11/01/2020

**Select Clause with Where clause**

1) Display details of jobs where the minimum salary is greater than 10000.

**Ans**: select employee\_id, first\_name, last\_name, email, phone\_int, hire\_Date, job\_id, salary, commission\_pct, manager\_id, department\_id from employees where salary>10000;

2) Display the first name and join date of the employees who joined between 2002 and 2005.

**Ans:** select first\_name, last\_name, hire\_date from employees where year(hire\_date) between 2002 and 2005 order by hire\_date;

3) Display first name and join date of the employees who is either IT Programmer or Sales Man.

**Ans:** select first\_name, hire\_date from employees where job\_id in('IT\_PROG', 'ST\_MAN');

4) Display first name, salary, commission pct, and hire date for employees with salary less than 10000.

**Ans:** select first\_name, hire\_date,salary, commission\_pct from employees where salary<10000;

5) Display job Title, the difference between minimum and maximum salaries for jobs with max salary in the range 10000 to 20000.

**Ans**: select job\_title, max\_salary-min\_salary as salary from jobs where max\_salary between 10000 and 20000;

6) Display employees where the first name or last name starts with S.

**Ans:** select \*from employees where first\_name like 's%' or last\_name like 's%';

7) Display details of jobs in the descending order of the title.

**Ans:** select \*from jobs order by job\_title desc;

8) Display employees who joined in the month of May.

**Ans:** select \*from employees where month(hire\_date)='MAY';

9) Display details of the employees where commission percentage is null and salary in the range 5000 to 10000 and department is 30.

**Ans**: select \*from employees where commission\_pct is null and salary between 5000 and 10000 and department\_id=30;

**Join**

1). Display job title, employee ID, number of days between ending date and starting date for all jobs in department 30 from job history.

**Ans:** select employee\_id, job\_title, end\_date-start\_date Days from job\_history natural join jobs where department\_id=30;

2) Display department name and manager first name.

**Ans:** select dept.department\_id, dept.department\_name, dept.manager\_id, emp.first\_name from departments dept inner join employees emp on (dept.manager\_id = emp.employee\_id);

3) Display department name, manager name, and city.

**Ans:** select dept.department\_name, emp.first\_name,l.city from departments using(location\_id);from departments dept inner join employees emp on (dept.manager\_id = emp.employee\_id);

4) Display country name, city, and department name.

**Ans:** select country\_name,city,department\_name from countries join locations using (country\_id)join departments using(location\_id);

5) Display employee name and country in which he is working.

**Ans**: select first\_name, country\_name from employees join departments using(department\_id) join locations using(location\_id)join countries using(country\_id);

**Functions**

1. Display employees who joined in the month of May.

**Ans**: select first\_name from employees where month(hire\_date) like '5';

2. Display first name, salary, and round the salary to thousands.

**Ans:** select first\_name, salary,round(salary) as TotalSalary from employees;

3. Display first name and date of first salary of the employees.

**Ans:** select first\_name, last\_day(hire\_date)from employees;

4. Display first name and experience of the employees.

**Ans:** select first\_name, datediff(sysdate(),hire\_date)/365 from employees;

5. Display the length of first name for employees where last name contain character ‘b’ after 3rd position.

**Ans:** select first\_name, last\_name from employees where instr(last\_name,'b')>3;

6. Display first name in upper case and email address in lower case for employees where the first name and email address are same irrespective of the case.

**Ans**: select upper(first\_name), lower(email) from employees;

7. Display employees who joined in the current year.

**Ans:** select \*from employees where year(hire\_date) = '2021';

8. Display the number of days between system date and 1st January 1995.

9. Display how many employees joined in each month of the current year.

**MySQL Aggregate function**

1. Display employee ID and the date on which he ended his previous job.

**Ans:** select employee\_id, max(end\_date)from job\_history where employee\_id in(select employee\_id from job\_history group by 1 having count(employee\_id)>1)group by 1;

2. Display number of employees joined after 15th of the month.

**Ans**: select count(employee\_id) from employees where date(hire\_date)>15;

select with Group by

3. Display the country ID and number of cities we have in the country.

**Ans:** select country\_id, count(\*)from locations group by country\_id;

4. Display average salary of employees in each department who have commission percentage.

**Ans:** select department\_id, avg(salary)from employees where commission\_pct group by department\_id;

5. Display job ID, number of employees, sum of salary, and difference between highest salary and lowest salary of the employees of the job.

