;

10. Show the employee name and his manager name only for those employees who have joined after this manager.

11. Show the employees name and other employees name who receives higher salary than him

12. Show the employees name and other employees name who is hired after him.

13. For each region, show the region\_name and total no of employees in that region.

-------------

Show the employee name and his manager name only for those employees who have joined after their manager.

sql

Copy code

SELECT

e1.employee\_name AS employee\_name,

e2.employee\_name AS manager\_name

FROM

employees e1

JOIN

employees e2 ON e1.manager\_id = e2.employee\_id

WHERE

e1.join\_date > e2.join\_date;

Show the employees name and other employees name who receive a higher salary than them.

sql

Copy code

SELECT

e1.employee\_name AS employee\_name,

e2.employee\_name AS other\_employee\_name

FROM

employees e1

JOIN

employees e2 ON e1.salary < e2.salary;

Show the employees name and other employees name who were hired after them.

sql

Copy code

SELECT

e1.employee\_name AS employee\_name,

e2.employee\_name AS other\_employee\_name

FROM

employees e1

JOIN

employees e2 ON e1.join\_date < e2.join\_date;

For each region, show the region\_name and total number of employees in that region.

sql

Copy code

SELECT

r.region\_name,

COUNT(\*) AS total\_employees

FROM

employees e

JOIN

departments d ON e.department\_id = d.department\_id

JOIN

locations l ON d.location\_id = l.location\_id

JOIN

regions r ON l.region\_id = r.region\_id

GROUP BY

r.region\_name;

-------

14. For each job, show the job\_title and total no of employees.

SELECT j.job\_title,COUNT(e.employee\_id)

FROM employees AS e

JOIN jobs AS j

ON e.job\_id=j.job\_id

GROUP BY j.job\_title;

15. For each country, show the total no of departments in that country.

16. For each department, show the department\_name and corresponding total no of ex-employees (job\_history

table) from that department.

17. For each manager, show the manager\_name and total no of employees under his supervision.

18. For each manager, show the manager\_name and total no of employees under his supervision who receives

higher salary than him.

---------

15. For each country, show the total number of departments in that country.

sql

Copy code

SELECT l.country\_name, COUNT(d.department\_id) AS total\_departments

FROM locations l

JOIN departments d ON l.location\_id = d.location\_id

GROUP BY l.country\_name;

16. For each department, show the department\_name and corresponding total number of ex-employees (job\_history table) from that department.

sql

Copy code

SELECT d.department\_name, COUNT(jh.employee\_id) AS total\_ex\_employees

FROM departments d

LEFT JOIN job\_history jh ON d.department\_id = jh.department\_id

GROUP BY d.department\_name;

17. For each manager, show the manager\_name and total number of employees under his supervision.

sql

Copy code

SELECT CONCAT(m.first\_name, ' ', m.last\_name) AS manager\_name, COUNT(e.employee\_id) AS total\_employees

FROM employees e

JOIN employees m ON e.manager\_id = m.employee\_id

GROUP BY m.employee\_id;

18. For each manager, show the manager\_name and total number of employees under his supervision who receive a higher salary than him.

sql

Copy code

SELECT CONCAT(m.first\_name, ' ', m.last\_name) AS manager\_name,

COUNT(e.employee\_id) AS total\_higher\_salary\_employees

FROM employees e

JOIN employees m ON e.manager\_id = m.employee\_id

WHERE e.salary > m.salary

GROUP BY m.employee\_id;

--------

SELECT m.first\_name, count(e.employee\_id)

FROM employees AS e

JOIN employees AS m

ON e.employee\_id=m.manager\_id

GROUP BY m.manager\_id

HAVING e.salary>m.salary;

19. Show the employee name and no of employees who receives lower salary than him.

20. Show the employee name and no of employees who is hired before him.

------

19. Show the employee name and number of employees who receive a lower salary than him.

sql

Copy code

SELECT CONCAT(e1.first\_name, ' ', e1.last\_name) AS employee\_name,

COUNT(e2.employee\_id) AS num\_employees\_lower\_salary

FROM employees e1

JOIN employees e2 ON e1.salary < e2.salary

GROUP BY e1.employee\_id;

20. Show the employee name and number of employees who were hired before him.

sql

Copy code

SELECT CONCAT(e1.first\_name, ' ', e1.last\_name) AS employee\_name,

COUNT(e2.employee\_id) AS num\_employees\_hired\_before

FROM employees e1

JOIN employees e2 ON e1.hire\_date > e2.hire\_date

GROUP BY e1.employee\_id;

--------

**Subquery**

* **Retrieve the first name, last name, and email of all employees.**

**SELECT first\_name, last\_name, email FROM employees;**

* **List the job titles and salaries of all jobs.**

**SELECT job\_title, salary FROM jobs;**

* **Display the country names and their corresponding region names.**

**SELECT country\_name, region\_name FROM countries**

**JOIN regions ON countries.region\_id = regions.region\_id;**

* **Find the number of employees in each department.**

**SELECT department\_id, COUNT(employee\_id) AS num\_employees FROM employees**

**GROUP BY department\_id;**

* **Show the street address, city, and postal code of all locations.**

**SELECT street\_address, city, postal\_code**

**FROM locations;**

* **Get the total number of employees in the company.**

**SELECT COUNT(employee\_id) AS total\_employees**

**FROM employees;**

* **List all employees hired after January 1, 2020.**

**SELECT \* FROM employees**

**WHERE hire\_date > '2020-01-01';**

* **Display the department names and the number of employees in each department.**

