Implementation of Database in Application Development (.NET)

Report by: Sayush Khadka

Table of Contents

1. Introduction: 1
2. Textual analysis: 2
3. Entity Relationship Diagram (ERD): 6
4. Normalization: 7
    4.1. Employee table: 7
        4.1.1. UNF: 7
        4.1.2. 1NF: 7
        4.1.3. 2NF: 7
        4.1.4. 3NF: 8
    4.2. Voting Record table: 9
        4.2.1. UNF: 9
        4.2.2. 1NF: 9
        4.2.3. 2NF: 9
        4.2.4. 3NF: 10
5. Integrations and assumptions: 11
6. Final ERD: 12
7. Data Dictionary: 13
    7.1. Data dictionary of address table: 13
    7.2. Data dictionary of Department table: 14
    7.3. Data dictionary Email table: 15
    7.4. Data dictionary of Employee table: 16
    7.5. Data dictionary EmployeeAddress: 17
    7.6. Data dictionary EmployeeHistory table: 18
    7.7. Data dictionary of Job table: 19
    7.8. Data dictionary of Manager table: 20
    7.9. Data dictionary of Role table: 21
    7.10. Data dictionary of VoteRecord table: 22
8. Script: 23
9. Insert statement: 28
    9.1. Address table insert statement: 28
    9.2. Department table insert statement: 29

9.3. Email table insert statement: 30
9.4. Employee table insert statement: 31
9.5. EmployeeAddress table insert statement: 32
9.6. EmployeeHistory table insert statement: 33
9.7. Job table insert statement: 35
9.8. Manager table insert statement: 36
9.9. Role table insert statement: 37
9.10. VoteRecord table insert statement: 38
10. Select statement: 39
10.1. Address table select statement: 39
10.2. Department table select statement: 39
10.3. Email table select statement: 40
10.4. Employee table select statement: 40
10.5. EmployeeAddress table select statement: 41
10.6. EmployeeHistory table select statement: 41
10.7. Job table select statement: 42
10.8. Manager table select statement: 42
10.9. Role table select statement: 43
10.10. VoteRecord table select statement: 43
11. Forms: 44
11.1. Dashboard for home page: 44
11.2. Complex forms and queries: 45
11.2.1. SQL Queries: 45
11.2.2. Complex forms: 46
11.3. Simple forms: 47
12. User manual: 50
12.1. Dashboard user manual: 50
12.2. Department user manual: 50
12.3. Employee user manual: 51
12.4. Job user manual: 51
12.5. Access Employee info through complex form: 52
13. Testing: 53

13.1. Test Case: To add department to Department table 53
13.2. Test Case: To remove an employee from the Employee table: 55
13.3. Test Case: Update name of Services in Job table: 57
13.4. Test Case: To add address to Address table: 59
13.5. Test Case: To delete UI Developer from Role table 61
14. Further discussion: 63
14.1. Oracle SQL Developer: 63
14.2. Oracle Data Modeler: 63
14.3. Visual Studio: 63
15. Conclusion: 65
Bibliography 66

Table of tables:

Table 1 Data dictionary of Address table.. 13
Table 2 Data dictionary of Department table 14
Table 3 Data dictionary of Email table.. 15
Table 4 Data dictionary of Employee table 16
Table 5 Data dictionary of EmployeeAddress table 17
Table 6 Data dictionary of EmployeeHistory table... 18
Table 7 Data dictionary Job table 19
Table 8 Data dictionary of Manager table... 20
Table 9 Data dictionary Role table 21
Table 10 Data dictionary of VoteRecord table 22
Table 11 To add department to Department table.. 53
Table 12 To remove an employee from the Employee table 55
Table 13 Update name of Services in Job table 57
Table 14 To add address to Address table... 59
Table 15 To delete UI Developer from Role table 61

Table of figures:

Figure 1 Relation between Role and Job 2
Figure 2 Relation between Employee and Role 2
Figure 3 Relation between Employee and Department 3
Figure 4 Relation between Employee History and Employee 3
Figure 5 Relation between VoteRecord and Employee 4
Figure 6 Relation between Employee, Address, and (Employee,Address) 4
Figure 7 Relation between Department and Manager 5
Figure 8 Initial Entity Relationship Diagram (ERD) 6
Figure 9 Final ERD 12
Figure 10 Scripts execution in SQL Developer from DDL script 27
Figure 11 Address table insert statement 28
Figure 12 Department table insert statement 29
Figure 13 Email table insert statement 30
Figure 14 Employee table insert statement 31
Figure 15 EmployeeAddress table insert statement 32
Figure 16 EmployeeHistory table insert statement 34
Figure 17 Job table insert statement 35
Figure 18 Manager table insert statement 36
Figure 19 Role table insert statement 37
Figure 20 VoteRecord table insert statement 38
Figure 21 Address table select statement 39
Figure 22 Department table select statement 39
Figure 23 Email table select statement 40
Figure 24 Employee table select statement 40
Figure 25 EmployeeAddress table select statement 41
Figure 26 EmployeeHistory table select statement 41
Figure 27 Job table select statement 42
Figure 28 Manager table select statement 42
Figure 29 Role table select statement 43
Figure 30 VoteRecord table select statement 43
Figure 31 Dashboard for home page 44
Figure 32 Showing the details of the selected employee 46
Figure 33 Showing the details and the vote record of the selected employee 46
Figure 34 Department web form 47
Figure 35 Employee web form 47
Figure 36 Job web form 48
Figure 37 Role web form 48
Figure 38 Address web form 49
Figure 39 Dashboard user manual 50
Figure 40 Department user manual 50
Figure 41 Employee user manual 51

Figure 42 Job user manual.. 51
Figure 43 Access employee info through complex form... 52
Figure 44 Department Table (web form) 53
Figure 45 Filling department information 54
Figure 46 New department added (Test successful) 54
Figure 47 Employee table before delete.. 55
Figure 48 Employee table after delete (Test successful).. 56
Figure 49 Job table before update 57
Figure 50 Changing the name of Services to Services & Management 58
Figure 51 Job table updated (Test successful)... 58
Figure 52 Address table before 59
Figure 53 Adding new address.. 60
Figure 54 Address added to Address table (Test successful) 60
Figure 55 Role table before deleting UI Developer 61
Figure 56 Role table after deleting UI Developer (Test Successful 1)........................... 62
Figure 57 Role table after delete UI Developer (Test Successful 2).............................. 62

1. Introduction:

The database is a well-organized set of data that has been set up to be simple to manage and update. To put it simply, a database is a place where the data is saved. The best example is a library. The library has many books of different genres; in this situation, the library is a database, and the books are the data (R, 2023).

Simply put, think of the registration at a school. One file contains all the data on the students in one place. In this alleged database, one has access to the information of any student (R, 2023).

With Oracle SQL Developer, Oracle SQL Developer Data Modeler, and ASP.NET with C# in Visual Studio, the project is to create an online voting system. This project entails assessing the case study organization's requirements, creating a web-based database application, and putting the application into use utilizing the aforementioned tools.

The database's tables, columns, and relationships are specified in the DDL script that Data Modeler was used to generate. This will be a crucial step in the development of the application and will guarantee that the data is arranged and saved properly.

2. Textual analysis:

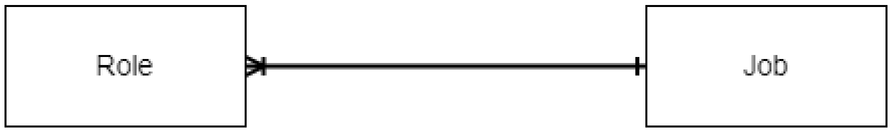


Figure 1 Relation between Role and Job

It can be assumed that a job can have multiple roles associated with it. This means that a job may require different employees with different roles to fulfil the responsibilities of the job. However, it can be assumed that a role can only be assigned to one job at a time, indicating a one-to-many relationship between jobs and roles.

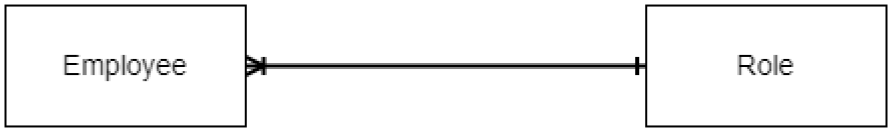


Figure 2 Relation between Employee and Role

It can be assumed that each employee in the organization has a specific role assigned to them. This means that each employee has a unique set of responsibilities and duties within the company. Additionally, it can be assumed that an employee can only have one role assigned to them at a time. However, it is possible for multiple employees to be assigned to the same role.

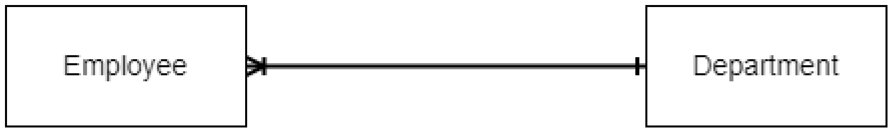


Figure 3 Relation between Employee and Department

It can be assumed that an employee can only belong to one department at a time. This means that an employee cannot work in multiple departments simultaneously. However, multiple employees can work in the same department at the same time. This implies that the company has a hierarchical or divisional structure where employees are organized into different departments based on their roles and responsibilities.

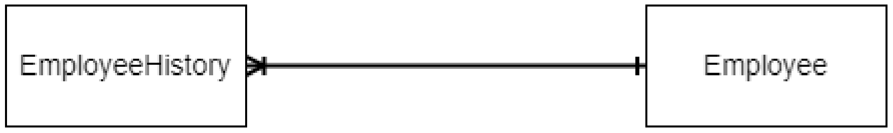


Figure 4 Relation between Employee History and Employee

It can be assumed that an employee must have at least one history, indicating a one-to-many relationship between the employee and employee history entities. Additionally, it can be assumed that an employee can have multiple histories associated with them, but each history is assigned to only one employee. This means that the employee history entity is dependent on the employee entity, and each employee can have one or more histories associated with them. The one-to-many relationship between the employee and employee history entities suggests that the company may maintain records or archives of employee information and work history for various purposes, such as performance evaluation or career development.

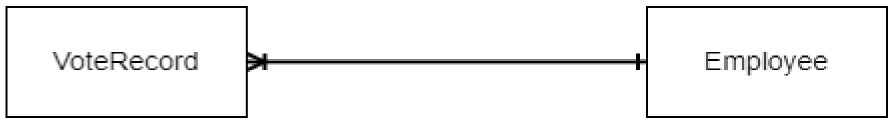


Figure 5 Relation between VoteRecord and Employee

It can be assumed that an employee may or may not have a vote record, indicating an optional relationship between the employee and vote record entities. Additionally, it can be assumed that a vote record is assigned to only one employee, indicating a one-to-many relationship between the employee and vote record entities.

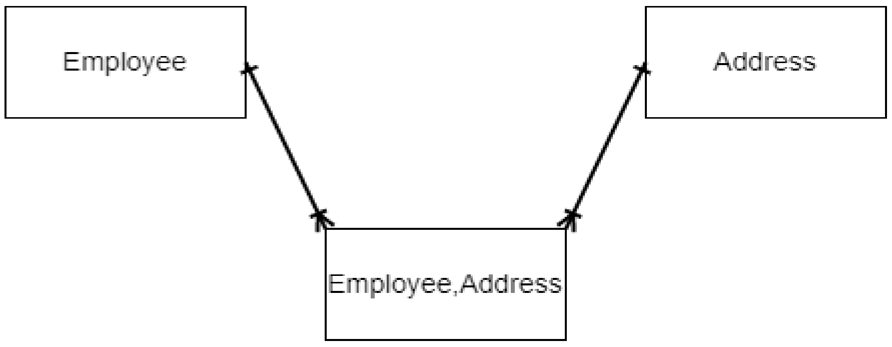


Figure 6 Relation between Employee, Address, and (Employee,Address)

It can be assumed that an employee can have multiple addresses and one address can be assigned to multiple employees. This indicates a many-to-many relationship between the employee and address entities. However, to resolve the potential data anomalies that can occur in a many-to-many relationship, a bridge entity named "employee address" is created. The employee address entity stores the primary keys of both the employee and address entities and combines them to create a composite primary key. This composite primary key ensures that each combination of employee and address is unique and avoids duplication or inconsistencies in the database. The use of the bridge entity in this relationship suggests that it may maintain records of employee addresses for various purposes.

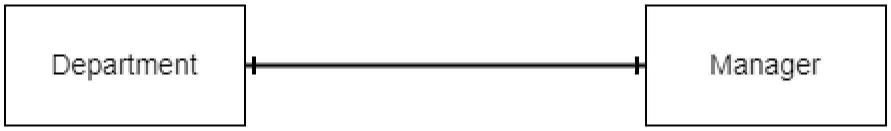


Figure 7 Relation between Department and Manager

It can be assumed that a manager can only manage one department at a time. This means that a manager cannot manage multiple departments simultaneously. Additionally, it can be assumed that a department has only one manager, indicating a one-to-one relationship between the manager and department entities.

3. Entity Relationship Diagram (ERD):

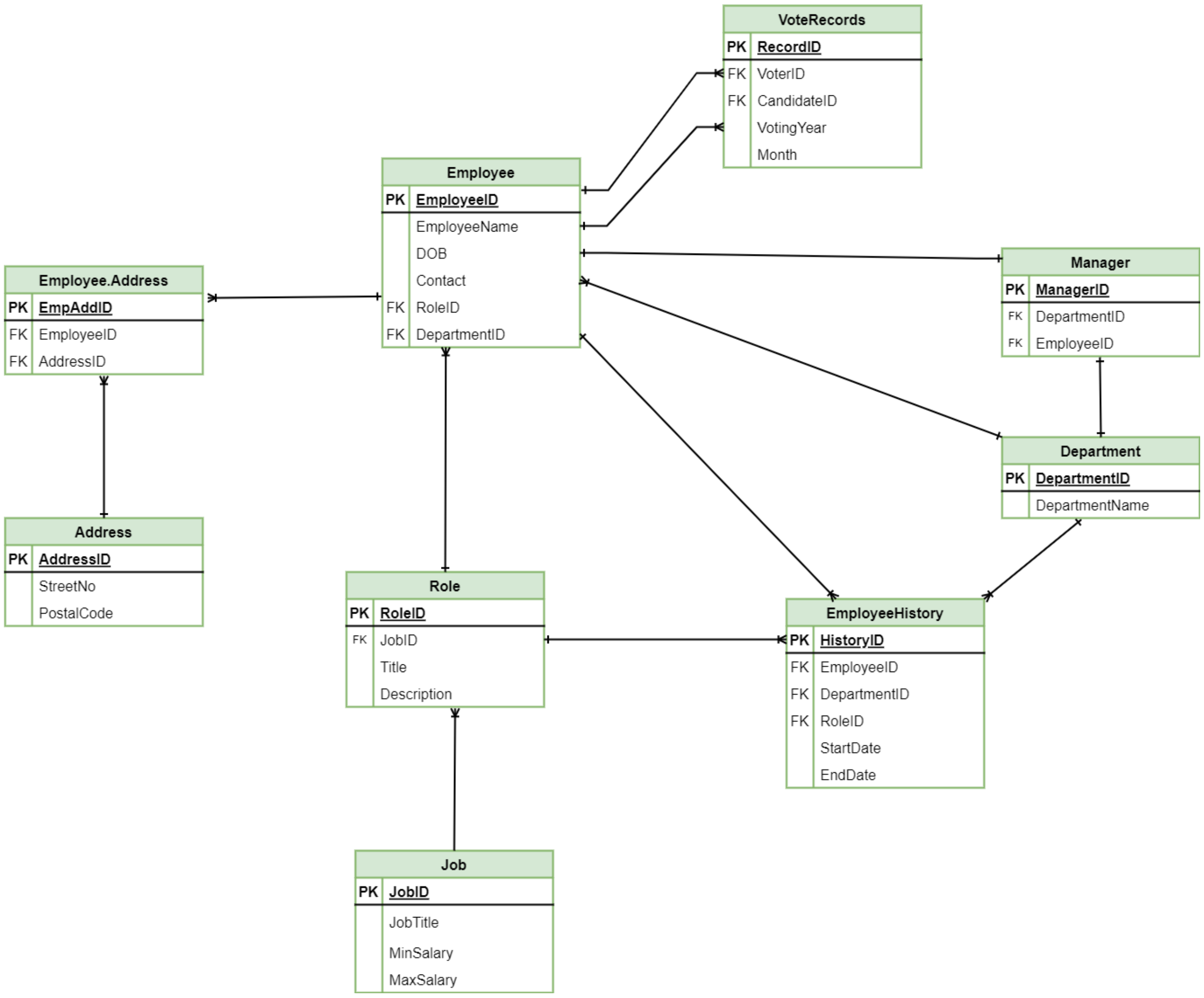


Figure 8 Initial Entity Relationship Diagram (ERD)

4. Normalization:

4.1. Employee table:

4.1.1. UNF:

Employee (**EmployeeID**, EmployeeName, Contact, DOB, Department, {EmailAddress}, {Address})

4.1.2. 1NF:

For 1NF,

Employee 1(**EmployeeID**, EmployeeName, DOB, Contact, Department)

Email 1(Email Address, **<u>EmployeeID</u>**)

Address 1(**AddressID**, Address, **<u>EmployeeID</u>)**

4.1.3. 2NF:

For 2NF,

Checking for partial dependency,

Employee 1 → is already in 2NF as it has no partial dependency.

Email 1 → is already in 2NF as it has no partial dependency.

AddressID → Address

EmployeeID →

EmployeeID, AddressID → forms table

Finally, in 2NF,

Employee 2(**EmployeeID**, EmployeeName, DOB, Contact, Department)

Email 2(EmailAddress, <u>**EmployeeID**</u>)

Address 2(**AddressID**, Address)

Employee -Address (<u>**EmployeeID, AddressID**</u>)

4.1.4. 3NF:

For 3NF,

Checking for transitive dependency,

Employee 2 → is already in 3NF as it has no transitive dependency.

Email 2 → is already in 3NF as it has no transitive dependency.

Address 2 → is already in 3NF as it has no transitive dependency.

Employee -Address → is already in 3NF as it has no transitive dependency.

Finally, in 3NF,

Employee 3(**EmployeeID**, EmployeeName, DOB, Contact, Department)

Email 3(EmailAddress, <u>**EmployeeID**</u>)

Address 3(**AddressID**, Address)

Employee -Address (<u>**EmployeeID, AddressID**</u>)

4.2. Voting Record table:

4.2.1. UNF:

Voter (**VoterID**, VoterName, {VotingYear, {VotingMonth, CandidateID, CandidateName, CandidateDepartment}})

4.2.2. 1NF:

For 1NF,

Voter 1(**VoterID**, VoterName)

VoteYear 1(**VotingYear**, <u>**VoterID**</u>)

Record 1(**VotingMonth**, CandidateID, CandidateName, CandidateDepartment, <u>**VoterID, VotingYear**</u>)

4.2.3. 2NF:

For 2NF,

Checking for partial dependency,

Voter 1 → is already in 2NF as it has no partial dependency.

VoteYear 1 → is already in 2NF as it has no partial dependency.

Record 1 → is already in 2NF as it has no partial dependency.

Finally, in 2NF,

Voter 2(**VoterID**, VoterName)

VoteYear 2(**VotingYear**, <u>**VoterID**</u>)

Record 2(**VotingMonth**, CandidateID, CandidateName, CandidateDepartment, <u>**VoterID, VotingYear**</u>)

4.2.4. 3NF:

For 3NF,

Checking for transitive dependency,

Voter 2 → is already in 3NF as it has no transitive dependency.

VoteYear 2 → is already in 3NF as it has no transitive dependency.

For Record 2 table,

VotingMonth, VoterID, VotingYear → CandidateID

And,

CandidateID → CandidateName, CandidateDepartment

Finally, in 3NF,

Voter 3(**VoterID**, VoterName)

VoteYear 3(**VotingYear**, **<u>VoterID</u>**)

Record 3(**VotingMonth**, **<u>VoterID, VotingYear, CandidateID</u>**)

Candidate (<u>Candidate, </u>CandidateName, CandidateDepartment**)**

5. Integrations and assumptions:

i) Setting the end date to null, it indicates that the employee is currently working in that department and has not yet transferred to a new department or left the company. This avoids the need to constantly update the "Employee" table every time an employee changes department.

ii) A department can have multiple employees in it.

iii) EmployeeAddress table is created as a bridge entity to resolve anomalies as employees can have multiple addresses.

iv) The same job may have multiple roles, which could cause two “many to one” joins converge on a single table; therefore, to address this issue, the role is referenced for the job, from which all employees can access both the job and role data.

v) The table containing information about an employee's work history includes their present department, but the end date for this department is left blank or undefined.

