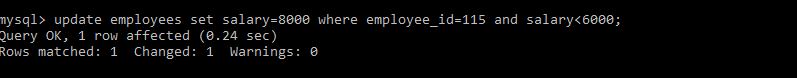
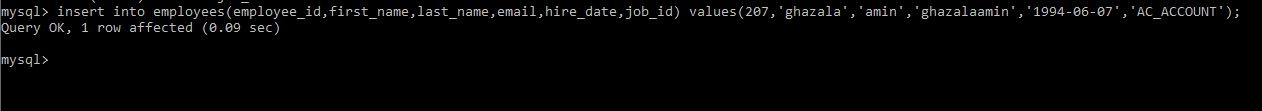
**12/01/2020**

**DML Operation**

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

****

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

****

**3. Delete department 20.**

**dml(3)**

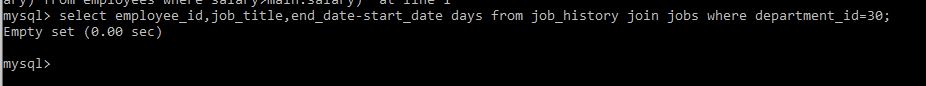
**4. Change job ID of employee 110 to IT\_PROG if the employee belongs to department 10 and the existing job ID does not start with IT.**

**dml(4)**

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

**dml(5)**

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

****

**DDL Assignments with Constraints**

**Table ---> Customer**

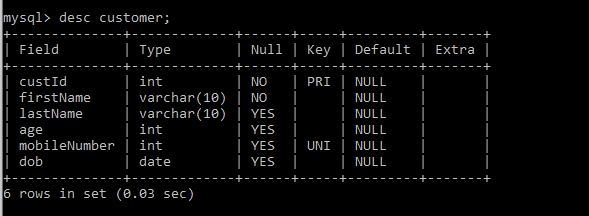
**custId, firstName,lastName,age,city, mobileNumber, dob**

**Add the Constraints**

**custId is Primary Key**

**firstName not null**

**mobile must be unique.**

****

**Table ----> Branch**

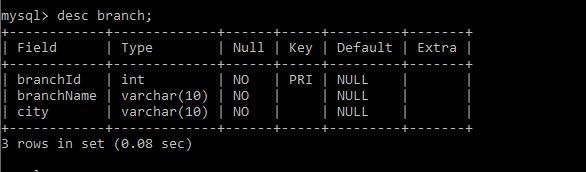
**branchId, branchName, city**

**Add the Constraints**

**branchId is Primary Key**

**branchName not null**

**city not null**

****

**Table -----> Account**

**accountNumber, openingBalance, typeOfAccount, status,BankId,CustId**

**Add the Constraints**

**accountNumber is primary key**

**typeOfAccount must be saving/current**

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

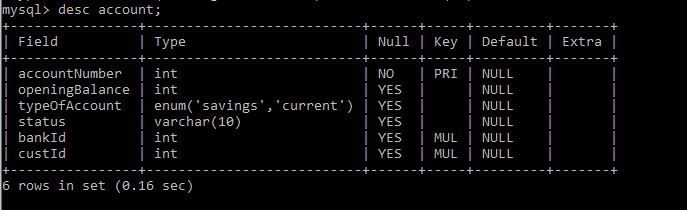
**CustId is foreign key refer to Customer(Primary key) Customer table**

**Table ----> Transaction**

**transactionId, transactionDate, MediumOfTransaction, TransactionAmount**

**Add the Constraints**

**transactionId is primary key**

****

**Table ----> Loan**

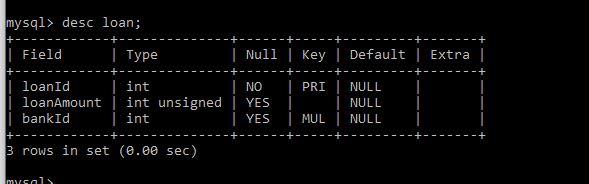
**LoanId, loanAmount, customerId and bankdId**