**SELECT departments.department\_name, COUNT(employees.employee\_id) AS num\_employees**

**FROM departments**

**LEFT JOIN employees ON departments.department\_id = employees.department\_id**

**GROUP BY departments.department\_id;**

* **Find the average salary of all employees.**

**SELECT AVG(salary) AS average\_salary FROM employees;**

* **Show the job titles and the maximum salary for each job.**

**SELECT job\_title, MAX(salary) AS max\_salary FROM jobs**

**GROUP BY job\_title;**

* **Retrieve the employee ID, first name, last name, and salary of employees earning more than $5000.**

**SELECT employee\_id, first\_name, last\_name, salary FROM employees**

**WHERE salary > 5000;**

* **List the department names and their respective manager IDs.**

**SELECT department\_name, manager\_id FROM departments;**

* **Get the employee IDs, first names, and last names of employees with no manager.**

**SELECT employee\_id, first\_name, last\_name FROM employees**

**WHERE manager\_id IS NULL;**

* **Show the job titles and their corresponding minimum salary.**

**SELECT job\_title, MIN(salary) AS min\_salary FROM jobs**

**GROUP BY job\_title;**

* **Find the employee with the highest salary.**

**SELECT \* FROM employees**

**WHERE salary = (SELECT MAX(salary) FROM employees);**

* **Retrieve the employee IDs, first names, last names, and commission percentages of employees with a commission percentage.**

**SELECT employee\_id, first\_name, last\_name, commission\_pct FROM employees**

**WHERE commission\_pct IS NOT NULL;**

* **List the department IDs and names where the department has no manager.**

**SELECT department\_id, department\_name FROM departments**

**WHERE manager\_id IS NULL;**

* **Display the job titles and the difference between the maximum and minimum salaries for each job.**

**SELECT job\_title, (MAX(salary) - MIN(salary)) AS salary\_difference FROM jobs**

**GROUP BY job\_title;**

* **Find the employee with the longest tenure in the company.**

**SELECT \* FROM employees**

**WHERE hire\_date = (SELECT MIN(hire\_date) FROM employees);**

* **Show the job IDs and titles of jobs with no minimum or maximum salary specified.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE min\_salary IS NULL OR max\_salary IS NULL;**

* **List the department IDs and names along with the city and country of their corresponding locations.**

**SELECT departments.department\_id, departments.department\_name, locations.city, countries.country\_name**

**FROM departments**

**JOIN locations ON departments.location\_id = locations.location\_id**

**JOIN countries ON locations.country\_id = countries.country\_id;**

* **Retrieve the employee IDs and names of employees who do not belong to any department.**

**SELECT employee\_id, first\_name, last\_name FROM employees**

**WHERE department\_id IS NULL;**

* **Show the job IDs and titles of jobs where the title contains 'Manager'.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE job\_title LIKE '%Manager%';**

* **Find the department with the highest average salary.**

**SELECT department\_id, AVG(salary) AS avg\_salary FROM employees**

**GROUP BY department\_id**

**ORDER BY avg\_salary DESC**

**LIMIT 1;**

* **Display the employee IDs, first names, last names, and hire dates of employees hired in 2022.**

**SELECT employee\_id, first\_name, last\_name, hire\_date FROM employees**

**WHERE hire\_date BETWEEN '2022-01-01' AND '2022-12-31';**

* **List the job titles and the number of employees in each job.**

**SELECT job\_title, COUNT(employee\_id) AS num\_employees FROM employees**

**GROUP BY job\_title;**

* **Get the employee IDs and names of employees who started after their managers.**

**SELECT e.employee\_id, e.first\_name, e.last\_name**

**FROM employees e**

**JOIN employees m ON e.manager\_id = m.employee\_id**

**WHERE e.hire\_date > m.hire\_date;**

* **Show the department names and the total salary of employees in each department.**

**SELECT departments.department\_name, SUM(employees.salary) AS total\_salary**

**FROM departments**

**JOIN employees ON departments.department\_id = employees.department\_id**

**GROUP BY departments.department\_name;**

* **Retrieve the employee IDs and names of employees whose first name starts with 'J'.**

**SELECT employee\_id, first\_name, last\_name FROM employees**

**WHERE first\_name LIKE 'J%';**

* **Display the job IDs and titles of jobs where the minimum salary is less than the average salary of all jobs.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE min\_salary < (SELECT AVG(salary) FROM jobs);**

* **Find the department with the lowest total salary.**

**SELECT department\_id, SUM(salary) AS total\_salary FROM employees**

**GROUP BY department\_id**

**ORDER BY total\_salary ASC**

**LIMIT 1;**

* **List the job IDs and titles of jobs where the title ends with 'Analyst'.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE job\_title LIKE '%Analyst';**