6. Final ERD:

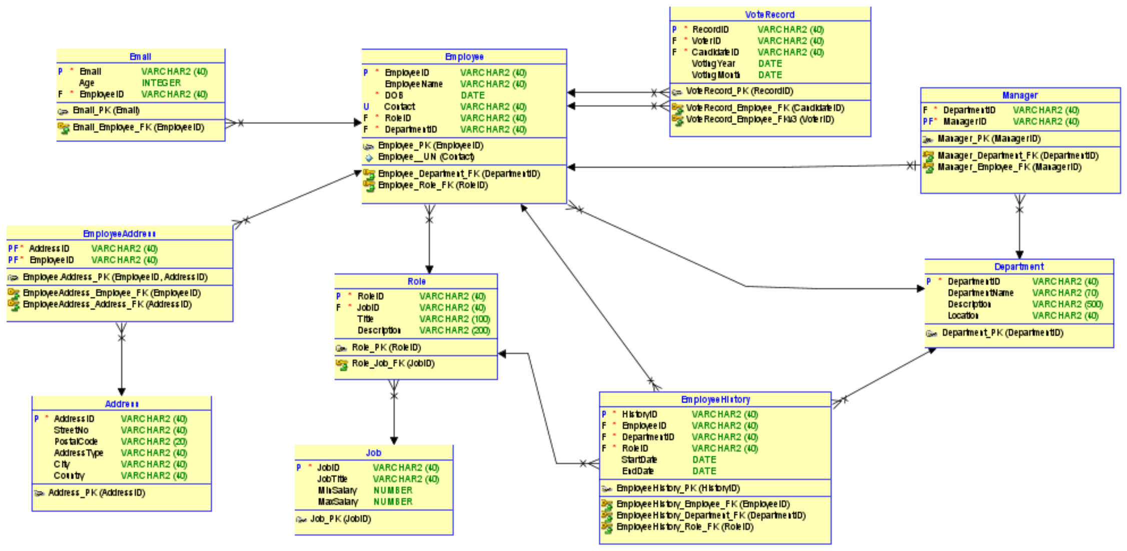


Figure 9 Final ERD

7. Data Dictionary:

7.1. Data dictionary of address table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
ADDRESSID	VARCHAR	40	PRIMARY KEY			Defines unique address.	Add4
STREETNO	VARCHAR	40	NOT NULL			Defines street no.	Kusunti 04
POSTALCODE	VARCHAR	20	NOT NULL			Defines postal code.	432-112
ADDRESSTYPE	VARCHAR	40	NOT NULL			Defines address type.	Permanent
CITY	VARCHAR	40	NOT NULL			Mentions city.	Lalitpur
COUNTRY	VARCHAR	40	NOT NULL			Mention country.	Nepal

Table 1 Data dictionary of Address table

7.2. Data dictionary of Department table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
DEPARTMENTID	VARCHAR	40	PRIMARY KEY			Defines unique department id.	Dep3
DEPARTMENTNAME	VARCHAR	40	NOT NULL			Defines department name.	Human Resource
DESCRIPTION	VARCHAR	500	NOT NULL			Provides description of the department.	Help provide organizational structure and the ability to meet business needs
LOCATION	VARCHAR	40	NOT NULL			Mentions location of the department.	Alice Block

Table 2 Data dictionary of Department table

7.3. Data dictionary Email table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
EMAIL	VARCHAR	40	PRIMARY KEY			Defines unique email.	kurt@gmail.com
AGE	NUMBER	40	NOT NULL			Provides age.	19
EMPLOYEEID	VARCHAR	40	FOREIGN KEY	EMPLOYEE	EMPLOYEEID	Foreign key of employee table.	Emp3

Table 3 Data dictionary of Email table

7.4. Data dictionary of Employee table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Examp Data
EMPLOYEEID	VARCHAR	40	PRIMARY KEY			Defines unique employee id.	Add4
EMPLOYEENAME	VARCHAR	40	NOT NULL			Define employee name.	Kusunt
DOB	DATE		NOT NULL			Provides date of birth.	432-11
CONTACT	VARCHAR	40	NOT NULL			Provides contact number.	Perma
ROLEID	VARCHAR	40	FOREIGN KEY	ROLE	ROLEID	Foreign key of role table	Lalitpu
DEPARTMENTID	VARCHAR	40	FOREIGN KEY	DEPARTMENT	DEPARTMENTID	Foreign key of department table.	Nepal

Table 4 Data dictionary of Employee table

7.5. Data dictionary EmployeeAddress:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
ADDRESSID	VARCHAR	40	PRIMARY KEY			Defines unique address id.	Add3
EMPLOYEEID	VARCHAR	40	FOREIGN KEY	EMPLOYEE	EMPLOYEEID	Foreign key of employee table.	Emp3

Table 5 Data dictionary of EmployeeAddress table

7.6. Data dictionary EmployeeHistory table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
HISTORYID	VARCHAR	40	PRIMARY KEY			Defines unique history id.	His3
EMPLOYEEID	VARCHAR	40	FOREIGN KEY	EMPLOYEE	EMPLOYEEID	Foreign key of employee table.	Emp3
DEPARTMENTID	VARCHAR	20	FOREIGN KEY	DEPARTMENT	DEPARTMENTID	Foreign key of department table.	Dep2
ROLEID	VARCHAR	40	FOREIGN KEY	ROLE	ROLEID	Foreign key of role table.	Role2
STARTDATE	DATE		NOT NULL			Start date of the employee history.	12-DEC-02
ENDDATE	DATE		NULL			End date of the employee history.	Null

Table 6 Data dictionary of EmployeeHistory table

7.7. Data dictionary of Job table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
JOBID	VARCHAR	40	PRIMARY KEY			Defines unique job id.	Job3
JOBTITLE	VARCHAR	40	NOT NULL			Defines job title.	Web Developer
MINSALARY	NUMBER		NOT NULL			Minimum salary of the job.	17000
MAXSALARY	NUMBER		NOT NULL			Maximum salary of the job.	20000

Table 7 Data dictionary Job table

7.8. Data dictionary of Manager table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
DEPARTMENTID	VARCHAR	40	FOREIGN KEY	DEPARTMENT	DEPARTMENTID	Foreign of department table.	Add4
MANAGERID	VARCHAR	40	PRIMARY KEY			Defines unique manager id.	Kusunti 04

Table 8 Data dictionary of Manager table

7.9. Data dictionary of Role table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
ROLEID	VARCHAR	40	PRIMARY KEY			Defines unique role id.	Role3
JOBID	VARCHAR	40	FOREIGN KEY	JOB	JOBID	Foreign key of job table.	Job3
TITLE	VARCHAR	100	NOT NULL			Provides title of role.	Full Stack Developer
DESCRIPTION	VARCHAR	200	NOT NULL			Provides description of role.	Related with both frontend and backend of the software (Web)

Table 9 Data dictionary Role table

7.10. Data dictionary of VoteRecord table:

Column Name	Data Type	Size	Constraint	Reference Table	Reference Column	Description	Example Data
RECORDID	VARCHAR	40	PRIMARY KEY			Defines unique record id.	Rec3
VOTERID	VARCHAR	40	FOREIGN KEY	EMPLOYEE	EMPLOYEEID	Provides voter id references.	Emp3
CANDIDATEID	VARCHAR	40	FOREIGN KEY	EMPLOYEE	EMPLOYEEID	Provides candidate id references.	Emp1
VOTINGYEAR	DATE	40	NOT NULL			Provides voting year	2007
VOTINGMONTH	DATE	40	NOT NULL			Provides voting month.	DEC

Table 10 Data dictionary of VoteRecord table

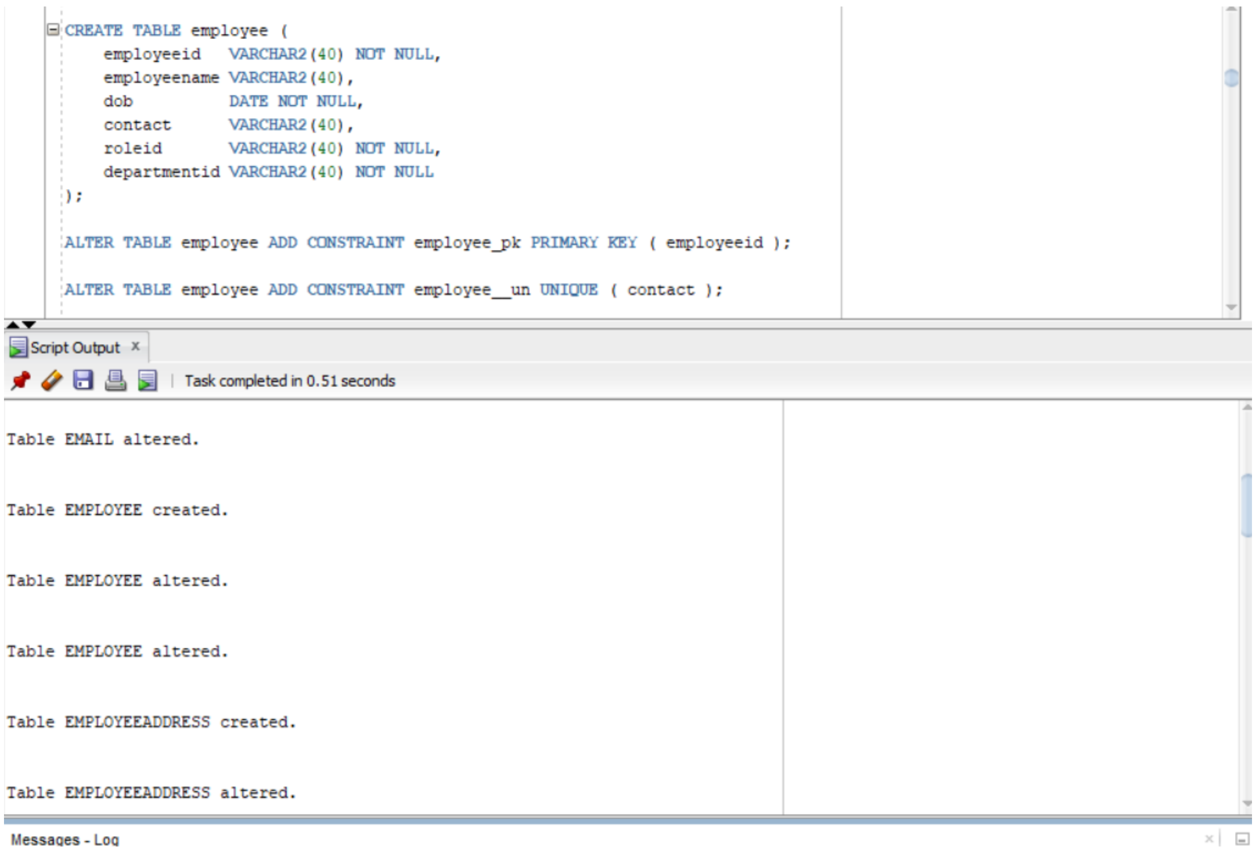
8. Script:

```
CREATE TABLE address (
    addressid   VARCHAR2(40) NOT NULL,
    streetno    VARCHAR2(40),
    postalcode  VARCHAR2(20),
    addresstype VARCHAR2(40),
    city        VARCHAR2(40),
    country     VARCHAR2(40)
);

ALTER TABLE address ADD CONSTRAINT address_pk PRIMARY KEY ( addressid );

CREATE TABLE department (
    departmentid   VARCHAR2(40) NOT NULL,
    departmentname VARCHAR2(70),
    description    VARCHAR2(500),
    location       VARCHAR2(40)
);

ALTER TABLE department ADD CONSTRAINT department_pk PRIMARY KEY (
departmentid );

CREATE TABLE email (
    email      VARCHAR2(40) NOT NULL,
    age        INTEGER,
    employeeid VARCHAR2(40) NOT NULL
);

ALTER TABLE email ADD CONSTRAINT email_pk PRIMARY KEY ( email );

CREATE TABLE employee (
    employeeid   VARCHAR2(40) NOT NULL,
    employeename VARCHAR2(40),
    dob          DATE NOT NULL,
    contact      VARCHAR2(40),
    roleid       VARCHAR2(40) NOT NULL,
    departmentid VARCHAR2(40) NOT NULL
);

ALTER TABLE employee ADD CONSTRAINT employee_pk PRIMARY KEY (
employeeid );

ALTER TABLE employee ADD CONSTRAINT employee__un UNIQUE ( contact );
```

```
CREATE TABLE employeeaddress (
    addressid  VARCHAR2(40) NOT NULL,
    employeeid VARCHAR2(40) NOT NULL
);

ALTER TABLE employeeaddress ADD CONSTRAINT "Employee.Address_PK"
PRIMARY KEY ( employeeid,
                                                                  addressid );

CREATE TABLE employeehistory (
    historyid    VARCHAR2(40) NOT NULL,
    employeeid   VARCHAR2(40) NOT NULL,
    departmentid VARCHAR2(40) NOT NULL,
    roleid       VARCHAR2(40) NOT NULL,
    startdate    DATE,
    enddate      DATE
);

ALTER TABLE employeehistory ADD CONSTRAINT employeehistory_pk PRIMARY
KEY ( historyid );

CREATE TABLE job (
    jobid     VARCHAR2(40) NOT NULL,
    jobtitle  VARCHAR2(40),
    minsalary NUMBER,
    maxsalary NUMBER
);

ALTER TABLE job ADD CONSTRAINT job_pk PRIMARY KEY ( jobid );

CREATE TABLE manager (
    departmentid VARCHAR2(40) NOT NULL,
    managerid    VARCHAR2(40) NOT NULL
);

ALTER TABLE manager ADD CONSTRAINT manager_pk PRIMARY KEY ( managerid
);

CREATE TABLE role (
    roleid      VARCHAR2(40) NOT NULL,
    jobid       VARCHAR2(40) NOT NULL,
    title       VARCHAR2(100),
    description VARCHAR2(200)
);
```

```
ALTER TABLE role ADD CONSTRAINT role_pk PRIMARY KEY ( roleid );

CREATE TABLE voterecord (
    recordid    VARCHAR2(40) NOT NULL,
    voterid     VARCHAR2(40) NOT NULL,
    candidateid VARCHAR2(40) NOT NULL,
    votingyear  DATE,
    votingmonth DATE
);

ALTER TABLE voterecord ADD CONSTRAINT voterecord_pk PRIMARY KEY (
recordid );

ALTER TABLE email
    ADD CONSTRAINT email_employee_fk FOREIGN KEY ( employeeid )
        REFERENCES employee ( employeeid )
            ON DELETE CASCADE;

ALTER TABLE employee
    ADD CONSTRAINT employee_department_fk FOREIGN KEY ( departmentid )
        REFERENCES department ( departmentid )
            ON DELETE CASCADE;

ALTER TABLE employee
    ADD CONSTRAINT employee_role_fk FOREIGN KEY ( roleid )
        REFERENCES role ( roleid )
            ON DELETE CASCADE;

ALTER TABLE employeeaddress
    ADD CONSTRAINT employeeaddress_address_fk FOREIGN KEY ( addressid )
        REFERENCES address ( addressid )
            ON DELETE CASCADE;

ALTER TABLE employeeaddress
    ADD CONSTRAINT employeeaddress_employee_fk FOREIGN KEY ( employeeid )
        REFERENCES employee ( employeeid )
            ON DELETE CASCADE;

ALTER TABLE employeehistory
    ADD CONSTRAINT employeehistory_department_fk FOREIGN KEY ( departmentid )
        REFERENCES department ( departmentid )
            ON DELETE CASCADE;

ALTER TABLE employeehistory
    ADD CONSTRAINT employeehistory_employee_fk FOREIGN KEY ( employeeid )
        REFERENCES employee ( employeeid )
```

```
      ON DELETE CASCADE;

ALTER TABLE employeehistory
  ADD CONSTRAINT employeehistory_role_fk FOREIGN KEY ( roleid )
    REFERENCES role ( roleid )
      ON DELETE CASCADE;

ALTER TABLE manager
  ADD CONSTRAINT manager_department_fk FOREIGN KEY ( departmentid )
    REFERENCES department ( departmentid )
      ON DELETE CASCADE;

ALTER TABLE manager
  ADD CONSTRAINT manager_employee_fk FOREIGN KEY ( managerid )
    REFERENCES employee ( employeeid )
      ON DELETE CASCADE;

ALTER TABLE role
  ADD CONSTRAINT role_job_fk FOREIGN KEY ( jobid )
    REFERENCES job ( jobid )
      ON DELETE CASCADE;

ALTER TABLE voterecord
  ADD CONSTRAINT voterecord_employee_fk FOREIGN KEY ( candidateid )
    REFERENCES employee ( employeeid )
      ON DELETE CASCADE;

ALTER TABLE voterecord
  ADD CONSTRAINT voterecord_employee_fkv3 FOREIGN KEY ( voterid )
    REFERENCES employee ( employeeid )
      ON DELETE CASCADE;
```



```
CREATE TABLE employee (
    employeeid   VARCHAR2(40) NOT NULL,
    employeename VARCHAR2(40),
    dob          DATE NOT NULL,
    contact      VARCHAR2(40),
    roleid       VARCHAR2(40) NOT NULL,
    departmentid VARCHAR2(40) NOT NULL
);

ALTER TABLE employee ADD CONSTRAINT employee_pk PRIMARY KEY ( employeeid );

ALTER TABLE employee ADD CONSTRAINT employee__un UNIQUE ( contact );
```

Script Output

Task completed in 0.51 seconds

```
Table EMAIL altered.


Table EMPLOYEE created.


Table EMPLOYEE altered.


Table EMPLOYEE altered.


Table EMPLOYEEADDRESS created.


Table EMPLOYEEADDRESS altered.
```

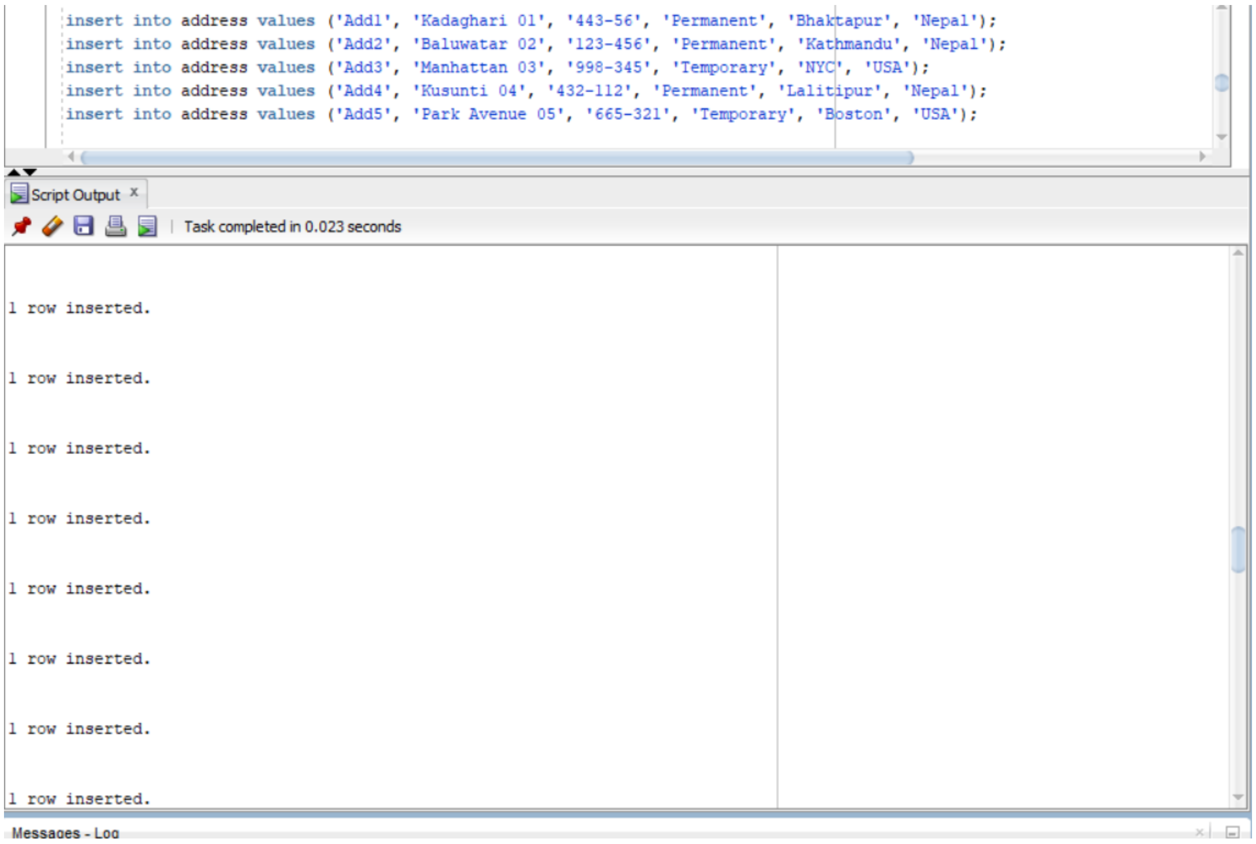
Messages - Log

Figure 10 Scripts execution in SQL Developer from DDL script

9. Insert statement:

9.1. Address table insert statement:

```
insert into address values ('Add1', 'Kadaghari 01', '443-56', 'Permanent', 'Bhaktapur', 'Nepal');
insert into address values ('Add2', 'Baluwatar 02', '123-456', 'Permanent', 'Kathmandu', 'Nepal');
insert into address values ('Add3', 'Manhattan 03', '998-345', 'Temporary', 'NYC', 'USA');
insert into address values ('Add4', 'Kusunti 04', '432-112', 'Permanent', 'Lalitipur', 'Nepal');
insert into address values ('Add5', 'Park Avenue 05', '665-321', 'Temporary', 'Boston', 'USA');
```



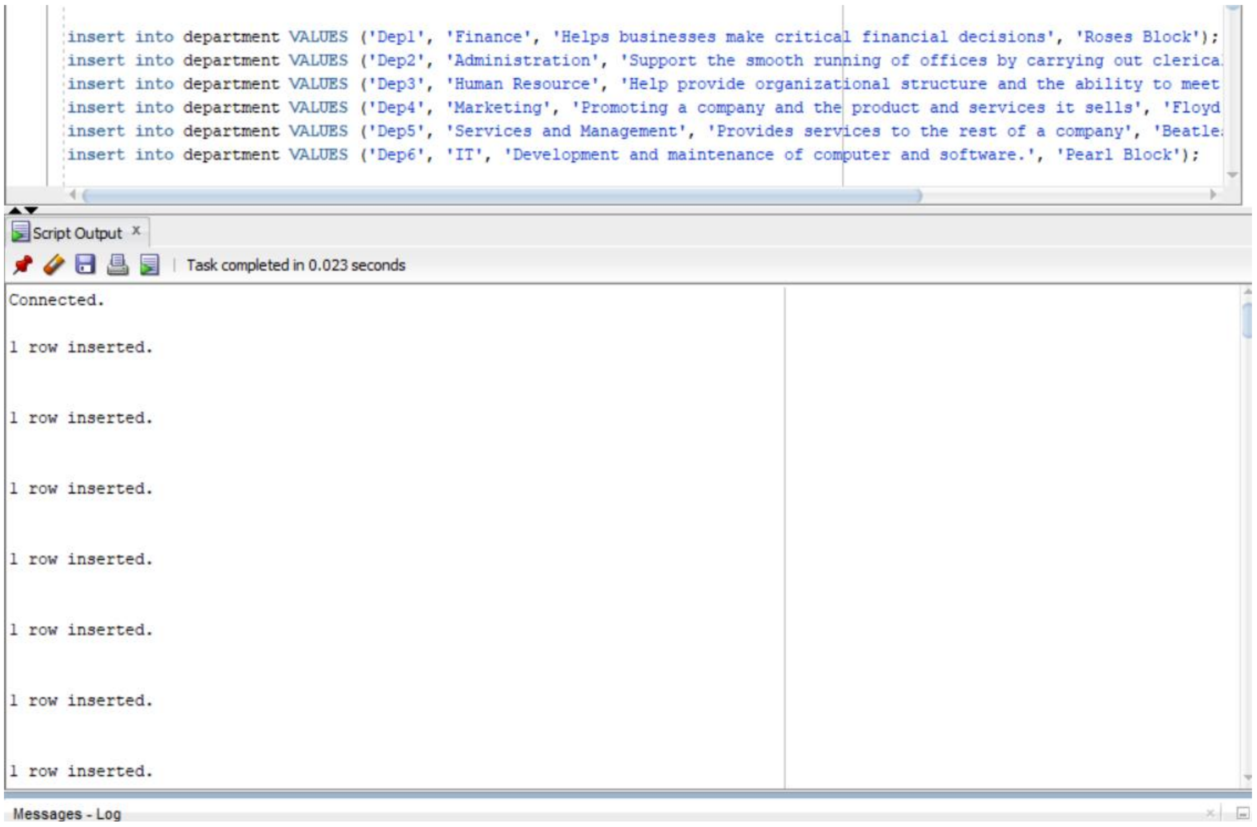
```
insert into address values ('Add1', 'Kadaghari 01', '443-56', 'Permanent', 'Bhaktapur', 'Nepal');
insert into address values ('Add2', 'Baluwatar 02', '123-456', 'Permanent', 'Kathmandu', 'Nepal');
insert into address values ('Add3', 'Manhattan 03', '998-345', 'Temporary', 'NYC', 'USA');
insert into address values ('Add4', 'Kusunti 04', '432-112', 'Permanent', 'Lalitipur', 'Nepal');
insert into address values ('Add5', 'Park Avenue 05', '665-321', 'Temporary', 'Boston', 'USA');
```

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

Figure 11 Address table insert statement

9.2. Department table insert statement:

```
insert into department VALUES ('Dep1', 'Finance', 'Helps businesses make critical financial decisions', 'Roses Block');
insert into department VALUES ('Dep2', 'Administration', 'Support the smooth running of offices by carrying out clerical tasks and projects', 'Nirvana Block');
insert into department VALUES ('Dep3', 'Human Resource', 'Help provide organizational structure and the ability to meet business needs', 'Alice Block');
insert into department VALUES ('Dep4', 'Marketing', 'Promoting a company and the product and services it sells', 'Floyd Block');
insert into department VALUES ('Dep5', 'Services and Management', 'Provides services to the rest of a company', 'Beatles Hall');
insert into department VALUES ('Dep6', 'IT', 'Development and maintenance of computer and software.', 'Pearl Block');
```