**Ans**: select employee\_id, max(end\_date)from job\_history where employee\_id in(select employee\_id from job\_history group by 1 having count(employee\_id)>1)group by 1;

6. Display job ID for jobs with average salary more than 10000.

**Ans**: select job\_id, avg(salary) from employees group by job\_id having avg(salary)>10000;

7. Display years in which more than 10 employees joined.

**Ans**: select date(hire\_date)from employees group by date(hire\_date) having count(employee\_id)>10;

8. Display departments in which more than five employees have commission percentage.

**Ans:** select department\_id from employees where commission\_pct group by department\_id having count(commission\_pct)>5;

9. Display department name and number of employees in the department**.**

**Ans**: select department\_name, count(\*) from departments inner join employees on employees.department\_id = departments.department\_id group by departments.department\_id,department\_name order by department\_name;

10. Display employee ID for employees who did more than one job in the past.

**Ans:** select employee\_id from job\_history group by employee\_id having count(\*)>=1;

**12/01/2020**

**DML Operation**

1. Change salary of employee 115 to 8000 if the existing salary is less than 6000.

**Ans:** update employees set salary = 8000 where employee\_id=115 and salary<6000;

desc employees;

2. Insert a new employee into employees with all the required details.

**Ans**: insert into employees(employee\_id,first\_name,last\_name, email,phone\_int,hire\_date,job\_id,salary,department\_id)values(222,'xxx','yyy','zzz','111 55 542',2018-08-02,'IT\_PROG',20000,50);

3. Delete department 20.

4. Change job ID of employee 110 to IT\_PROG if the employee belongs to department 10 and the existing job ID does not start with IT.

**Ans:** update employees set job\_id='IT\_PROG' where employee\_id=110 and department\_id=10 and not job\_id like 'IT%';

desc employees;

5. Insert a row into departments table with manager ID 120 and location ID in any location ID for city Tokyo.

6. Display job title, employee ID, number of days between ending date and starting date for all jobs in department 30 from job history.

**Ans:** select employee\_id, job\_title, end\_date=start\_date days from job\_history natural join jobs where department\_id=30;

**DDL Assignments with Constraints**

1)Table ---> Customer

custId, firstName,lastName,age,city, mobileNumber, dob

Add the Constraints

custId is Primary Key

firstName not null

age must be greater than 21

mobile must be unique

**Ans:** create table Customer(custid int,firstName varchar(20) not null,lastName varchar(20),age int,city varchar(20),mobileNumber int,dob date,constraint c\_pk primary key(custid),constraint a\_pk check(age>21),constraint p\_pk unique(mobileNumber));

2)Table ----> Branch

branchId, branchName, city

Add the Constraints

branchId is Primary Key

branchName not null

city not null

**Ans:** create table Branch(branchId int,branchName varchar(20) not null,city varchar(20) not null,constraint b\_pk primary key(branchId));

Table -----> Account

accountNumber, openingBalance, typeOfAccount, status,BankId,CustId

Add the Constraints

accountNumber is primary key

openingBalance must be greater than 5000

typeOfAccount must be saving/current

BankId is foreign key refer to BranchId(Primary key) Branch table

CustId is foreign key refer to Customer(Primary key) Customer table

**Ans**: Create table Account(accountNumber varchar(20),openingBalance int,typeOfAccount enum('savings','current'),status varchar(20),bankid int,constraint a\_pk primary key(accountNumber),constraint ob\_pk check(openingBalance>5000),constraint bi\_pk foreign key(bankid) references Branch(branchId));

Table ----> Transaction

transactionId, transactionDate, MediumOfTransaction, TransactionAmount

Add the Constraints

transactionId is primary key

**Ans:** Create table Account(accountNumber varchar(20),openingBalance int,typeOfAccount enum('savings','current'),status varchar(20),bankid int,constraint a\_pk primary key(accountNumber),constraint ob\_pk check(openingBalance>5000),constraint b\_pk foreign key(bankid) references Branch(branchId));

Table ----> Loan

LoanId, loanAmount, customerId and bankdId

Add the Constraints

loadId is primary key

loanAmount must be +ve

BankId is foreign key refer to BranchId(Primary key) Branch table

**Ans:** Create table Loan(LoanId int,loanAmount int,customerId int,bankid int,constraint li\_pk primary key(LoanId),constraint la\_pk check(loanAmount>=0),constraint bi\_pk foreign key(bankid) references Branch(branchId));

**Sub Query**

1. Display details of departments managed by ‘John’.

**Ans:** select \* from departments where manager\_id in (select employee\_id from employees where first\_name='john');

2. Display employees who did not do any job in the past.

**Ans**: select \* from employees where employee\_id not in (select employee\_id from job\_history);

3. Display job title and average salary for employees who did a job in the past.

**Ans:** select job\_title, avg(salary) from employees natural join jobs group by job\_title;

4. Display country name, city, and number of departments where department has more than 5 employees.

**Ans** :select country\_name, city, count(department\_id) from countries join locations using(country\_id) join departments using(location\_id) where department\_id in (select department\_id from employees group by department\_id having count(department\_id)>=2)group by country\_name,city;

5. Display details of manager who manages more than 5 employees.

**Ans:** select distinct department\_id from employees group by department\_id, manager\_id having c

6. Display details of current job for employees who worked as IT Programmers in the past.

**Ans:** select \*from jobs where job\_id in(select job\_id from employees where employee\_id in(select employee\_id from job\_history where job\_id='IT\_PROG'));

7. Display the details of employees drawing the highest salary in the department.

**Ans:** select \*from employees where salary in( select max(salary) from employees group by department\_id);

8. Display third highest salary of all employees

**Ans:** select salary from(select salary from employees order by salary desc limit 3)as comp order by salary limit 1;