**Add the Constraints**

**loadId is primary key**

**loanAmount must be +ve**

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

****

**Sub Query**

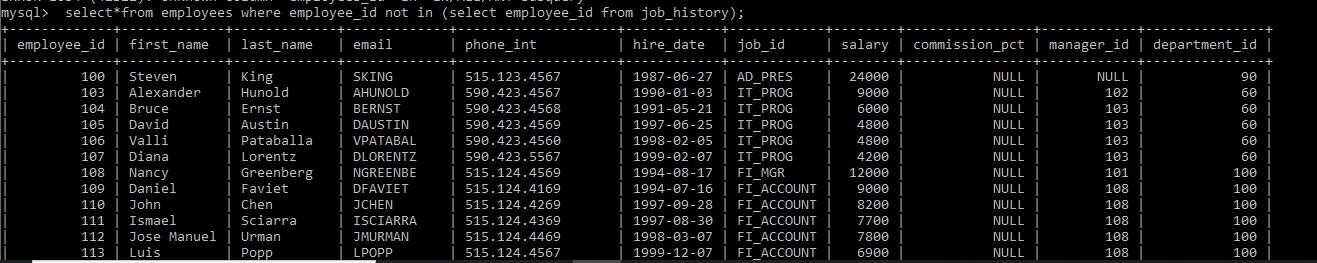
**1. Display details of departments managed by ‘John’.**

**select\*from departments where manager\_name in(select manager\_name from managers where manager\_name like 'john';**

**sub(1)**

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

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

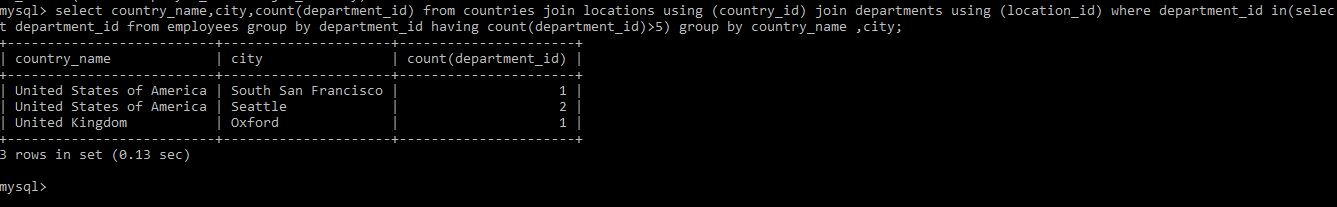
****

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

**select job\_title,avg(salary) from jobs join employees group by job\_title where employee\_id in (select employee\_id from job\_history);**

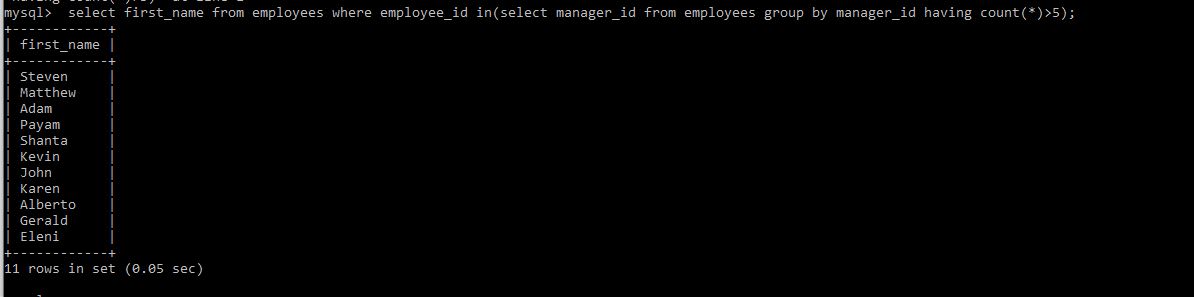
1. **Display country name, city, and number of departments where department has more than 5 employees.**

**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)>5) group by country\_name ,city;**

****

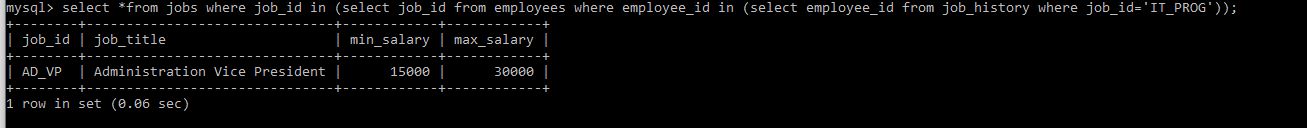
1. **Display details of manager who manages more than 5 employees.**

**select first\_name from employees where employee\_id in(select manager\_id from employees group by manager\_id having count(\*)>5);**

****

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

**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'));**

****

1. **Display the details of employees drawing the highest salary in the department.**

**select department\_id,first\_name,salary from employees outer where salary=(select max(salary) from employees where department\_id=outer.department\_id);**

1. **Display third highest salary of all employees**

**select salary from employees main where 2 = (select count (distinct salary) from employees where salary>main.salary);**