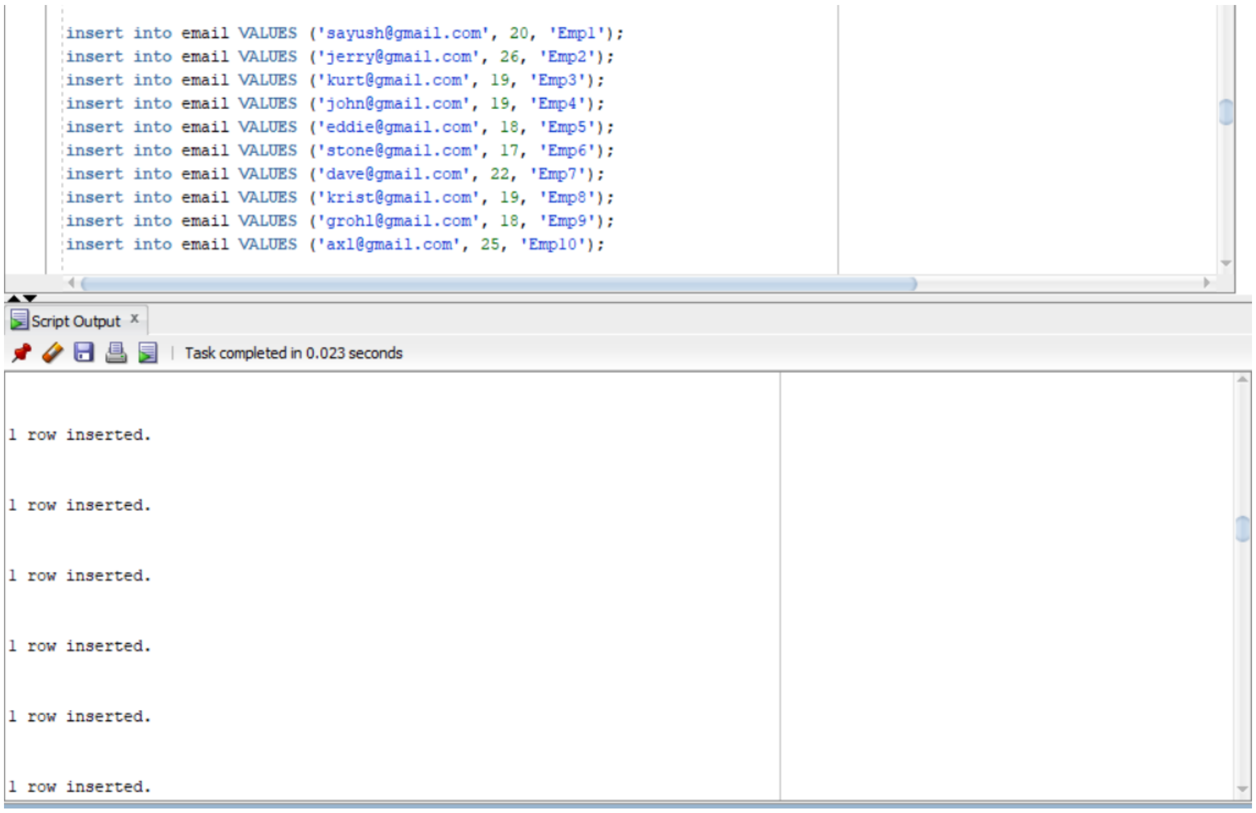
```
insert into department VALUES ('Dep1', 'Finance', 'Helps businesses make critical financial decisions', 'Roses Block');
insert into department VALUES ('Dep2', 'Administration', 'Support the smooth running of offices by carrying out clerica
insert into department VALUES ('Dep3', 'Human Resource', 'Help provide organizational structure and the ability to meet
insert into department VALUES ('Dep4', 'Marketing', 'Promoting a company and the product and services it sells', 'Floyd
insert into department VALUES ('Dep5', 'Services and Management', 'Provides services to the rest of a company', 'Beatle
insert into department VALUES ('Dep6', 'IT', 'Development and maintenance of computer and software.', 'Pearl Block');
```

```
Connected.

1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

Figure 12 Department table insert statement

9.3. Email table insert statement:

```
insert into email VALUES ('sayush@gmail.com', 20, 'Emp1');
insert into email VALUES ('jerry@gmail.com', 26, 'Emp2');
insert into email VALUES ('kurt@gmail.com', 19, 'Emp3');
insert into email VALUES ('john@gmail.com', 19, 'Emp4');
insert into email VALUES ('eddie@gmail.com', 18, 'Emp5');
insert into email VALUES ('stone@gmail.com', 17, 'Emp6');
insert into email VALUES ('dave@gmail.com', 22, 'Emp7');
insert into email VALUES ('krist@gmail.com', 19, 'Emp8');
insert into email VALUES ('grohl@gmail.com', 18, 'Emp9');
insert into email VALUES ('axl@gmail.com', 25, 'Emp10');
```



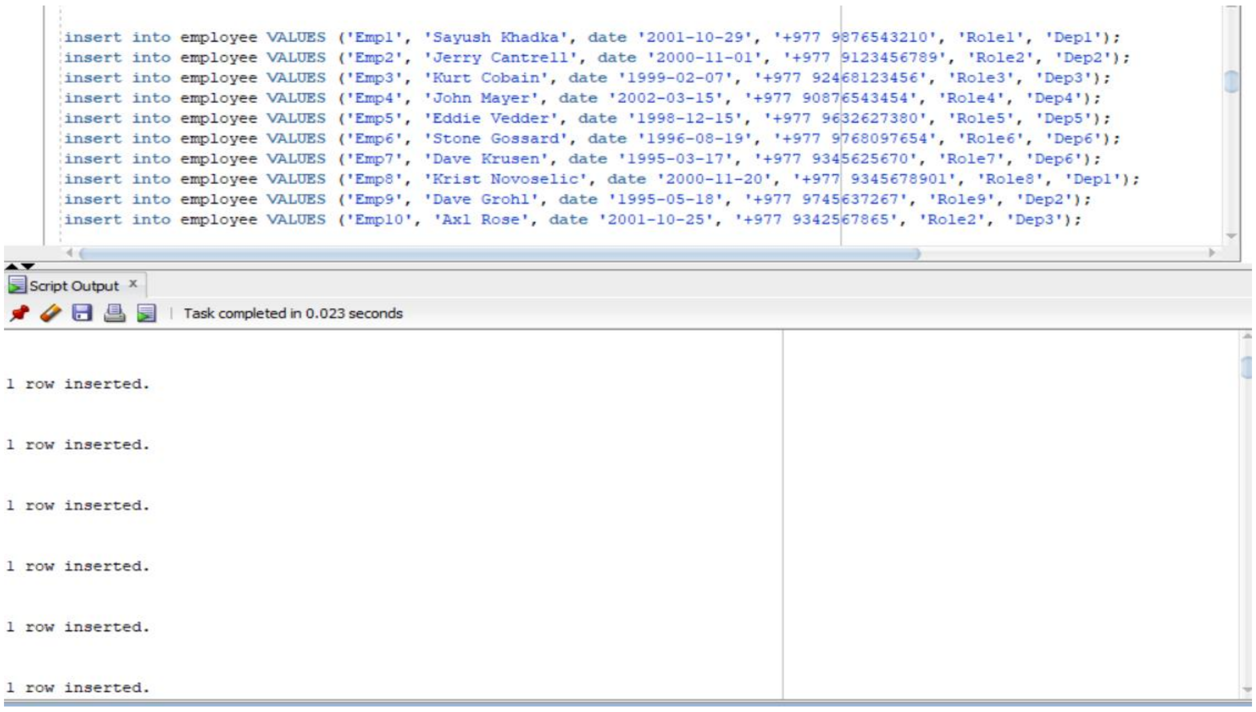
```
insert into email VALUES ('sayush@gmail.com', 20, 'Emp1');
insert into email VALUES ('jerry@gmail.com', 26, 'Emp2');
insert into email VALUES ('kurt@gmail.com', 19, 'Emp3');
insert into email VALUES ('john@gmail.com', 19, 'Emp4');
insert into email VALUES ('eddie@gmail.com', 18, 'Emp5');
insert into email VALUES ('stone@gmail.com', 17, 'Emp6');
insert into email VALUES ('dave@gmail.com', 22, 'Emp7');
insert into email VALUES ('krist@gmail.com', 19, 'Emp8');
insert into email VALUES ('grohl@gmail.com', 18, 'Emp9');
insert into email VALUES ('axl@gmail.com', 25, 'Emp10');
```

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

Figure 13 Email table insert statement

9.4. Employee table insert statement:

```
insert into employee VALUES ('Emp1', 'Sayush Khadka', date '2001-10-29', '+977
9876543210', 'Role1', 'Dep1');
insert into employee VALUES ('Emp2', 'Jerry Cantrell', date '2000-11-01', '+977
9123456789', 'Role2', 'Dep2');
insert into employee VALUES ('Emp3', 'Kurt Cobain', date '1999-02-07', '+977
92468123456', 'Role3', 'Dep3');
insert into employee VALUES ('Emp4', 'John Mayer', date '2002-03-15', '+977
90876543454', 'Role4', 'Dep4');
insert into employee VALUES ('Emp5', 'Eddie Vedder', date '1998-12-15', '+977
9632627380', 'Role5', 'Dep5');
insert into employee VALUES ('Emp6', 'Stone Gossard', date '1996-08-19', '+977
9768097654', 'Role6', 'Dep6');
insert into employee VALUES ('Emp7', 'Dave Krusen', date '1995-03-17', '+977
9345625670', 'Role7', 'Dep6');
insert into employee VALUES ('Emp8', 'Krist Novoselic', date '2000-11-20', '+977
9345678901', 'Role8', 'Dep1');
insert into employee VALUES ('Emp9', 'Dave Grohl', date '1995-05-18', '+977
9745637267', 'Role9', 'Dep2');
insert into employee VALUES ('Emp10', 'Axl Rose', date '2001-10-25', '+977
9342567865', 'Role2', 'Dep3');
```



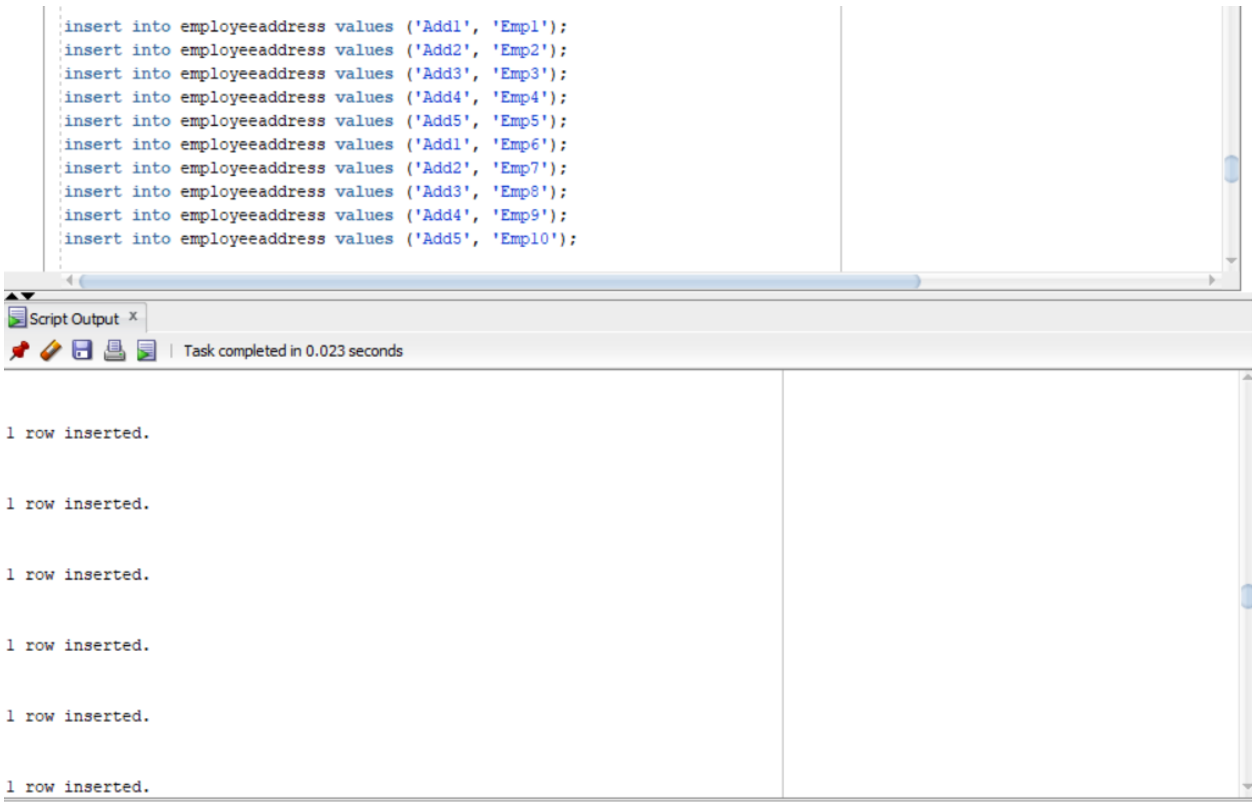
```
insert into employee VALUES ('Emp1', 'Sayush Khadka', date '2001-10-29', '+977 9876543210', 'Role1', 'Dep1');
insert into employee VALUES ('Emp2', 'Jerry Cantrell', date '2000-11-01', '+977 9123456789', 'Role2', 'Dep2');
insert into employee VALUES ('Emp3', 'Kurt Cobain', date '1999-02-07', '+977 92468123456', 'Role3', 'Dep3');
insert into employee VALUES ('Emp4', 'John Mayer', date '2002-03-15', '+977 90876543454', 'Role4', 'Dep4');
insert into employee VALUES ('Emp5', 'Eddie Vedder', date '1998-12-15', '+977 9632627380', 'Role5', 'Dep5');
insert into employee VALUES ('Emp6', 'Stone Gossard', date '1996-08-19', '+977 9768097654', 'Role6', 'Dep6');
insert into employee VALUES ('Emp7', 'Dave Krusen', date '1995-03-17', '+977 9345625670', 'Role7', 'Dep6');
insert into employee VALUES ('Emp8', 'Krist Novoselic', date '2000-11-20', '+977 9345678901', 'Role8', 'Dep1');
insert into employee VALUES ('Emp9', 'Dave Grohl', date '1995-05-18', '+977 9745637267', 'Role9', 'Dep2');
insert into employee VALUES ('Emp10', 'Axl Rose', date '2001-10-25', '+977 9342567865', 'Role2', 'Dep3');
```

```
Task completed in 0.023 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

Figure 14 Employee table insert statement

9.5. EmployeeAddress table insert statement:

```
insert into employeeaddress values ('Add1', 'Emp1');
insert into employeeaddress values ('Add2', 'Emp2');
insert into employeeaddress values ('Add3', 'Emp3');
insert into employeeaddress values ('Add4', 'Emp4');
insert into employeeaddress values ('Add5', 'Emp5');
insert into employeeaddress values ('Add1', 'Emp6');
insert into employeeaddress values ('Add2', 'Emp7');
insert into employeeaddress values ('Add3', 'Emp8');
insert into employeeaddress values ('Add4', 'Emp9');
insert into employeeaddress values ('Add5', 'Emp10');
```



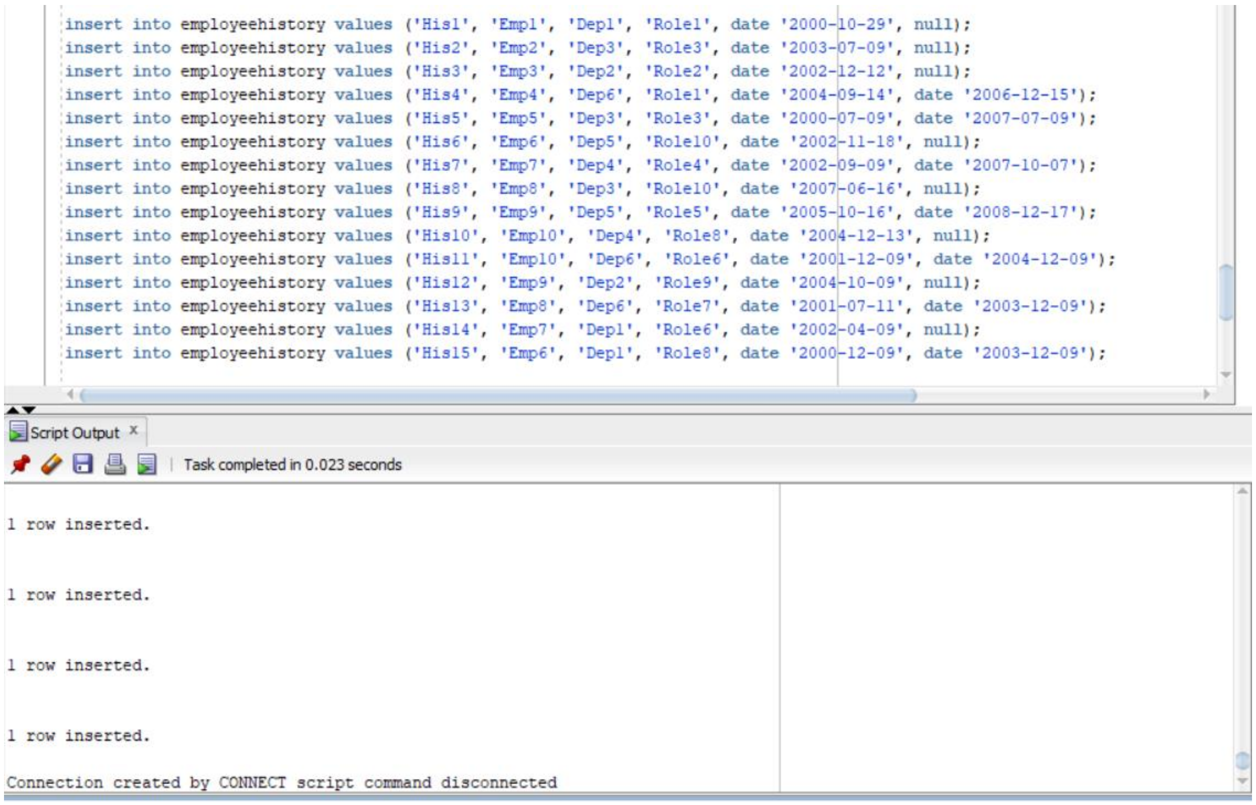
```
insert into employeeaddress values ('Add1', 'Emp1');
insert into employeeaddress values ('Add2', 'Emp2');
insert into employeeaddress values ('Add3', 'Emp3');
insert into employeeaddress values ('Add4', 'Emp4');
insert into employeeaddress values ('Add5', 'Emp5');
insert into employeeaddress values ('Add1', 'Emp6');
insert into employeeaddress values ('Add2', 'Emp7');
insert into employeeaddress values ('Add3', 'Emp8');
insert into employeeaddress values ('Add4', 'Emp9');
insert into employeeaddress values ('Add5', 'Emp10');
```

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

Figure 15 EmployeeAddress table insert statement

9.6. EmployeeHistory table insert statement:

```
insert into employeehistory values ('His1', 'Emp1', 'Dep1', 'Role1', date '2000-10-29',
null);
insert into employeehistory values ('His2', 'Emp2', 'Dep3', 'Role3', date '2003-07-09',
null);
insert into employeehistory values ('His3', 'Emp3', 'Dep2', 'Role2', date '2002-12-12',
null);
insert into employeehistory values ('His4', 'Emp4', 'Dep6', 'Role1', date '2004-09-14',
date '2006-12-15');
insert into employeehistory values ('His5', 'Emp5', 'Dep3', 'Role3', date '2000-07-09',
date '2007-07-09');
insert into employeehistory values ('His6', 'Emp6', 'Dep5', 'Role10', date '2002-11-18',
null);
insert into employeehistory values ('His7', 'Emp7', 'Dep4', 'Role4', date '2002-09-09',
date '2007-10-07');
insert into employeehistory values ('His8', 'Emp8', 'Dep3', 'Role10', date '2007-06-16',
null);
insert into employeehistory values ('His9', 'Emp9', 'Dep5', 'Role5', date '2005-10-16',
date '2008-12-17');
insert into employeehistory values ('His10', 'Emp10', 'Dep4', 'Role8', date '2004-12-13',
null);
insert into employeehistory values ('His11', 'Emp10', 'Dep6', 'Role6', date '2001-12-09',
date '2004-12-09');
insert into employeehistory values ('His12', 'Emp9', 'Dep2', 'Role9', date '2004-10-09',
null);
insert into employeehistory values ('His13', 'Emp8', 'Dep6', 'Role7', date '2001-07-11',
date '2003-12-09');
insert into employeehistory values ('His14', 'Emp7', 'Dep1', 'Role6', date '2002-04-09',
null);
insert into employeehistory values ('His15', 'Emp6', 'Dep1', 'Role8', date '2000-12-09',
date '2003-12-09');
```



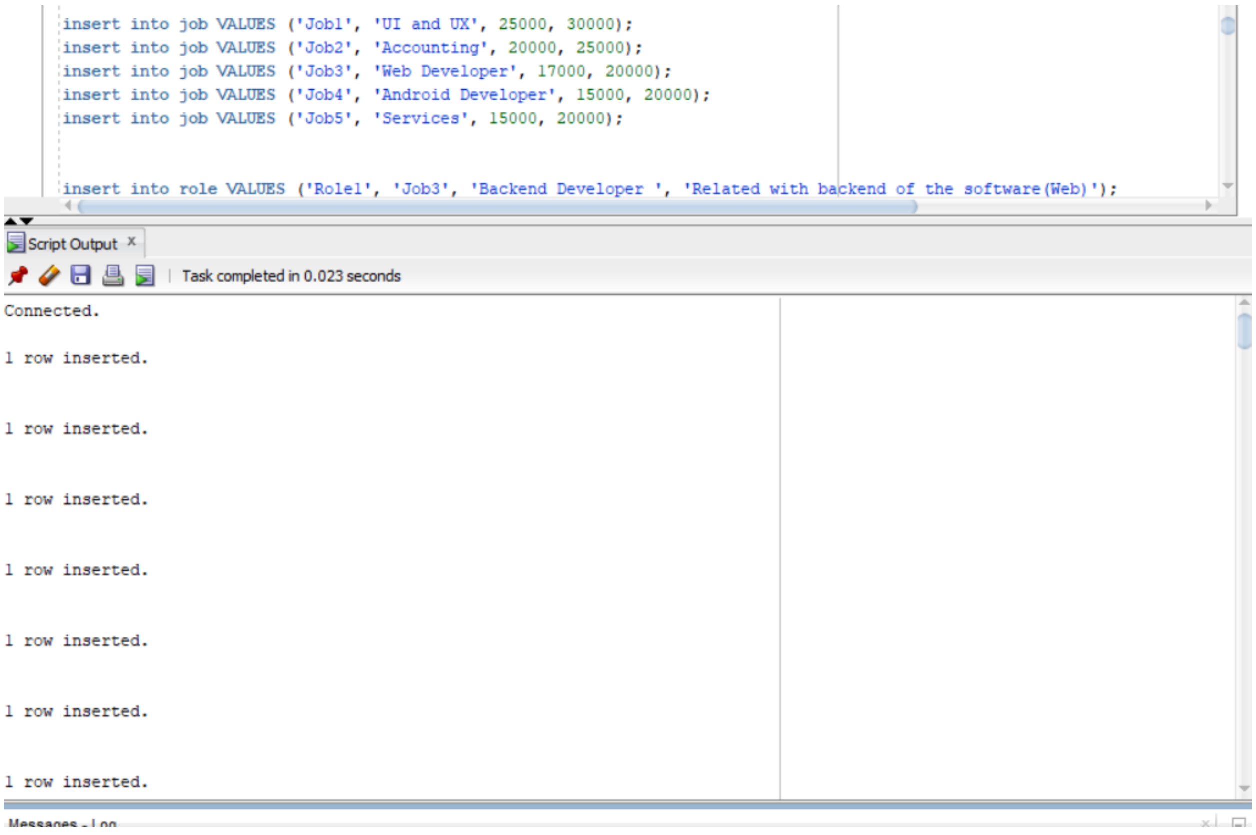
```
insert into employeehistory values ('His1', 'Emp1', 'Dep1', 'Role1', date '2000-10-29', null);
insert into employeehistory values ('His2', 'Emp2', 'Dep3', 'Role3', date '2003-07-09', null);
insert into employeehistory values ('His3', 'Emp3', 'Dep2', 'Role2', date '2002-12-12', null);
insert into employeehistory values ('His4', 'Emp4', 'Dep6', 'Role1', date '2004-09-14', date '2006-12-15');
insert into employeehistory values ('His5', 'Emp5', 'Dep3', 'Role3', date '2000-07-09', date '2007-07-09');
insert into employeehistory values ('His6', 'Emp6', 'Dep5', 'Role10', date '2002-11-18', null);
insert into employeehistory values ('His7', 'Emp7', 'Dep4', 'Role4', date '2002-09-09', date '2007-10-07');
insert into employeehistory values ('His8', 'Emp8', 'Dep3', 'Role10', date '2007-06-16', null);
insert into employeehistory values ('His9', 'Emp9', 'Dep5', 'Role5', date '2005-10-16', date '2008-12-17');
insert into employeehistory values ('His10', 'Emp10', 'Dep4', 'Role8', date '2004-12-13', null);
insert into employeehistory values ('His11', 'Emp10', 'Dep6', 'Role6', date '2001-12-09', date '2004-12-09');
insert into employeehistory values ('His12', 'Emp9', 'Dep2', 'Role9', date '2004-10-09', null);
insert into employeehistory values ('His13', 'Emp8', 'Dep6', 'Role7', date '2001-07-11', date '2003-12-09');
insert into employeehistory values ('His14', 'Emp7', 'Dep1', 'Role6', date '2002-04-09', null);
insert into employeehistory values ('His15', 'Emp6', 'Dep1', 'Role8', date '2000-12-09', date '2003-12-09');
```

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.