* **Show the employee IDs, first names, last names, and hire dates of employees hired before their managers.**

**SELECT e.employee\_id, e.first\_name, e.last\_name, e.hire\_date**

**FROM employees e**

**JOIN employees m ON e.manager\_id = m.employee\_id**

**WHERE e.hire\_date < m.hire\_date;**

* **Retrieve the employee IDs, first names, last names, and hire dates of employees hired in 2020 or 2021.**

**SELECT employee\_id, first\_name, last\_name, hire\_date FROM employees**

**WHERE hire\_date BETWEEN '2020-01-01' AND '2021-12-31';**

* **Display the job IDs and titles of jobs where the maximum salary is greater than the average salary of all jobs.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE max\_salary > (SELECT AVG(salary) FROM jobs);**

* **Find the department with the highest number of employees.**

**SELECT department\_id, COUNT(employee\_id) AS num\_employees FROM employees**

**GROUP BY department\_id**

**ORDER BY num\_employees DESC**

**LIMIT 1;**

* **List the job IDs and titles of jobs where the minimum salary is not equal to the maximum salary.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE min\_salary != max\_salary;**

* **Show the department names and the total number of employees in each department where the number of employees is greater than 5.**

**SELECT departments.department\_name, COUNT(employees.employee\_id) AS num\_employees**

**FROM departments**

**LEFT JOIN employees ON departments.department\_id = employees.department\_id**

**GROUP BY departments.department\_id**

**HAVING num\_employees > 5;**

* **Retrieve the employee IDs, first names, last names, and salaries of employees whose salary is not within the range $3000 to $6000.**

**SELECT employee\_id, first\_name, last\_name, salary FROM employees**

**WHERE salary NOT BETWEEN 3000 AND 6000;**

* **Display the job IDs and titles of jobs where the difference between the minimum and maximum salary is greater than $5000.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE (max\_salary - min\_salary) > 5000;**

* **Find the department with the highest total commission.**

**SELECT department\_id, SUM(commission\_pct) AS total\_commission FROM employees**

**GROUP BY department\_id**

**ORDER BY total\_commission DESC**

**LIMIT 1;**

* **List the job IDs and titles of jobs where the title contains either 'Engineer' or 'Developer'.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE job\_title LIKE '%Engineer%' OR job\_title LIKE '%Developer%';**

* **Show the department names and the average salary of employees in each department where the average salary is less than $7000.**

**SELECT departments.department\_name, AVG(employees.salary) AS avg\_salary**

**FROM departments**

**JOIN employees ON departments.department\_id = employees.department\_id**

**GROUP BY departments.department\_id**

**HAVING avg\_salary < 7000;**

* **Retrieve the employee IDs, first names, last names, and salaries of employees whose salary is the maximum salary in their department.**

**SELECT e.employee\_id, e.first\_name, e.last\_name, e.salary**

**FROM employees e**

**JOIN (**

**SELECT department\_id, MAX(salary) AS max\_salary**

**FROM employees**

**GROUP BY department\_id**

**) max\_salaries ON e.department\_id = max\_salaries.department\_id AND e.salary = max\_salaries.max\_salary;**

* **Display the job IDs and titles of jobs where the maximum salary is less than the minimum salary of any other job.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE max\_salary < ANY (SELECT min\_salary FROM jobs WHERE min\_salary IS NOT NULL);**

* **Find the department with the highest number of managers.**

**SELECT department\_id, COUNT(manager\_id) AS num\_managers FROM employees**

**WHERE manager\_id IS NOT NULL**

**GROUP BY department\_id**

**ORDER BY num\_managers DESC**

**LIMIT 1;**

* **List the job IDs and titles of jobs where the minimum salary is not specified but the maximum salary is.**

**SELECT job\_id, job\_title FROM jobs**

**WHERE min\_salary IS NULL AND max\_salary IS NOT NULL;**

* **Show the department names and the average commission percentage of employees in each department where the average commission percentage is greater than 0.**

**SELECT departments.department\_name, AVG(employees.commission\_pct) AS avg\_commission**

**FROM departments**

**JOIN employees ON departments.department\_id = employees.department\_id**

**GROUP BY departments.department\_id**

**HAVING avg\_commission > 0;**

* **Retrieve the employee IDs, first names, last names, and hire dates of employees who were hired after their managers but before January 1, 2023.**

**SELECT e.employee\_id, e.first\_name, e.last\_name, e.hire\_date**

**FROM employees e**

**JOIN employees m ON e.manager\_id = m.employee\_id**

**WHERE e.hire\_date > m.hire\_date AND e.hire\_date < '2023-01-01';**

* **Display the job IDs and titles of jobs where the minimum salary is greater than the maximum salary of any other job.**

**SELECT job\_id, job\_title FROM jobs j**

**WHERE min\_salary > ALL (SELECT max\_salary FROM jobs WHERE max\_salary IS NOT NULL);**

**View**