Connection created by CONNECT script command disconnected
```

Figure 16 EmployeeHistory table insert statement

9.7. Job table insert statement:

```
insert into job VALUES ('Job1', 'UI and UX', 25000, 30000);
insert into job VALUES ('Job2', 'Accounting', 20000, 25000);
insert into job VALUES ('Job3', 'Web Developer', 17000, 20000);
insert into job VALUES ('Job4', 'Android Developer', 15000, 20000);
insert into job VALUES ('Job5', 'Services', 15000, 20000);
```



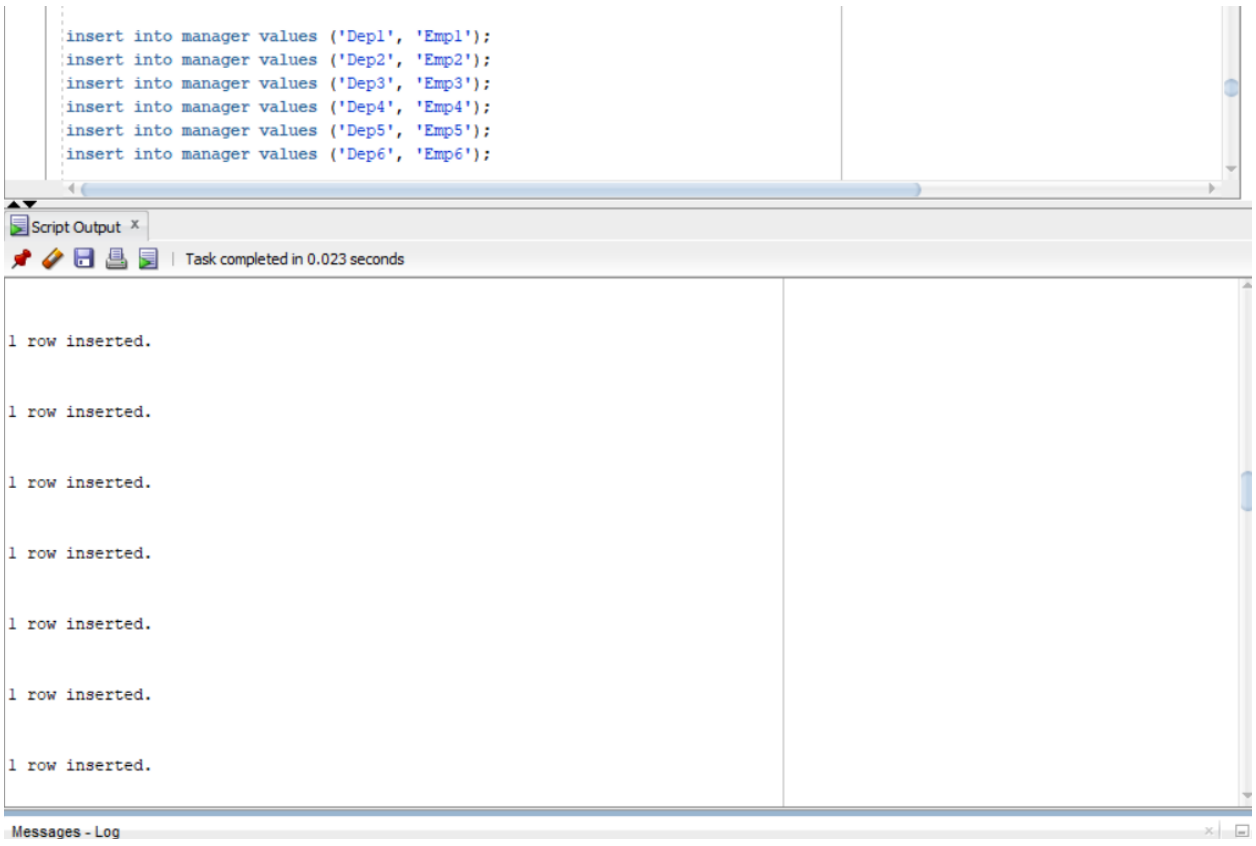
```
insert into job VALUES ('Job1', 'UI and UX', 25000, 30000);
insert into job VALUES ('Job2', 'Accounting', 20000, 25000);
insert into job VALUES ('Job3', 'Web Developer', 17000, 20000);
insert into job VALUES ('Job4', 'Android Developer', 15000, 20000);
insert into job VALUES ('Job5', 'Services', 15000, 20000);


insert into role VALUES ('Role1', 'Job3', 'Backend Developer ', 'Related with backend of the software(Web)');
```

```
Connected.

1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

Figure 17 Job table insert statement

9.8. Manager table insert statement:

```
insert into manager values ('Dep1', 'Emp1');
insert into manager values ('Dep2', 'Emp2');
insert into manager values ('Dep3', 'Emp3');
insert into manager values ('Dep4', 'Emp4');
insert into manager values ('Dep5', 'Emp5');
insert into manager values ('Dep6', 'Emp6');
```



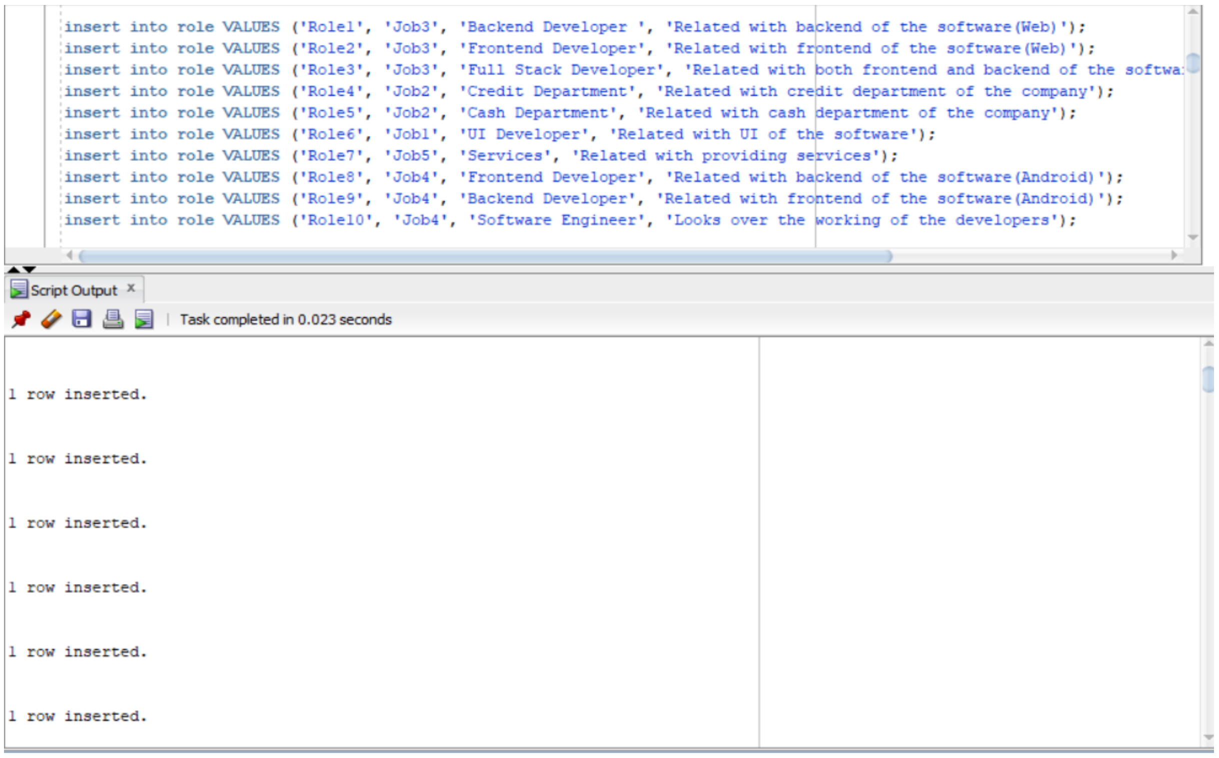
```
insert into manager values ('Dep1', 'Emp1');
insert into manager values ('Dep2', 'Emp2');
insert into manager values ('Dep3', 'Emp3');
insert into manager values ('Dep4', 'Emp4');
insert into manager values ('Dep5', 'Emp5');
insert into manager values ('Dep6', 'Emp6');
```

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

Figure 18 Manager table insert statement

9.9. Role table insert statement:

```
insert into role VALUES ('Role1', 'Job3', 'Backend Developer ', 'Related with backend of the software(Web)');
insert into role VALUES ('Role2', 'Job3', 'Frontend Developer', 'Related with frontend of the software(Web)');
insert into role VALUES ('Role3', 'Job3', 'Full Stack Developer', 'Related with both frontend and backend of the software(Web)');
insert into role VALUES ('Role4', 'Job2', 'Credit Department', 'Related with credit department of the company');
insert into role VALUES ('Role5', 'Job2', 'Cash Department', 'Related with cash department of the company');
insert into role VALUES ('Role6', 'Job1', 'UI Developer', 'Related with UI of the software');
insert into role VALUES ('Role7', 'Job5', 'Services', 'Related with providing services');
insert into role VALUES ('Role8', 'Job4', 'Frontend Developer', 'Related with backend of the software(Android)');
insert into role VALUES ('Role9', 'Job4', 'Backend Developer', 'Related with frontend of the software(Android)');
insert into role VALUES ('Role10', 'Job4', 'Software Engineer', 'Looks over the working of the developers');
```



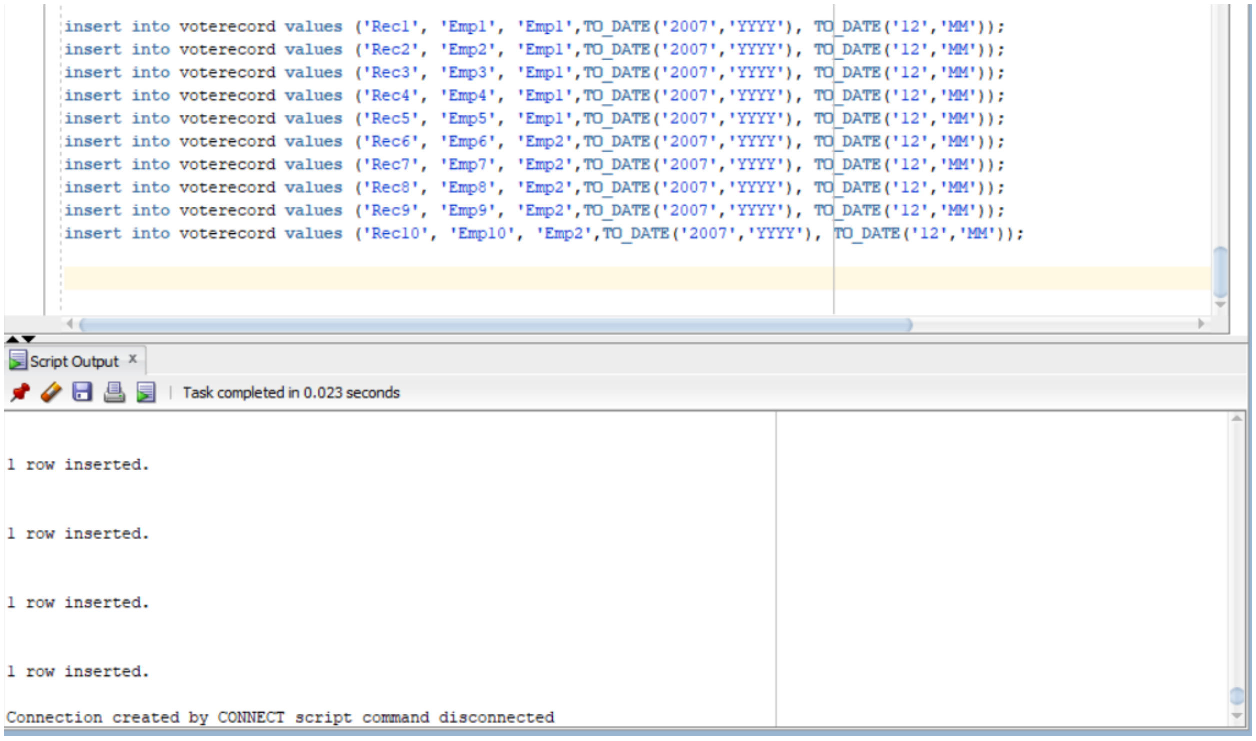
```
insert into role VALUES ('Role1', 'Job3', 'Backend Developer ', 'Related with backend of the software(Web)');
insert into role VALUES ('Role2', 'Job3', 'Frontend Developer', 'Related with frontend of the software(Web)');
insert into role VALUES ('Role3', 'Job3', 'Full Stack Developer', 'Related with both frontend and backend of the softwa
insert into role VALUES ('Role4', 'Job2', 'Credit Department', 'Related with credit department of the company');
insert into role VALUES ('Role5', 'Job2', 'Cash Department', 'Related with cash department of the company');
insert into role VALUES ('Role6', 'Job1', 'UI Developer', 'Related with UI of the software');
insert into role VALUES ('Role7', 'Job5', 'Services', 'Related with providing services');
insert into role VALUES ('Role8', 'Job4', 'Frontend Developer', 'Related with backend of the software(Android)');
insert into role VALUES ('Role9', 'Job4', 'Backend Developer', 'Related with frontend of the software(Android)');
insert into role VALUES ('Role10', 'Job4', 'Software Engineer', 'Looks over the working of the developers');
```

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

Figure 19 Role table insert statement

9.10. VoteRecord table insert statement:

```
insert into voterecord values ('Rec1', 'Emp1', 'Emp1',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec2', 'Emp2', 'Emp1',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec3', 'Emp3', 'Emp1',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec4', 'Emp4', 'Emp1',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec5', 'Emp5', 'Emp1',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec6', 'Emp6', 'Emp2',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec7', 'Emp7', 'Emp2',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec8', 'Emp8', 'Emp2',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec9', 'Emp9', 'Emp2',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
insert into voterecord values ('Rec10', 'Emp10', 'Emp2',TO_DATE('2007','YYYY'),
TO_DATE('12','MM'));
```



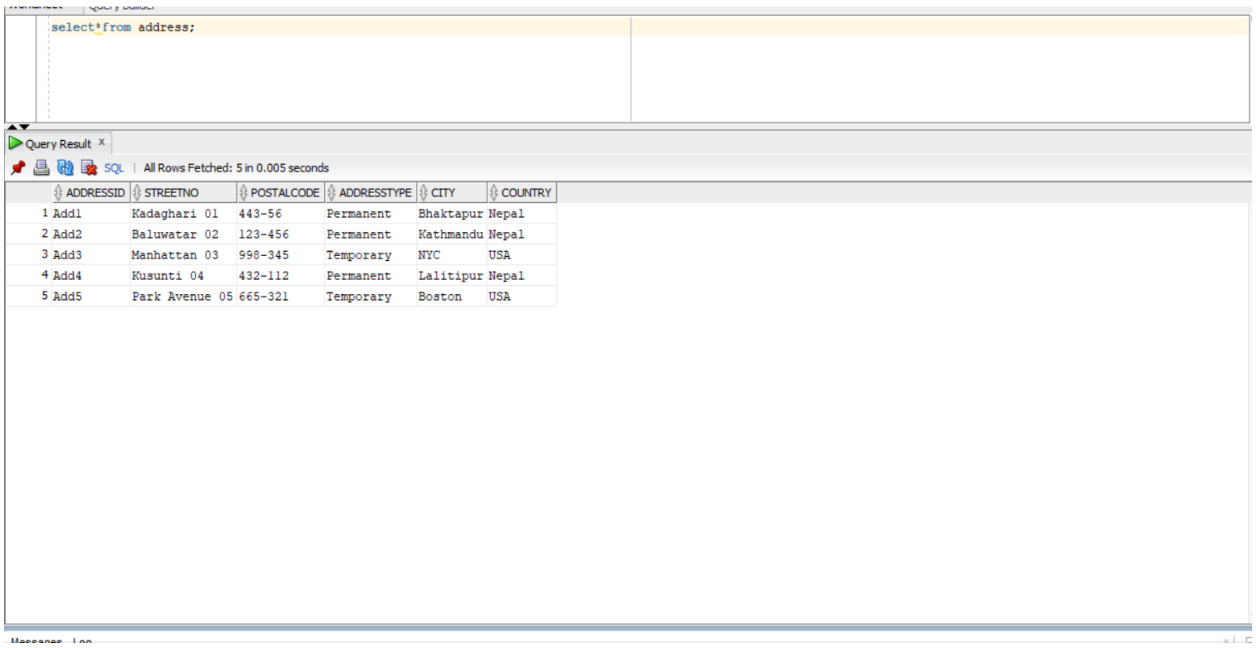
```
insert into voterecord values ('Rec1', 'Emp1', 'Emp1',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec2', 'Emp2', 'Emp1',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec3', 'Emp3', 'Emp1',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec4', 'Emp4', 'Emp1',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec5', 'Emp5', 'Emp1',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec6', 'Emp6', 'Emp2',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec7', 'Emp7', 'Emp2',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec8', 'Emp8', 'Emp2',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec9', 'Emp9', 'Emp2',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
insert into voterecord values ('Rec10', 'Emp10', 'Emp2',TO_DATE('2007','YYYY'), TO_DATE('12','MM'));
```

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.

Connection created by CONNECT script command disconnected
```

Figure 20 VoteRecord table insert statement

10. Select statement:

10.1. Address table select statement:



```
select*from address;
```

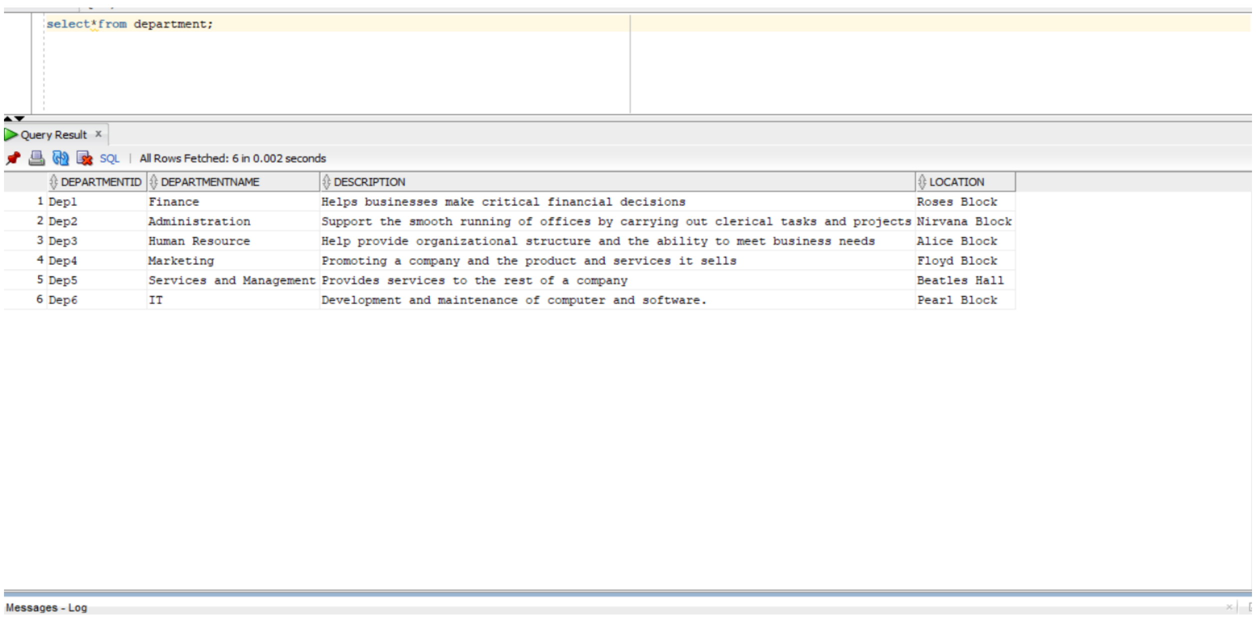
Query Result

All Rows Fetched: 5 in 0.005 seconds

```
    ADDRESSID  STREETNO        POSTALCODE  ADDRESSTYPE  CITY       COUNTRY
  1 Add1       Kadaghari 01    443-56      Permanent    Bhaktapur  Nepal
  2 Add2       Baluwatar 02    123-456     Permanent    Kathmandu  Nepal
  3 Add3       Manhattan 03    998-345     Temporary    NYC        USA
  4 Add4       Kusunti 04      432-112     Permanent    Lalitipur  Nepal
  5 Add5       Park Avenue 05  665-321     Temporary    Boston     USA
```

Figure 21 Address table select statement

10.2. Department table select statement:



```
select*from department;
```

Query Result

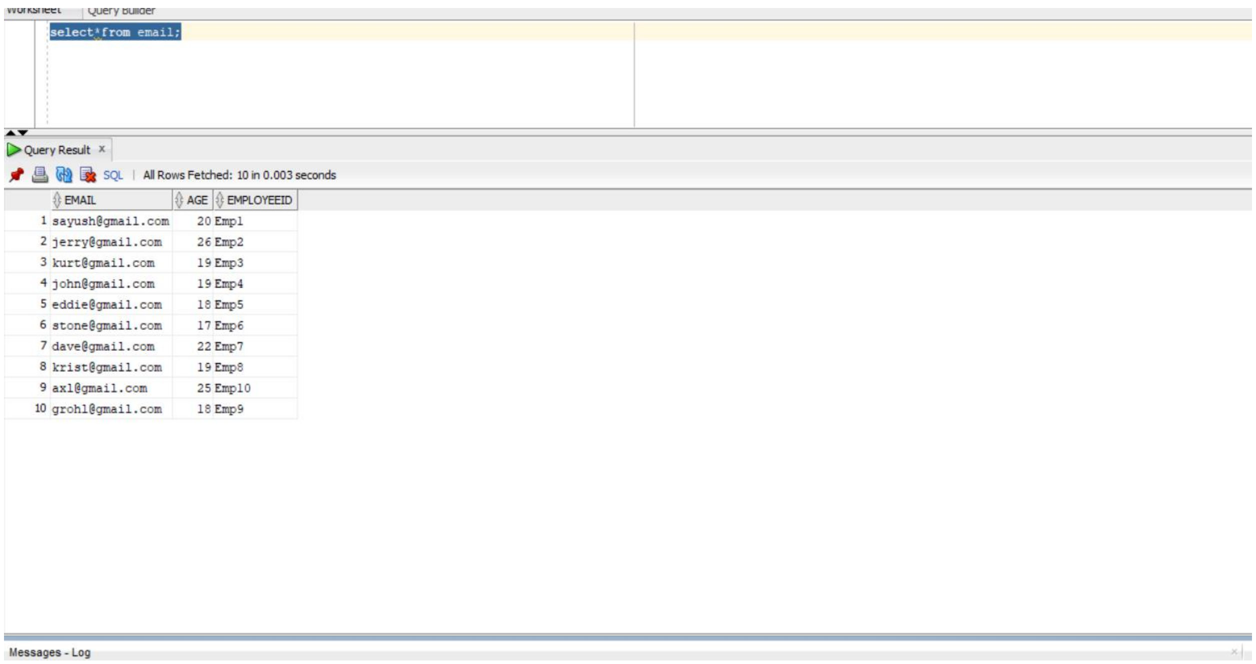
All Rows Fetched: 6 in 0.002 seconds

```
    DEPARTMENTID  DEPARTMENTNAME           DESCRIPTION                                                         LOCATION
  1 Dep1          Finance                  Helps businesses make critical financial decisions                  Roses Block
  2 Dep2          Administration           Support the smooth running of offices by carrying out clerical tasks and projects Nirvana Block
  3 Dep3          Human Resource           Help provide organizational structure and the ability to meet business needs  Alice Block
  4 Dep4          Marketing                Promoting a company and the product and services it sells           Floyd Block
  5 Dep5          Services and Management  Provides services to the rest of a company                          Beatles Hall
  6 Dep6          IT                       Development and maintenance of computer and software.               Pearl Block
```

Messages - Log

Figure 22 Department table select statement

10.3. Email table select statement:



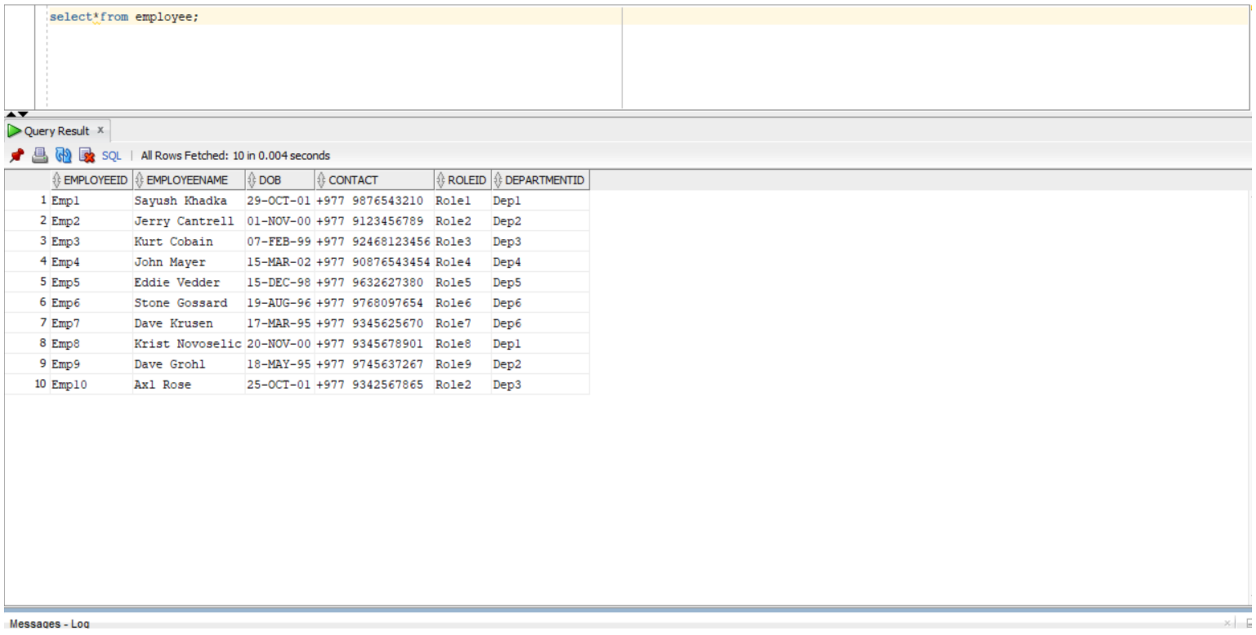
```
select*from email;
```

All Rows Fetched: 10 in 0.003 seconds

	EMAIL	AGE	EMPLOYEEID
1	sayush@gmail.com	20	Emp1
2	jerry@gmail.com	26	Emp2
3	kurt@gmail.com	19	Emp3
4	john@gmail.com	19	Emp4
5	eddie@gmail.com	18	Emp5
6	stone@gmail.com	17	Emp6
7	dave@gmail.com	22	Emp7
8	krist@gmail.com	19	Emp8
9	axl@gmail.com	25	Emp10
10	grohl@gmail.com	18	Emp9

Figure 23 Email table select statement

10.4. Employee table select statement:



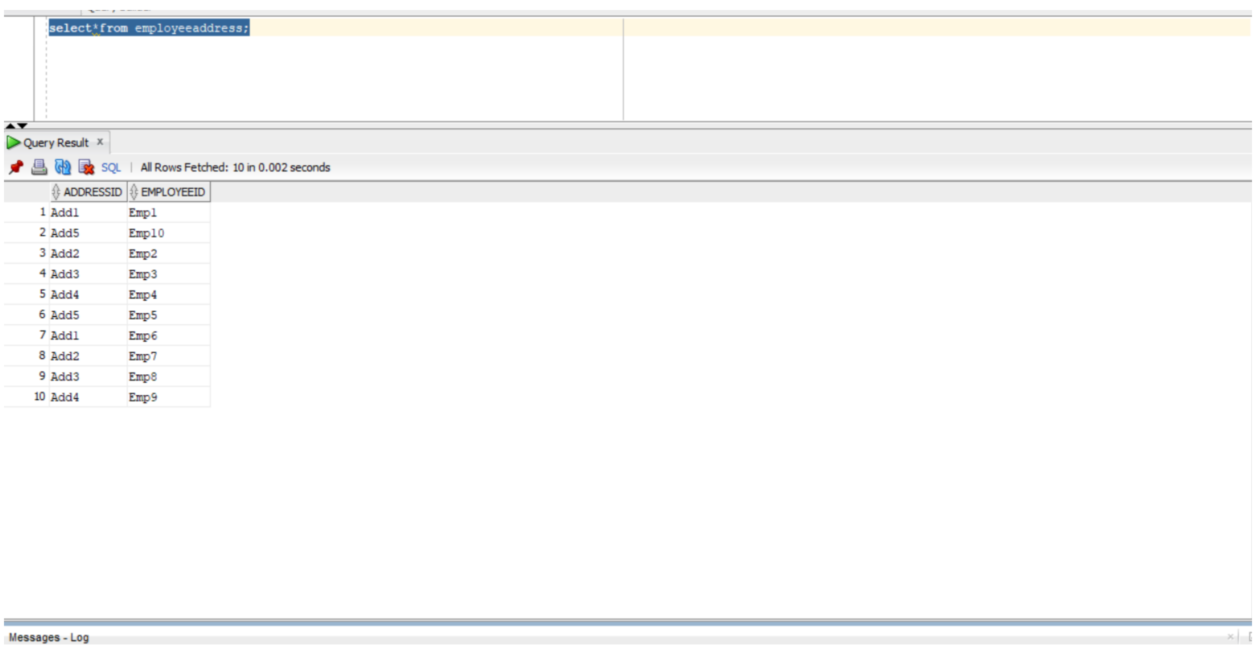
```
select*from employee;
```

All Rows Fetched: 10 in 0.004 seconds

	EMPLOYEEID	EMPLOYEENAME	DOB	CONTACT	ROLEID	DEPARTMENTID
1	Emp1	Sayush Khadka	29-OCT-01	+977 9876543210	Role1	Dep1
2	Emp2	Jerry Cantrell	01-NOV-00	+977 9123456789	Role2	Dep2
3	Emp3	Kurt Cobain	07-FEB-99	+977 92468123456	Role3	Dep3
4	Emp4	John Mayer	15-MAR-02	+977 90876543454	Role4	Dep4
5	Emp5	Eddie Vedder	15-DEC-98	+977 9632627380	Role5	Dep5
6	Emp6	Stone Gossard	19-AUG-96	+977 9768097654	Role6	Dep6
7	Emp7	Dave Krusen	17-MAR-95	+977 9345625670	Role7	Dep6
8	Emp8	Krist Novoselic	20-NOV-00	+977 9345678901	Role8	Dep1
9	Emp9	Dave Grohl	18-MAY-95	+977 9745637267	Role9	Dep2
10	Emp10	Axl Rose	25-OCT-01	+977 9342567865	Role2	Dep3

Figure 24 Employee table select statement

10.5. EmployeeAddress table select statement:



```
select*from employeeaddress;
```

Query Result

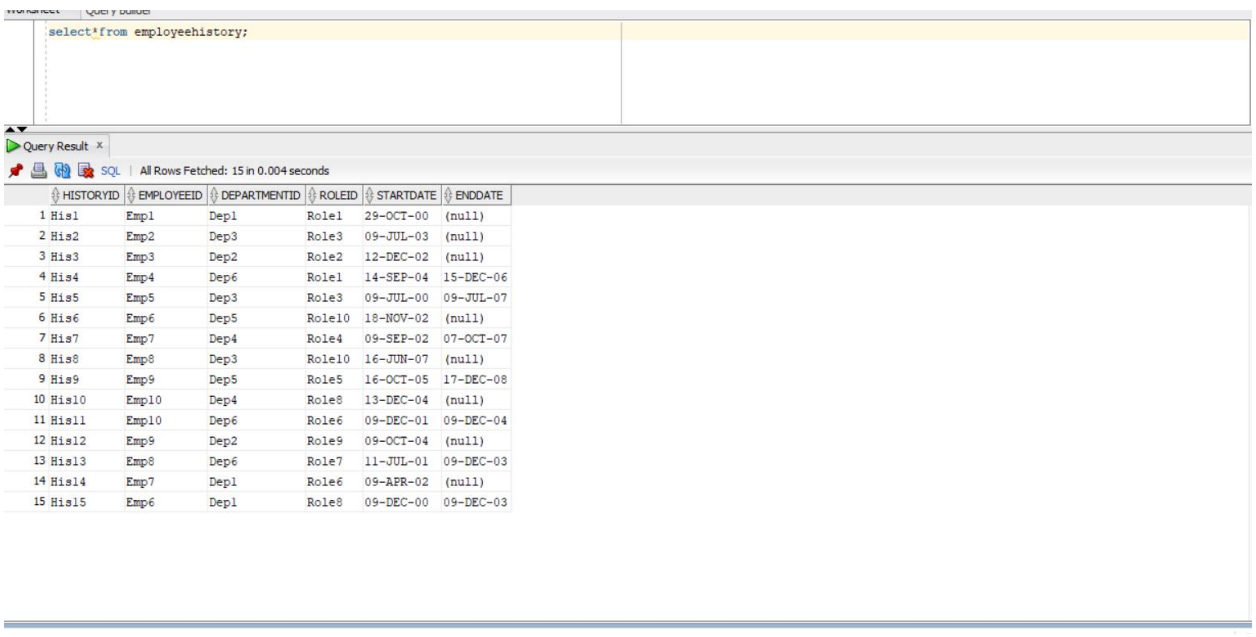
All Rows Fetched: 10 in 0.002 seconds

	ADDRESSID	EMPLOYEEID
1	Add1	Emp1
2	Add5	Emp10
3	Add2	Emp2
4	Add3	Emp3
5	Add4	Emp4
6	Add5	Emp5
7	Add1	Emp6
8	Add2	Emp7
9	Add3	Emp8
10	Add4	Emp9

Messages - Log

Figure 25 EmployeeAddress table select statement

10.6. EmployeeHistory table select statement:



```
select*from employeehistory;
```

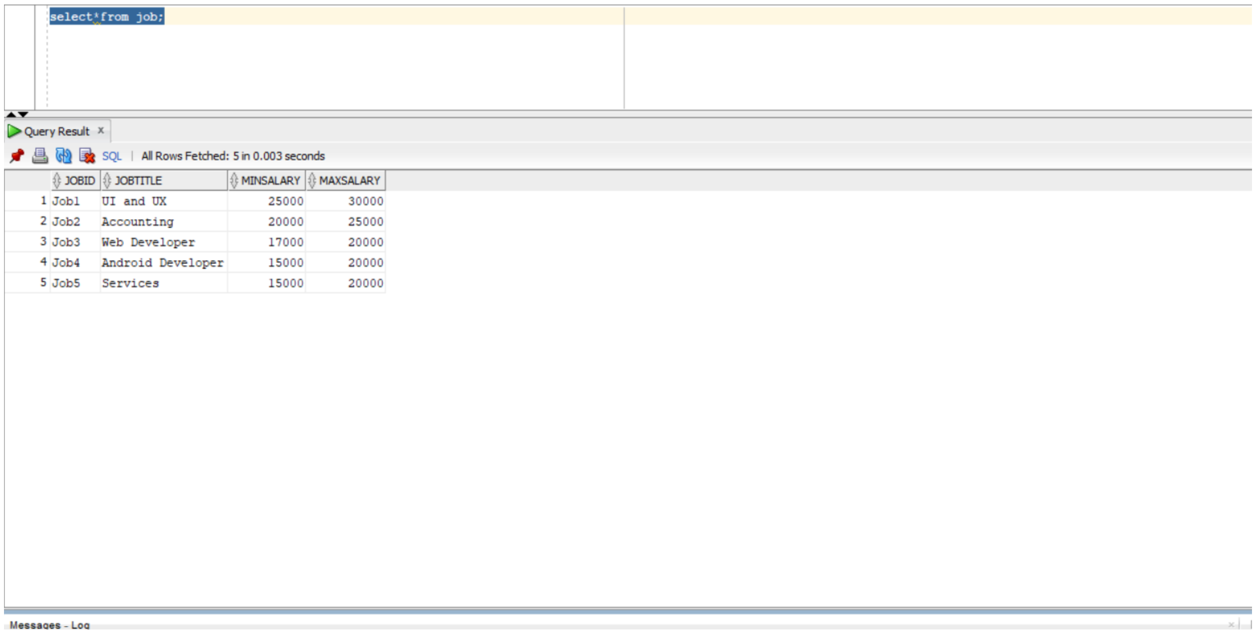
Query Result

All Rows Fetched: 15 in 0.004 seconds

	HISTORYID	EMPLOYEEID	DEPARTMENTID	ROLEID	STARTDATE	ENDDATE
1	His1	Emp1	Dep1	Role1	29-OCT-00	(null)
2	His2	Emp2	Dep3	Role3	09-JUL-03	(null)
3	His3	Emp3	Dep2	Role2	12-DEC-02	(null)
4	His4	Emp4	Dep6	Role1	14-SEP-04	15-DEC-06
5	His5	Emp5	Dep3	Role3	09-JUL-00	09-JUL-07
6	His6	Emp6	Dep5	Role10	18-NOV-02	(null)
7	His7	Emp7	Dep4	Role4	09-SEP-02	07-OCT-07
8	His8	Emp8	Dep3	Role10	16-JUN-07	(null)
9	His9	Emp9	Dep5	Role5	16-OCT-05	17-DEC-08
10	His10	Emp10	Dep4	Role8	13-DEC-04	(null)
11	His11	Emp10	Dep6	Role6	09-DEC-01	09-DEC-04
12	His12	Emp9	Dep2	Role9	09-OCT-04	(null)
13	His13	Emp8	Dep6	Role7	11-JUL-01	09-DEC-03
14	His14	Emp7	Dep1	Role6	09-APR-02	(null)
15	His15	Emp6	Dep1	Role8	09-DEC-00	09-DEC-03

Figure 26 EmployeeHistory table select statement

10.7. Job table select statement:



```
select*from job;
```

Query Result

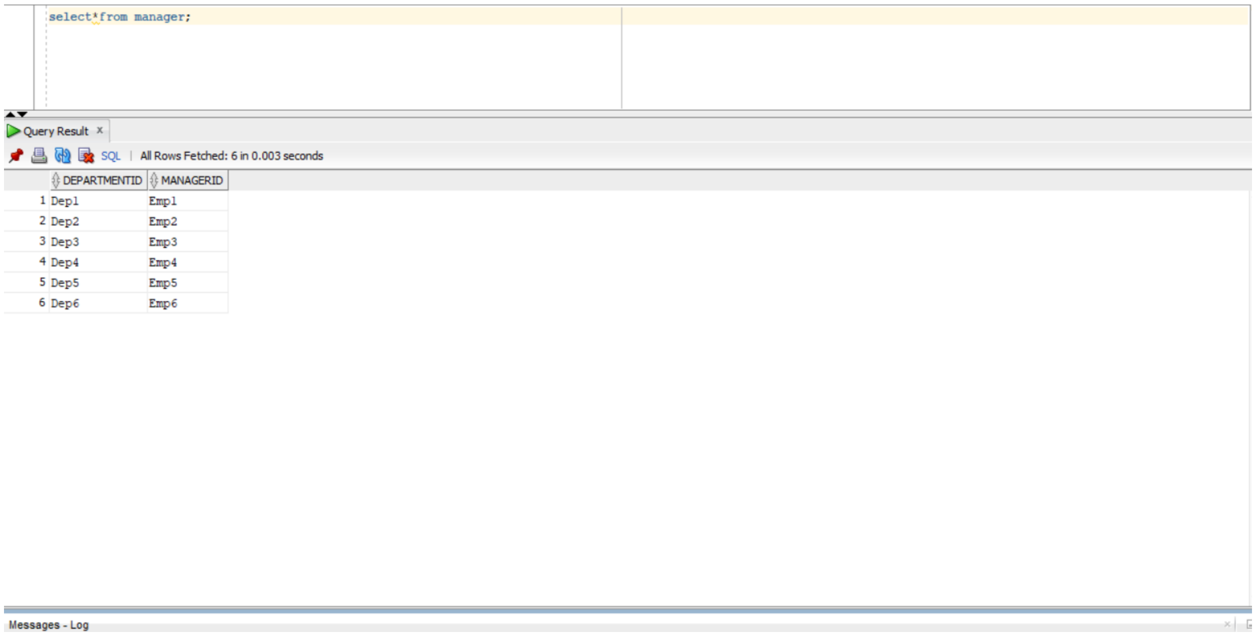
SQL | All Rows Fetched: 5 in 0.003 seconds

```
   JOBID JOBTITLE          MINSALARY MAXSALARY
 1 Job1  UI and UX             25000     30000
 2 Job2  Accounting            20000     25000
 3 Job3  Web Developer         17000     20000
 4 Job4  Android Developer     15000     20000
 5 Job5  Services              15000     20000
```

Messages - Log

Figure 27 Job table select statement

10.8. Manager table select statement:



```
select*from manager;
```

Query Result

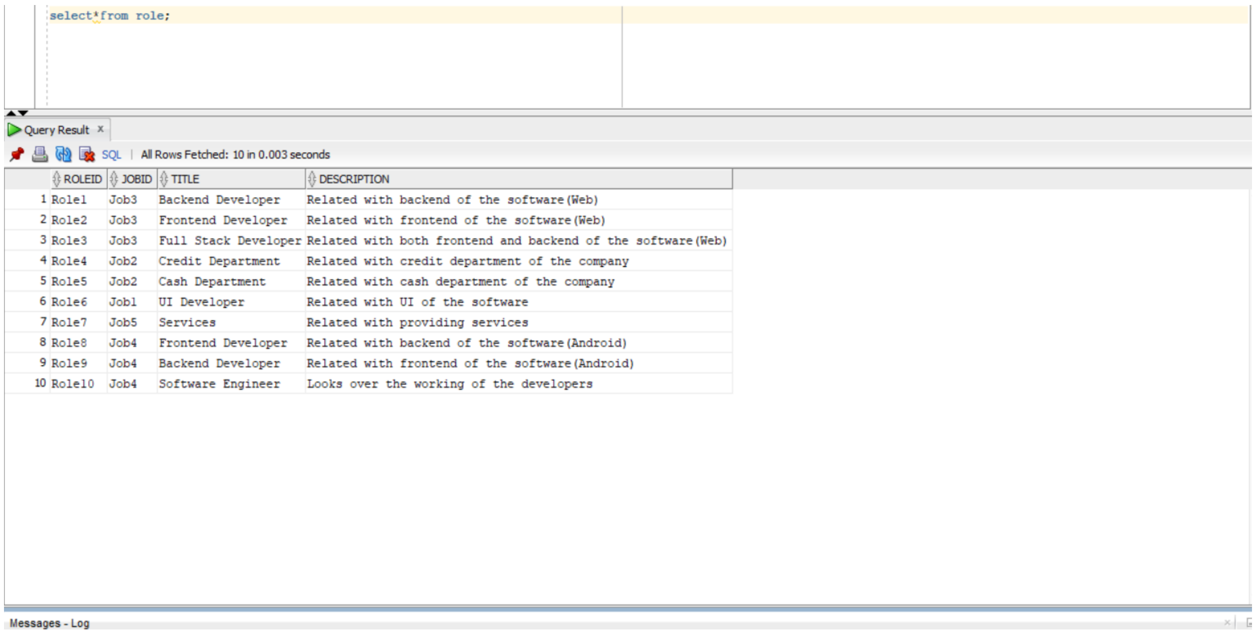
SQL | All Rows Fetched: 6 in 0.003 seconds

```
   DEPARTMENTID MANAGERID
 1 Dep1         Emp1
 2 Dep2         Emp2
 3 Dep3         Emp3
 4 Dep4         Emp4
 5 Dep5         Emp5
 6 Dep6         Emp6
```

Messages - Log

Figure 28 Manager table select statement

10.9. Role table select statement:



```
select*from role;
```

Query Result ×

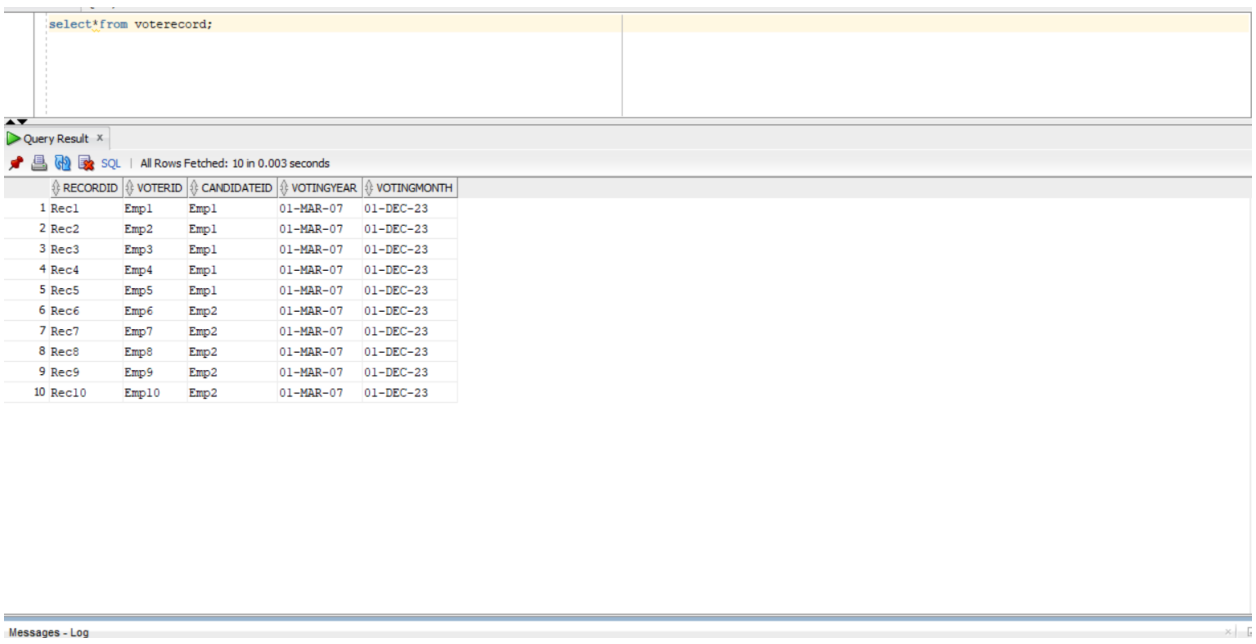
SQL | All Rows Fetched: 10 in 0.003 seconds

	ROLEID	JOBID	TITLE	DESCRIPTION
1	Role1	Job3	Backend Developer	Related with backend of the software(Web)
2	Role2	Job3	Frontend Developer	Related with frontend of the software(Web)
3	Role3	Job3	Full Stack Developer	Related with both frontend and backend of the software(Web)
4	Role4	Job2	Credit Department	Related with credit department of the company
5	Role5	Job2	Cash Department	Related with cash department of the company
6	Role6	Job1	UI Developer	Related with UI of the software
7	Role7	Job5	Services	Related with providing services
8	Role8	Job4	Frontend Developer	Related with backend of the software(Android)
9	Role9	Job4	Backend Developer	Related with frontend of the software(Android)
10	Role10	Job4	Software Engineer	Looks over the working of the developers

Messages - Log

Figure 29 Role table select statement

10.10. VoteRecord table select statement:



```
select*from voterecord;
```

Query Result ×

SQL | All Rows Fetched: 10 in 0.003 seconds

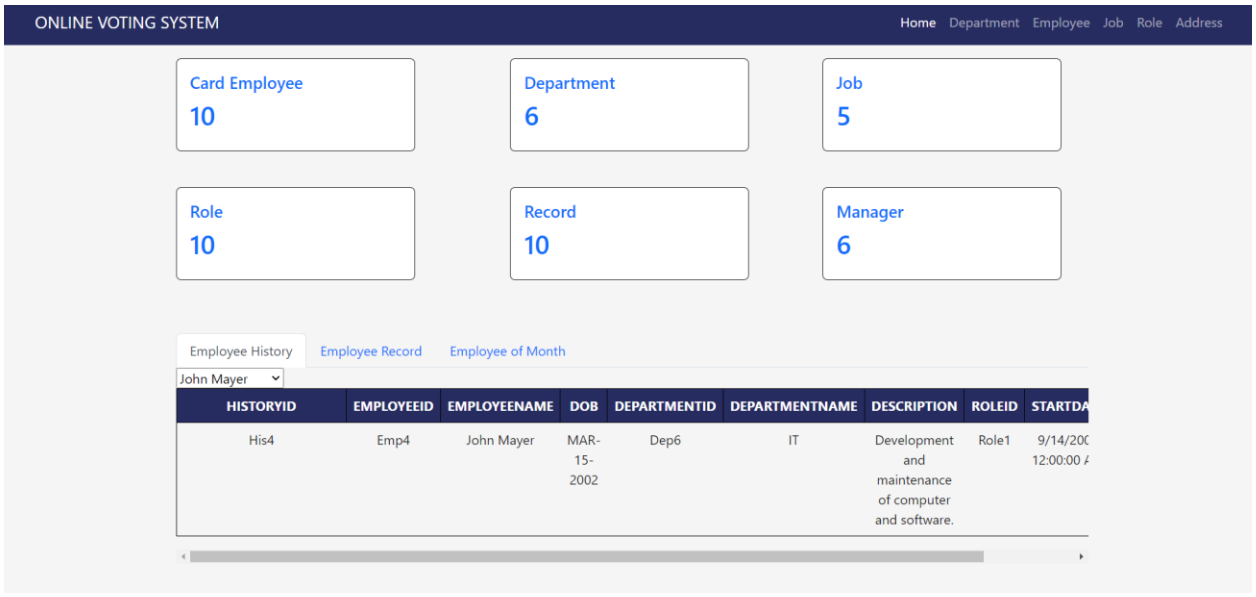
	RECORDID	VOTERID	CANDIDATEID	VOTINGYEAR	VOTINGMONTH
1	Rec1	Emp1	Emp1	01-MAR-07	01-DEC-23
2	Rec2	Emp2	Emp1	01-MAR-07	01-DEC-23
3	Rec3	Emp3	Emp1	01-MAR-07	01-DEC-23
4	Rec4	Emp4	Emp1	01-MAR-07	01-DEC-23
5	Rec5	Emp5	Emp1	01-MAR-07	01-DEC-23
6	Rec6	Emp6	Emp2	01-MAR-07	01-DEC-23
7	Rec7	Emp7	Emp2	01-MAR-07	01-DEC-23
8	Rec8	Emp8	Emp2	01-MAR-07	01-DEC-23
9	Rec9	Emp9	Emp2	01-MAR-07	01-DEC-23
10	Rec10	Emp10	Emp2	01-MAR-07	01-DEC-23

Messages - Log

Figure 30 VoteRecord table select statement

11. Forms:

11.1. Dashboard for home page:



Card Employee	10
Department	6
Job	5
Role	10
Record	10
Manager	6

HISTORYID	EMPLOYEEID	EMPLOYEENAME	DOB	DEPARTMENTID	DEPARTMENTNAME	DESCRIPTION	ROLEID	STARTDA
His4	Emp4	John Mayer	MAR-15-2002	Dep6	IT	Development and maintenance of computer and software.	Role1	9/14/200 12:00:00 /

Figure 31 Dashboard for home page

11.2. Complex forms and queries:

11.2.1. SQL Queries:

a) Showing the details of the selected employee:

```
SELECT "EMPLOYEEID", "EMPLOYEENAME" FROM "EMPLOYEE"
```

```
SELECT EH.HISTORYID, EH.EMPLOYEEID, E.EMPLOYEENAME,
TO_CHAR(E.DOB, 'MON-dd-YYYY') AS DOB, EH.DEPARTMENTID,
D.DEPARTMENTNAME, D.DESCRIPTION, EH.ROLEID, EH.STARTDATE,
EH.ENDDATE
FROM EMPLOYEEHISTORY EH, EMPLOYEE E, DEPARTMENT D
WHERE EH.EMPLOYEEID = E.EMPLOYEEID
AND EH.DEPARTMENTID = D.DEPARTMENTID
AND (EH.EMPLOYEEID = :EMPLOYEE)
AND EH.ENDDATE IS NOT NULL
```

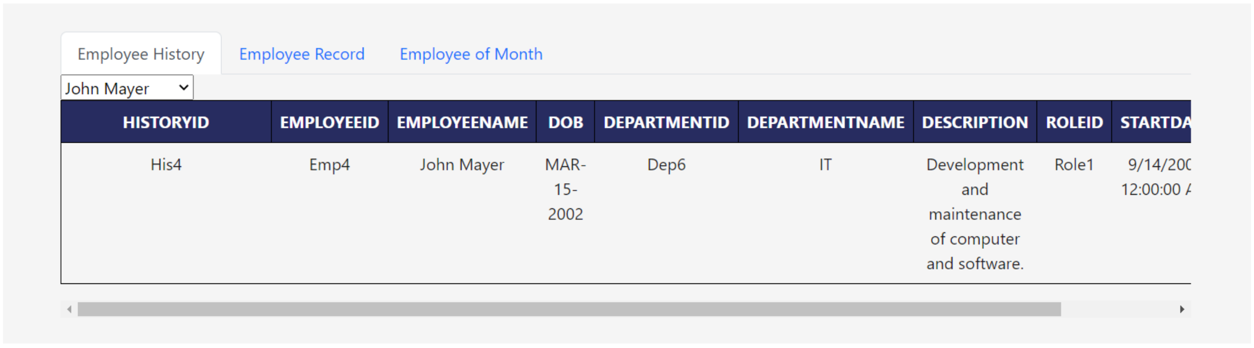
b) Showing the details and the vote record of the selected employee:

```
SELECT "EMPLOYEEID", "EMPLOYEENAME" FROM "EMPLOYEE"
```

```
SELECT VR.RECORDID, VR.VOTERID, E.EMPLOYEENAME, TO_CHAR(E.DOB,
'dd-MON-YYYY') AS DOB, E.CONTACT, VR.CANDIDATEID, C.EMPLOYEENAME AS
CANDIDATENAME, C.CONTACT AS CANDIDATECONTACT,
TO_CHAR(VR.VOTINGYEAR, 'YYYY') AS VOTEYEAR,
TO_CHAR(VR.VOTINGMONTH, 'MON') AS VOTEMONTH
FROM VOTERECORD VR, EMPLOYEE E, EMPLOYEE C
WHERE VR.VOTERID = E.EMPLOYEEID
AND VR.CANDIDATEID = C.EMPLOYEEID
AND VR.VOTERID = :VOTER
```

11.2.2. Complex forms:

a) Showing the details of the selected employee



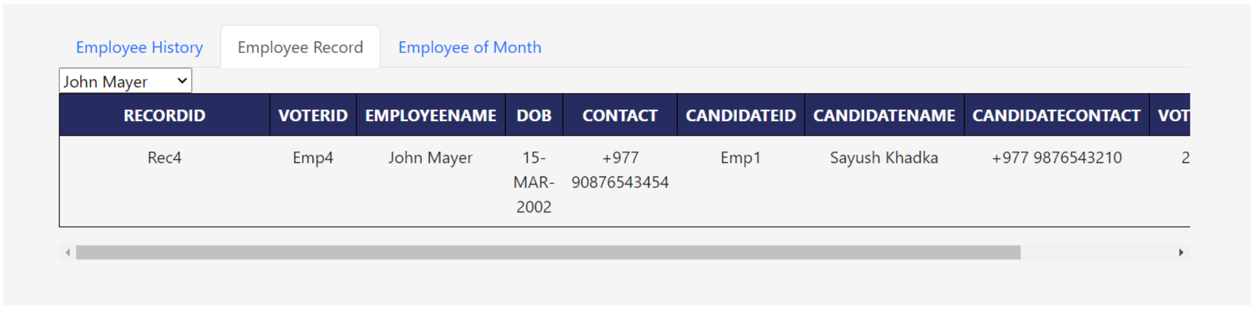
Employee History | Employee Record | Employee of Month

John Mayer

HISTORYID	EMPLOYEEID	EMPLOYEENAME	DOB	DEPARTMENTID	DEPARTMENTNAME	DESCRIPTION	ROLEID	STARTDA
His4	Emp4	John Mayer	MAR-15-2002	Dep6	IT	Development and maintenance of computer and software.	Role1	9/14/200 12:00:00 A

Figure 32 Showing the details of the selected employee

b) Showing the details and the vote record of the selected employee



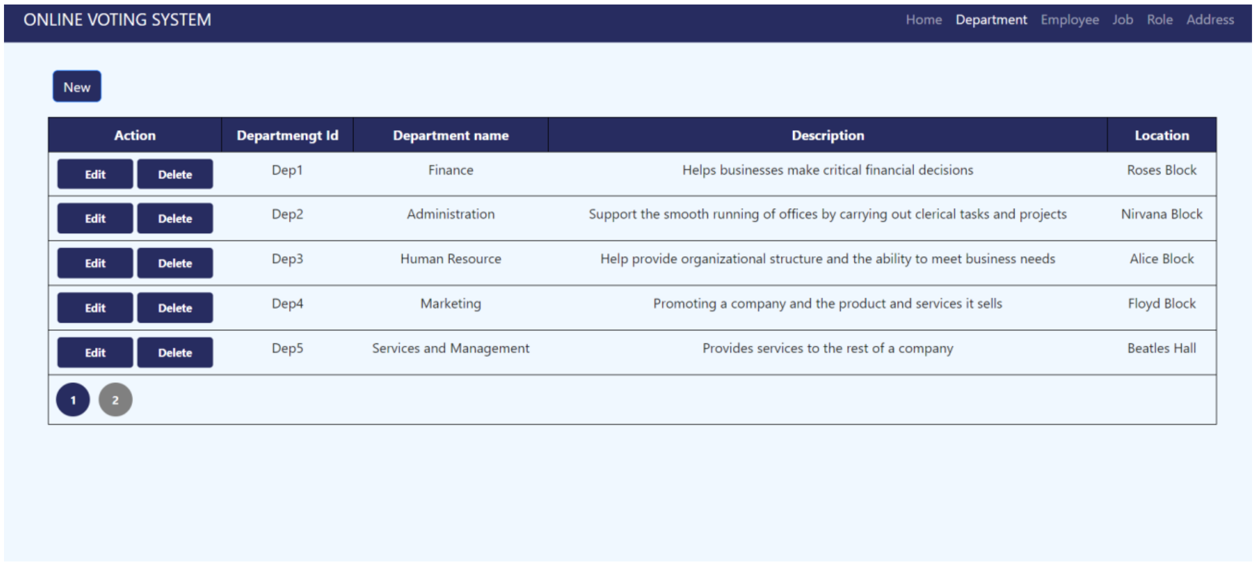
Employee History | Employee Record | Employee of Month

John Mayer

RECORDID	VOTERID	EMPLOYEENAME	DOB	CONTACT	CANDIDATEID	CANDIDATENAME	CANDIDATECONTACT	VOT
Rec4	Emp4	John Mayer	15-MAR-2002	+977 90876543454	Emp1	Sayush Khadka	+977 9876543210	2

Figure 33 Showing the details and the vote record of the selected employee

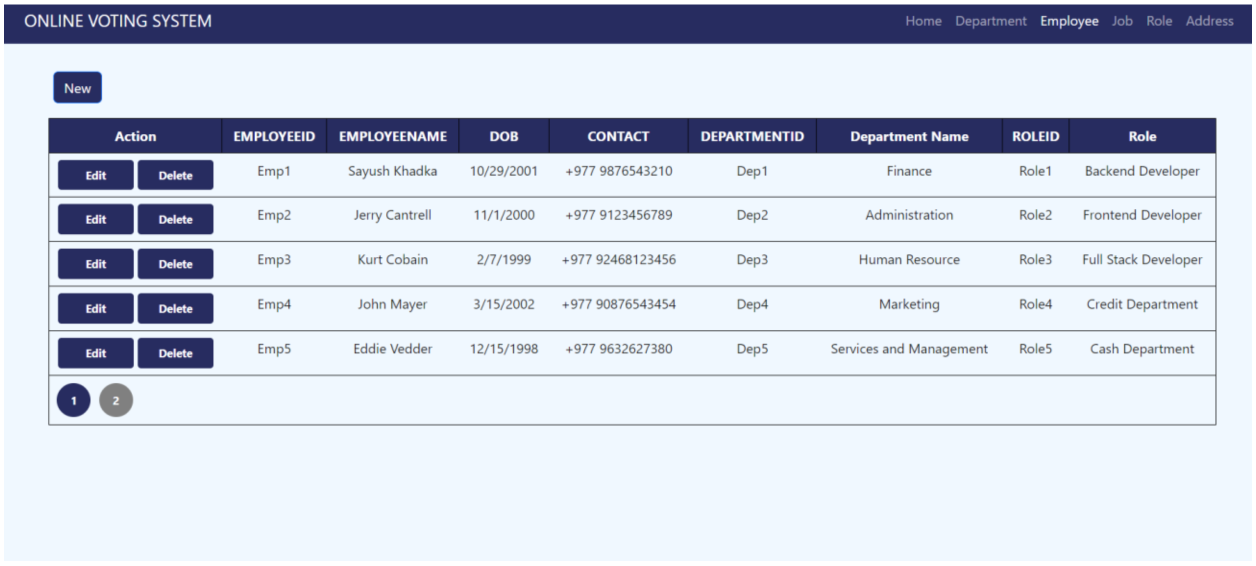
11.3. Simple forms:



ONLINE VOTING SYSTEM

Action	Departmengt Id	Department name	Description	Location
Edit Delete	Dep1	Finance	Helps businesses make critical financial decisions	Roses Block
Edit Delete	Dep2	Administration	Support the smooth running of offices by carrying out clerical tasks and projects	Nirvana Block
Edit Delete	Dep3	Human Resource	Help provide organizational structure and the ability to meet business needs	Alice Block
Edit Delete	Dep4	Marketing	Promoting a company and the product and services it sells	Floyd Block
Edit Delete	Dep5	Services and Management	Provides services to the rest of a company	Beatles Hall

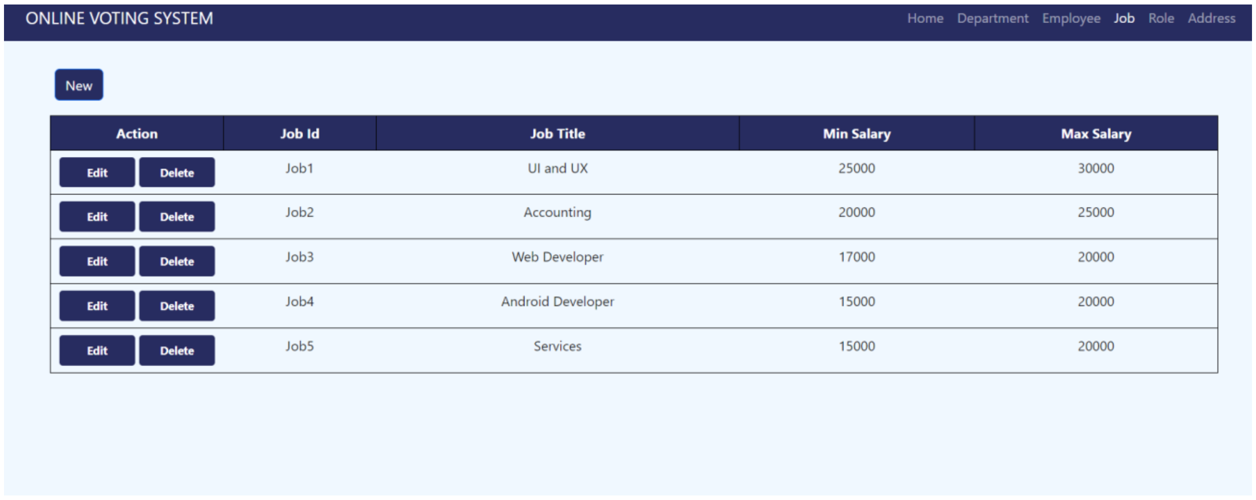
Figure 34 Department web form



ONLINE VOTING SYSTEM

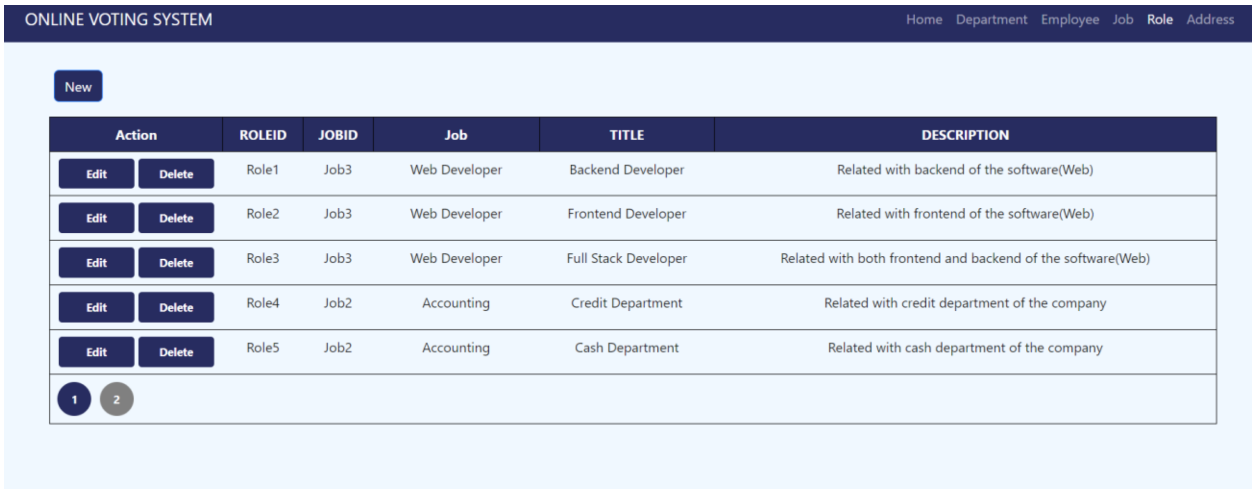
Action	EMPLOYEEID	EMPLOYEENAME	DOB	CONTACT	DEPARTMENTID	Department Name	ROLEID	Role
Edit Delete	Emp1	Sayush Khadka	10/29/2001	+977 9876543210	Dep1	Finance	Role1	Backend Developer
Edit Delete	Emp2	Jerry Cantrell	11/1/2000	+977 9123456789	Dep2	Administration	Role2	Frontend Developer
Edit Delete	Emp3	Kurt Cobain	2/7/1999	+977 92468123456	Dep3	Human Resource	Role3	Full Stack Developer
Edit Delete	Emp4	John Mayer	3/15/2002	+977 90876543454	Dep4	Marketing	Role4	Credit Department
Edit Delete	Emp5	Eddie Vedder	12/15/1998	+977 9632627380	Dep5	Services and Management	Role5	Cash Department

Figure 35 Employee web form


ONLINE VOTING SYSTEM

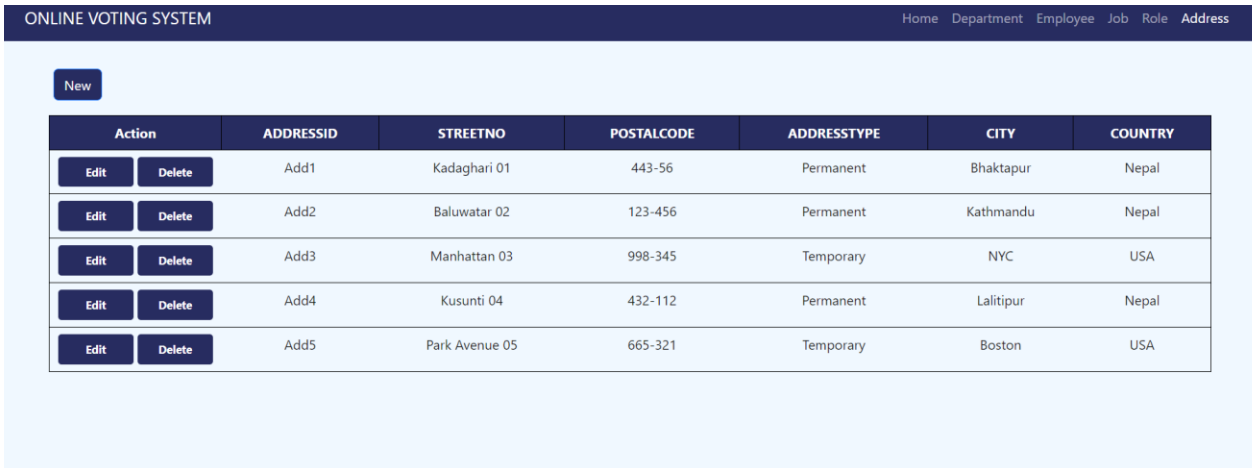
Action	Job Id	Job Title	Min Salary	Max Salary
Edit Delete	Job1	UI and UX	25000	30000
Edit Delete	Job2	Accounting	20000	25000
Edit Delete	Job3	Web Developer	17000	20000
Edit Delete	Job4	Android Developer	15000	20000
Edit Delete	Job5	Services	15000	20000

Figure 36 Job web form


ONLINE VOTING SYSTEM

Action	ROLEID	JOBID	Job	TITLE	DESCRIPTION
Edit Delete	Role1	Job3	Web Developer	Backend Developer	Related with backend of the software(Web)
Edit Delete	Role2	Job3	Web Developer	Frontend Developer	Related with frontend of the software(Web)
Edit Delete	Role3	Job3	Web Developer	Full Stack Developer	Related with both frontend and backend of the software(Web)
Edit Delete	Role4	Job2	Accounting	Credit Department	Related with credit department of the company
Edit Delete	Role5	Job2	Accounting	Cash Department	Related with cash department of the company

Figure 37 Role web form

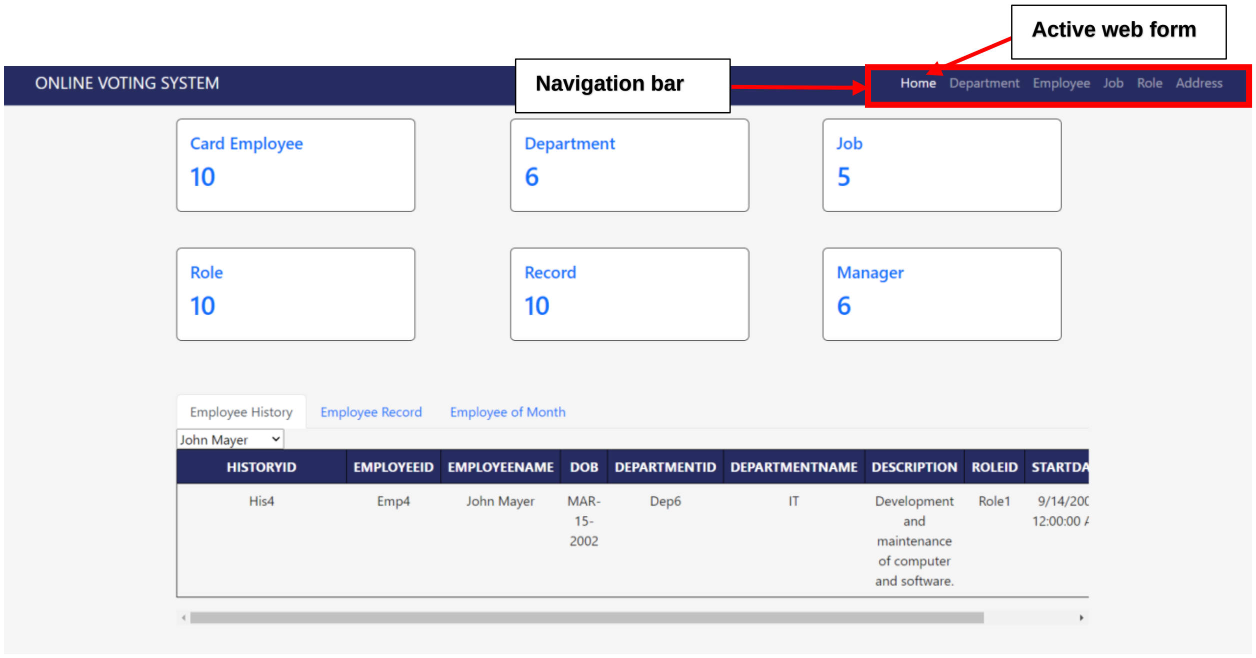


Action	ADDRESSID	STREETNO	POSTALCODE	ADDRESSTYPE	CITY	COUNTRY
Edit Delete	Add1	Kadaghari 01	443-56	Permanent	Bhaktapur	Nepal
Edit Delete	Add2	Baluwatar 02	123-456	Permanent	Kathmandu	Nepal
Edit Delete	Add3	Manhattan 03	998-345	Temporary	NYC	USA
Edit Delete	Add4	Kusunti 04	432-112	Permanent	Lalitipur	Nepal
Edit Delete	Add5	Park Avenue 05	665-321	Temporary	Boston	USA

Figure 38 Address web form

12. User manual:

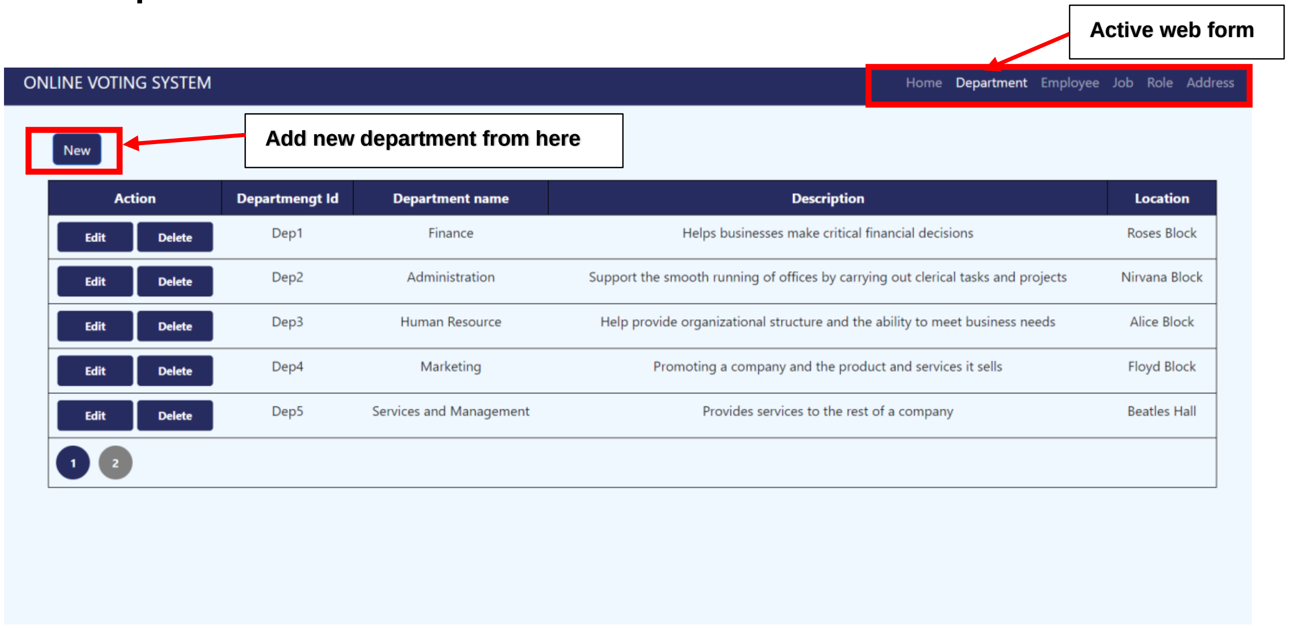
12.1. Dashboard user manual:



HISTORYID	EMPLOYEEID	EMPLOYEENAME	DOB	DEPARTMENTID	DEPARTMENTNAME	DESCRIPTION	ROLEID	STARTDA
His4	Emp4	John Mayer	MAR-15-2002	Dep6	IT	Development and maintenance of computer and software.	Role1	9/14/20C 12:00:00 A

Figure 39 Dashboard user manual

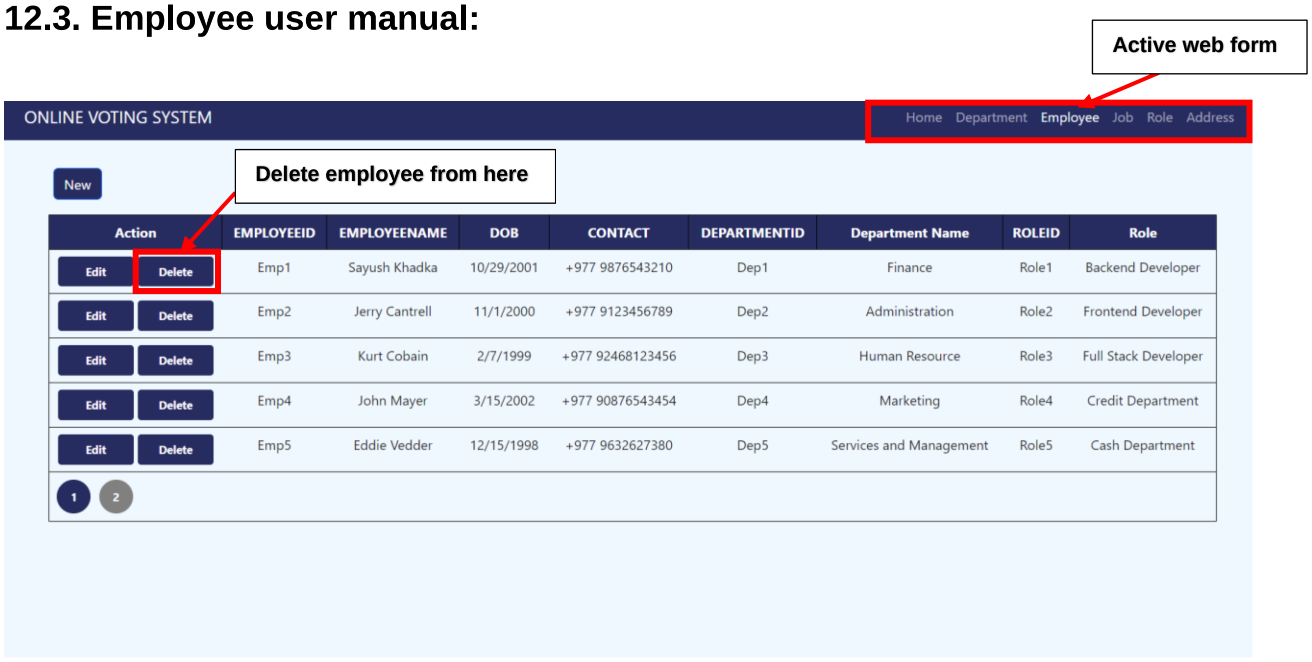
12.2. Department user manual:


Active web form
ONLINE VOTING SYSTEM
Home Department Employee Job Role Address
New
Add new department from here
1
2

Action	Departmengt Id	Department name	Description	Location
Edit Delete	Dep1	Finance	Helps businesses make critical financial decisions	Roses Block
Edit Delete	Dep2	Administration	Support the smooth running of offices by carrying out clerical tasks and projects	Nirvana Block
Edit Delete	Dep3	Human Resource	Help provide organizational structure and the ability to meet business needs	Alice Block
Edit Delete	Dep4	Marketing	Promoting a company and the product and services it sells	Floyd Block
Edit Delete	Dep5	Services and Management	Provides services to the rest of a company	Beatles Hall

Figure 40 Department user manual

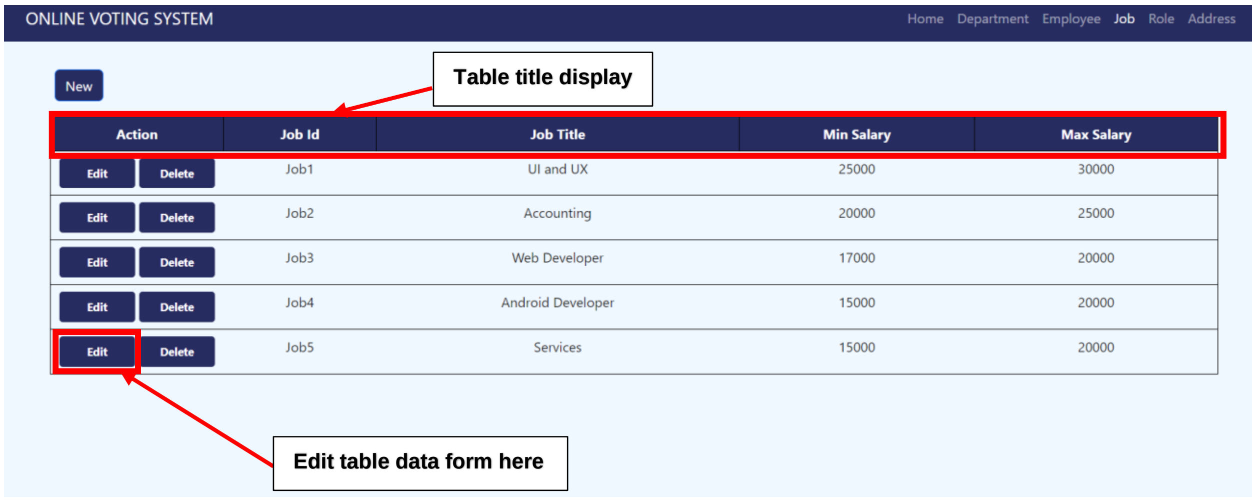
12.3. Employee user manual:


Active web form
ONLINE VOTING SYSTEM
Home Department Employee Job Role Address
New
Delete employee from here

Action	EMPLOYEEID	EMPLOYEENAME	DOB	CONTACT	DEPARTMENTID	Department Name	ROLEID	Role
Edit Delete	Emp1	Sayush Khadka	10/29/2001	+977 9876543210	Dep1	Finance	Role1	Backend Developer
Edit Delete	Emp2	Jerry Cantrell	11/1/2000	+977 9123456789	Dep2	Administration	Role2	Frontend Developer
Edit Delete	Emp3	Kurt Cobain	2/7/1999	+977 92468123456	Dep3	Human Resource	Role3	Full Stack Developer
Edit Delete	Emp4	John Mayer	3/15/2002	+977 90876543454	Dep4	Marketing	Role4	Credit Department
Edit Delete	Emp5	Eddie Vedder	12/15/1998	+977 9632627380	Dep5	Services and Management	Role5	Cash Department

Figure 41 Employee user manual

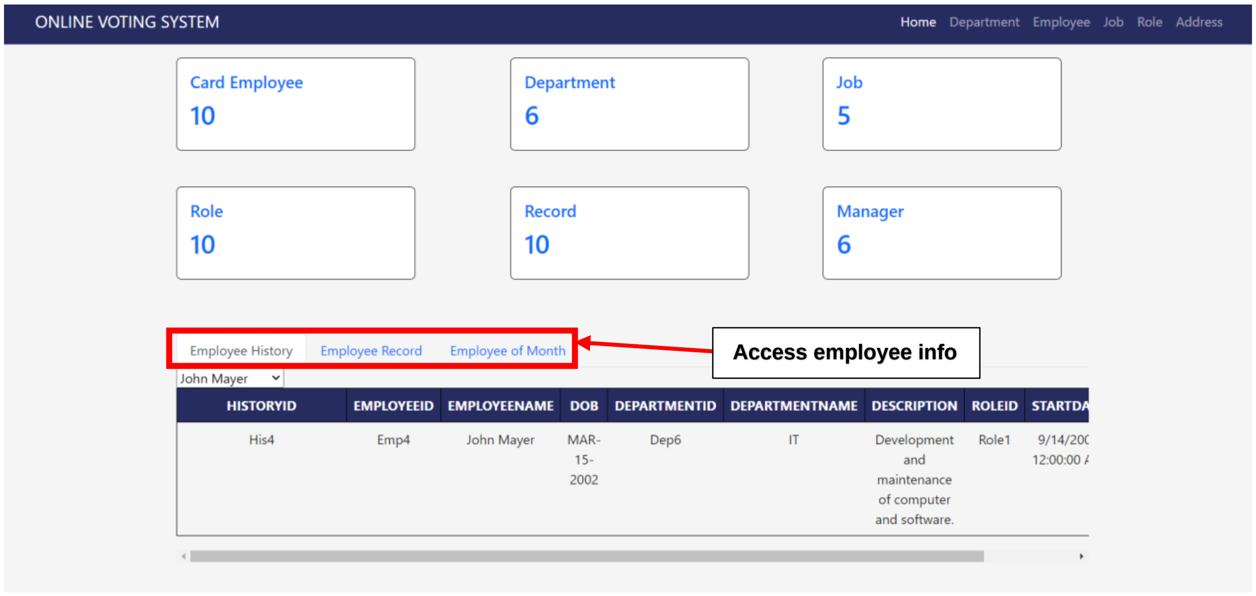
12.4. Job user manual:


ONLINE VOTING SYSTEM
Home Department Employee Job Role Address
New
Table title display
Edit table data form here

Action	Job Id	Job Title	Min Salary	Max Salary
Edit Delete	Job1	UI and UX	25000	30000
Edit Delete	Job2	Accounting	20000	25000
Edit Delete	Job3	Web Developer	17000	20000
Edit Delete	Job4	Android Developer	15000	20000
Edit Delete	Job5	Services	15000	20000

Figure 42 Job user manual

12.5. Access Employee info through complex form:


ONLINE VOTING SYSTEM

Home Department Employee Job Role Address

Card Employee 10	Department 6	Job 5
Role 10	Record 10	Manager 6

Employee History | Employee Record | Employee of Month

Access employee info

John Mayer

HISTORYID	EMPLOYEEID	EMPLOYEENAME	DOB	DEPARTMENTID	DEPARTMENTNAME	DESCRIPTION	ROLEID	STARTDA
His4	Emp4	John Mayer	MAR-15-2002	Dep6	IT	Development and maintenance of computer and software.	Role1	9/14/200 12:00:00

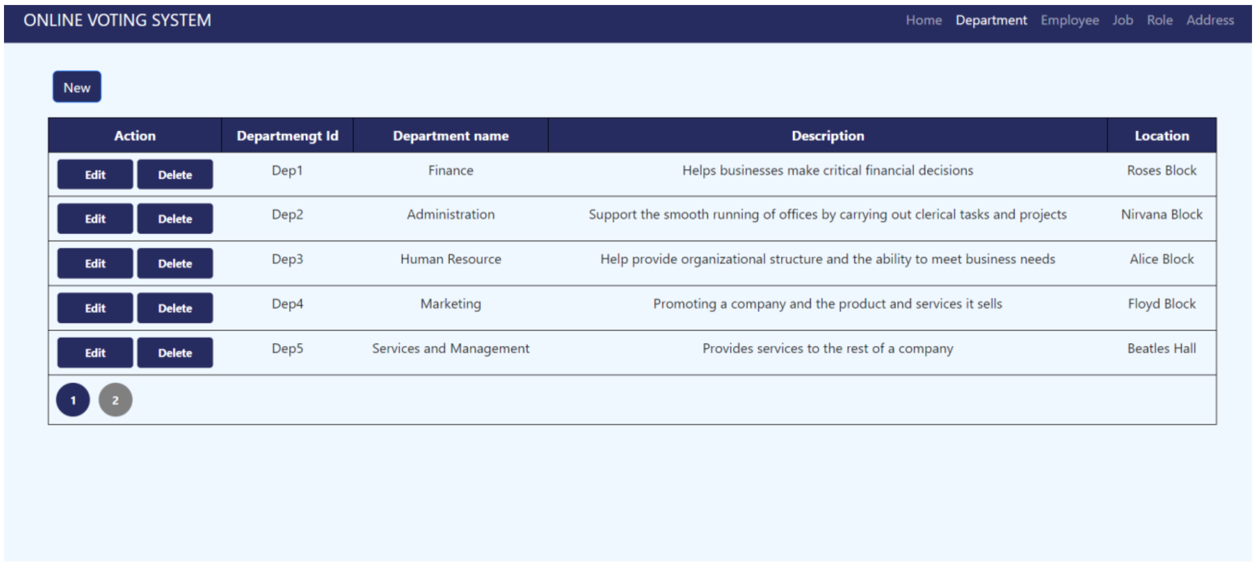
Figure 43 Access employee info through complex form

13. Testing:

13.1. Test Case: To add department to Department table

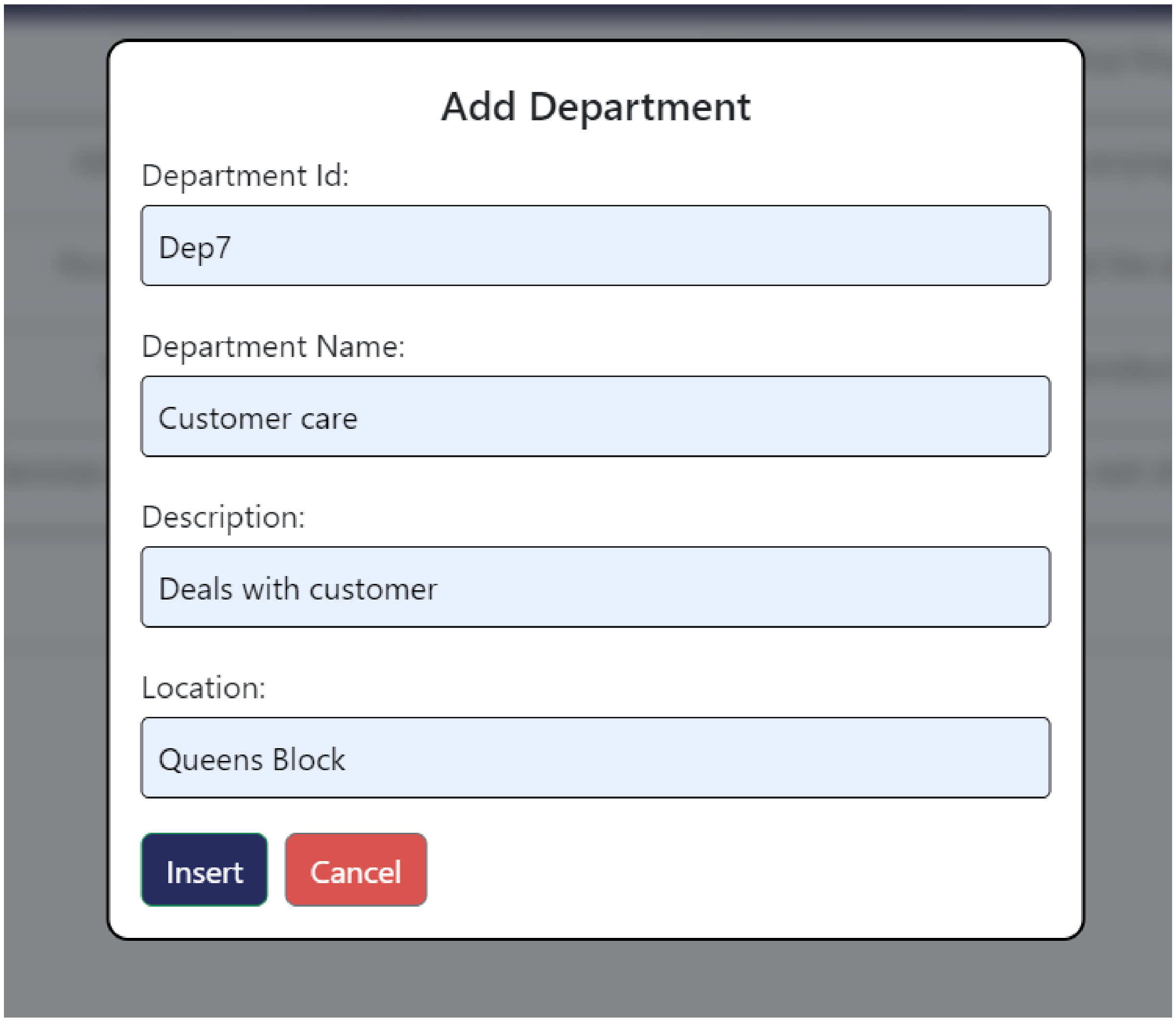
Test No	1
Objective	To add department to Department table.
Action	i) Click on new button. ii) Fill department information. iiI) Click insert button.
Expected Result	New department should be added.
Actual Result	New department was added.
Conclusion	Test successful.

Table 11 To add department to Department table



Action	Departmengt Id	Department name	Description	Location
Edit Delete	Dep1	Finance	Helps businesses make critical financial decisions	Roses Block
Edit Delete	Dep2	Administration	Support the smooth running of offices by carrying out clerical tasks and projects	Nirvana Block
Edit Delete	Dep3	Human Resource	Help provide organizational structure and the ability to meet business needs	Alice Block
Edit Delete	Dep4	Marketing	Promoting a company and the product and services it sells	Floyd Block
Edit Delete	Dep5	Services and Management	Provides services to the rest of a company	Beatles Hall

Figure 44 Department Table (web form)



Add Department

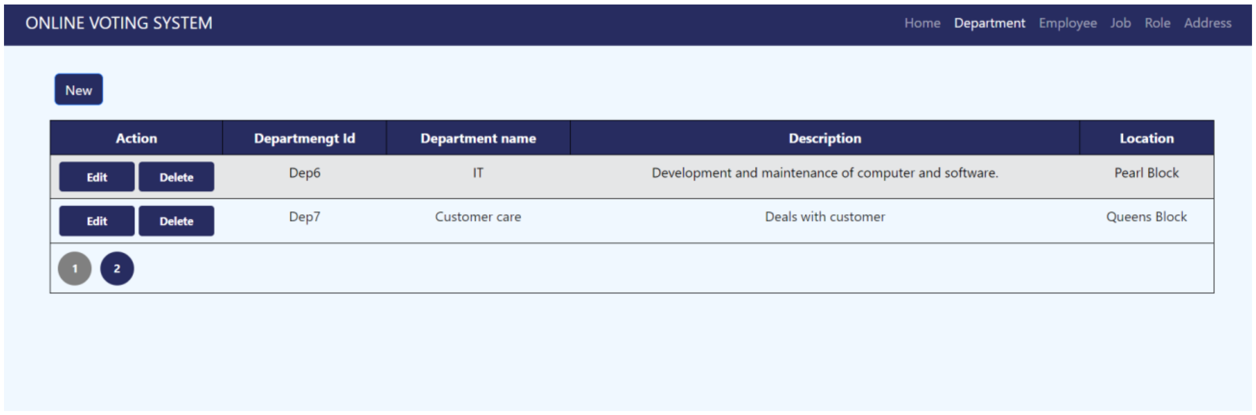
Department Id: Dep7

Department Name: Customer care

Description: Deals with customer

Location: Queens Block

Figure 45 Filling department information



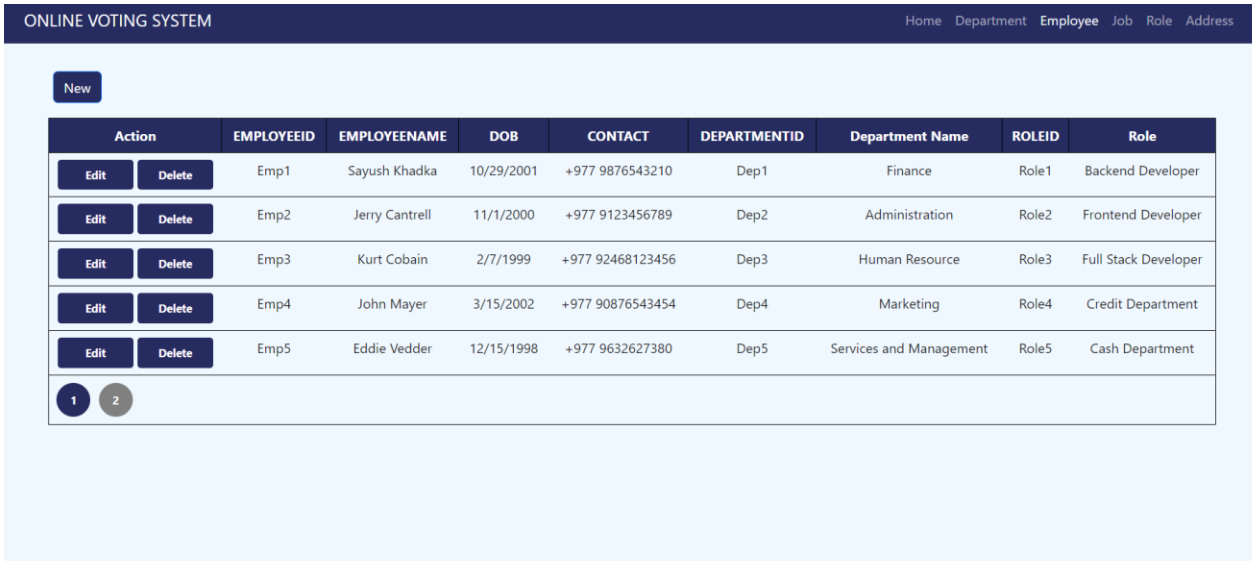
Action	Departmengt Id	Department name	Description	Location
Edit Delete	Dep6	IT	Development and maintenance of computer and software.	Pearl Block
Edit Delete	Dep7	Customer care	Deals with customer	Queens Block

Figure 46 New department added (Test successful)

13.2. Test Case: To remove an employee from the Employee table:

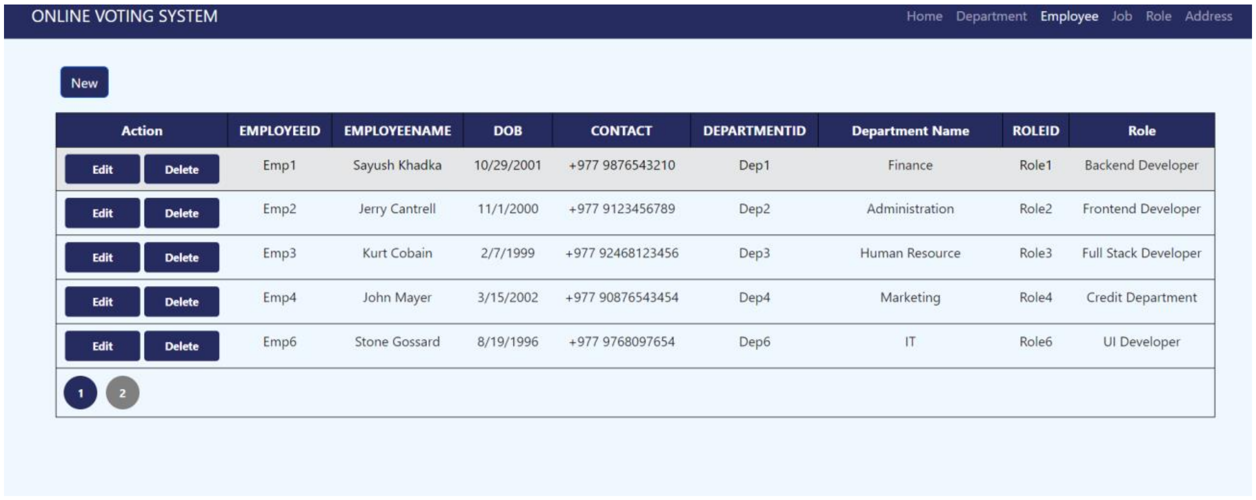
Test No	2
Objective	To remove and employee from the Employee table.
Action	Click on delete button.
Expected Result	The employee will be removed.
Actual Result	The employee was removed.
Conclusion	Test successful.

Table 12 To remove an employee from the Employee table



Action	EMPLOYEEID	EMPLOYEENAME	DOB	CONTACT	DEPARTMENTID	Department Name	ROLEID	Role
Edit Delete	Emp1	Sayush Khadka	10/29/2001	+977 9876543210	Dep1	Finance	Role1	Backend Developer
Edit Delete	Emp2	Jerry Cantrell	11/1/2000	+977 9123456789	Dep2	Administration	Role2	Frontend Developer
Edit Delete	Emp3	Kurt Cobain	2/7/1999	+977 92468123456	Dep3	Human Resource	Role3	Full Stack Developer
Edit Delete	Emp4	John Mayer	3/15/2002	+977 90876543454	Dep4	Marketing	Role4	Credit Department
Edit Delete	Emp5	Eddie Vedder	12/15/1998	+977 9632627380	Dep5	Services and Management	Role5	Cash Department

Figure 47 Employee table before delete



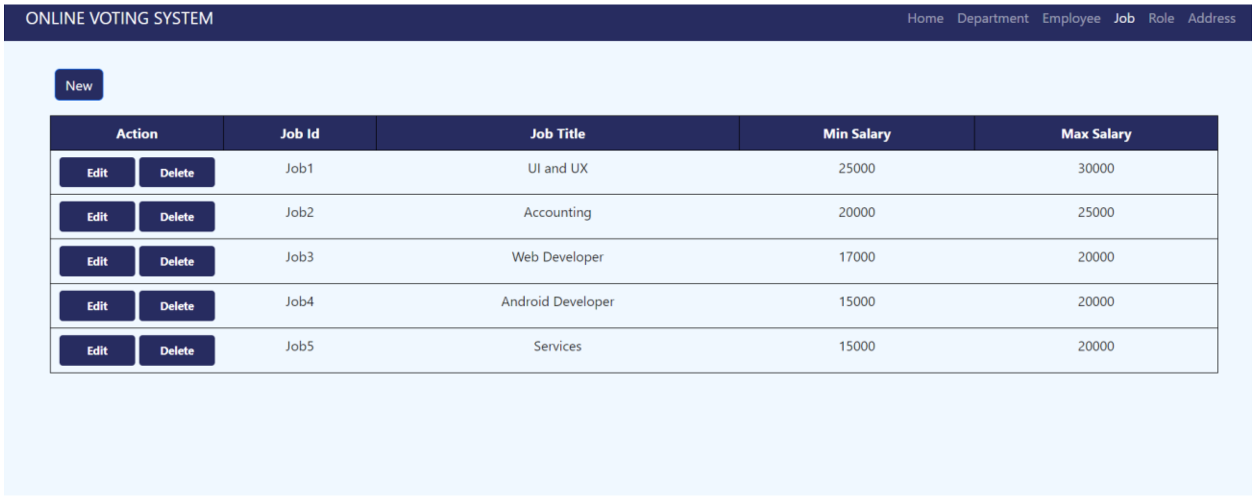
Action	EMPLOYEEID	EMPLOYEENAME	DOB	CONTACT	DEPARTMENTID	Department Name	ROLEID	Role
Edit Delete	Emp1	Sayush Khadka	10/29/2001	+977 9876543210	Dep1	Finance	Role1	Backend Developer
Edit Delete	Emp2	Jerry Cantrell	11/1/2000	+977 9123456789	Dep2	Administration	Role2	Frontend Developer
Edit Delete	Emp3	Kurt Cobain	2/7/1999	+977 92468123456	Dep3	Human Resource	Role3	Full Stack Developer
Edit Delete	Emp4	John Mayer	3/15/2002	+977 90876543454	Dep4	Marketing	Role4	Credit Department
Edit Delete	Emp6	Stone Gossard	8/19/1996	+977 9768097654	Dep6	IT	Role6	UI Developer

Figure 48 Employee table after delete (Test successful)

13.3. Test Case: Update name of Services in Job table:

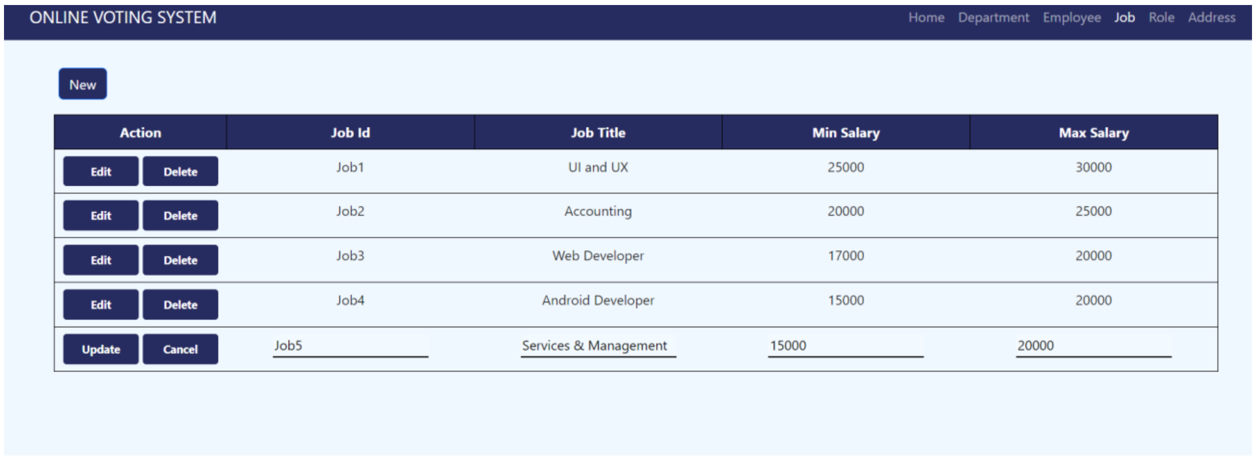
Test No	3
Objective	To update name of Services in Job table.
Action	i) Click on edit button. ii) Change name of Services.
Expected Result	The name of Services will be changed.
Actual Result	The names of Services was changed.
Conclusion	Test successful.

Table 13 Update name of Services in Job table



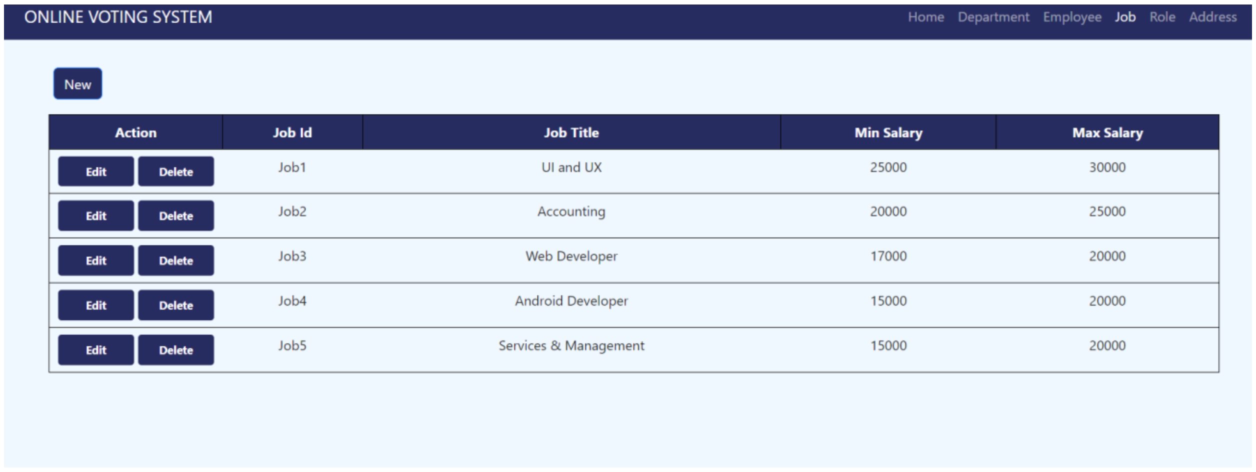
Action	Job Id	Job Title	Min Salary	Max Salary
Edit Delete	Job1	UI and UX	25000	30000
Edit Delete	Job2	Accounting	20000	25000
Edit Delete	Job3	Web Developer	17000	20000
Edit Delete	Job4	Android Developer	15000	20000
Edit Delete	Job5	Services	15000	20000

Figure 49 Job table before update


ONLINE VOTING SYSTEM

Action	Job Id	Job Title	Min Salary	Max Salary
Edit Delete	Job1	UI and UX	25000	30000
Edit Delete	Job2	Accounting	20000	25000
Edit Delete	Job3	Web Developer	17000	20000
Edit Delete	Job4	Android Developer	15000	20000
Update Cancel	Job5	Services & Management	15000	20000

Figure 50 Changing the name of Services to Services & Management


ONLINE VOTING SYSTEM

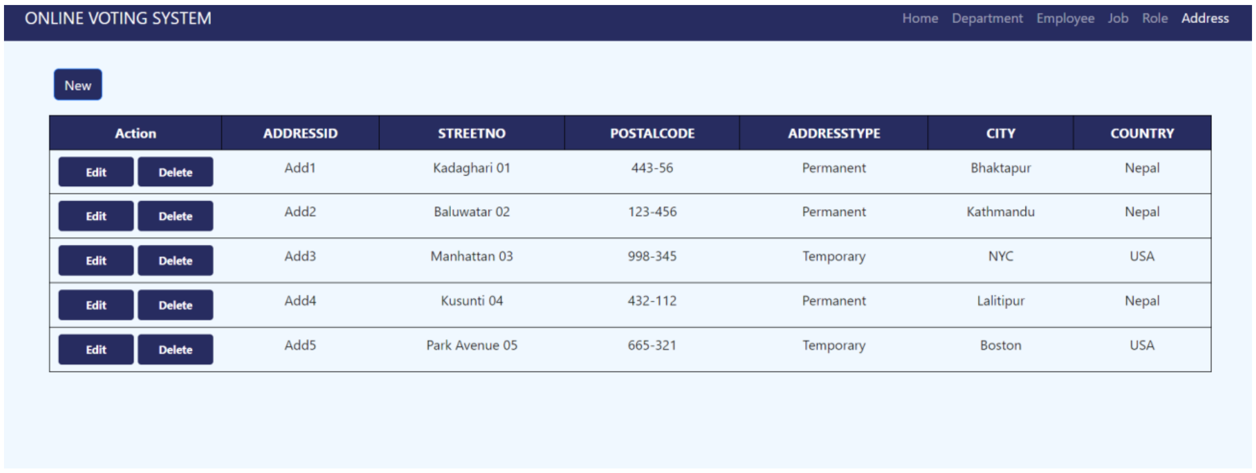
Action	Job Id	Job Title	Min Salary	Max Salary
Edit Delete	Job1	UI and UX	25000	30000
Edit Delete	Job2	Accounting	20000	25000
Edit Delete	Job3	Web Developer	17000	20000
Edit Delete	Job4	Android Developer	15000	20000
Edit Delete	Job5	Services & Management	15000	20000

Figure 51 Job table updated (Test successful)

13.4. Test Case: To add address to Address table:

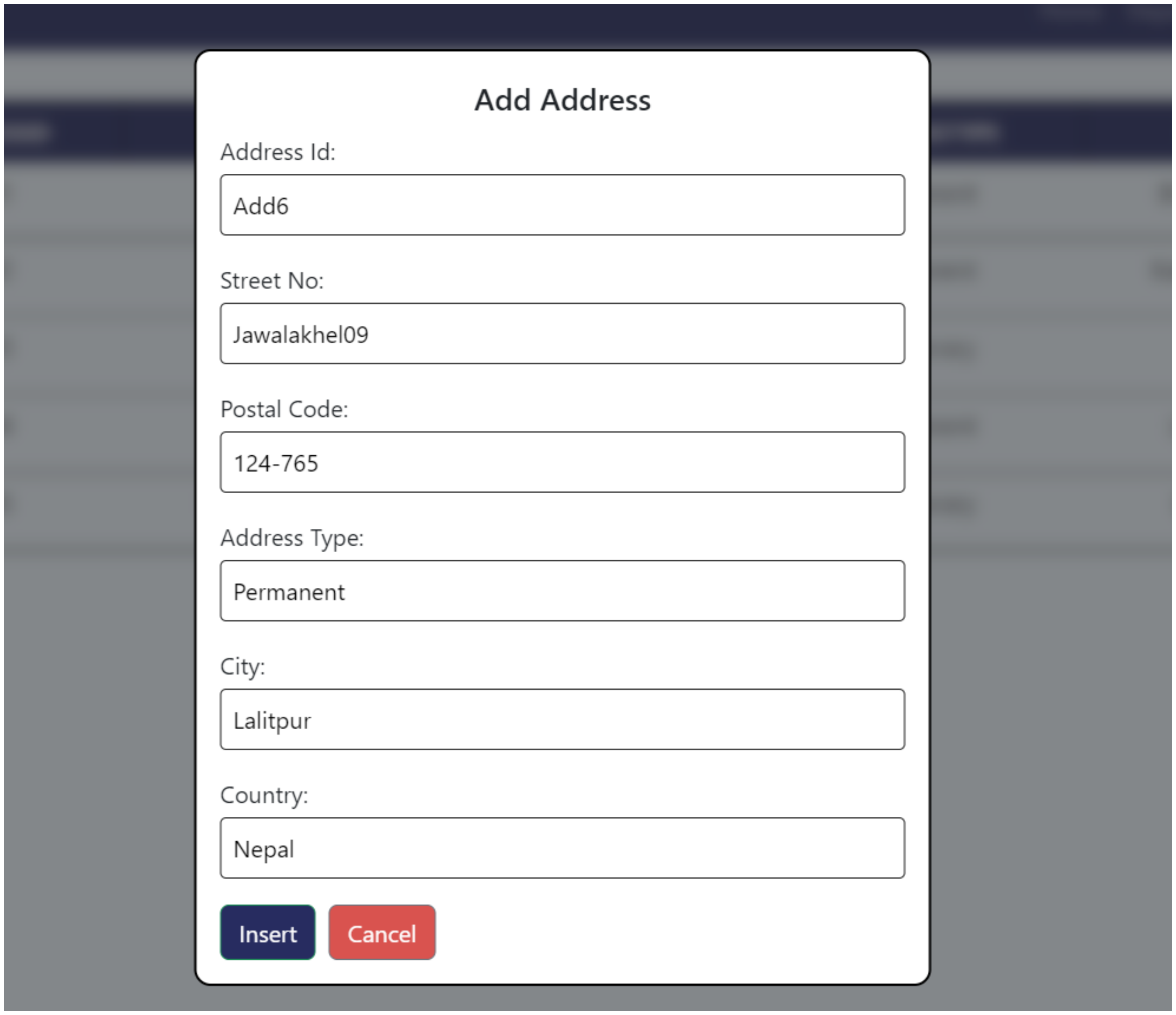
Test No	4
Objective	To add address to Address table.
Action	i) Click on new button. ii) Fill address information. iiI) Click insert button.
Expected Result	New address will be added.
Actual Result	New address was added.
Conclusion	Test successful.

Table 14 To add address to Address table



Action	ADDRESSID	STREETNO	POSTALCODE	ADDRESSTYPE	CITY	COUNTRY
Edit Delete	Add1	Kadaghari 01	443-56	Permanent	Bhaktapur	Nepal
Edit Delete	Add2	Baluwatar 02	123-456	Permanent	Kathmandu	Nepal
Edit Delete	Add3	Manhattan 03	998-345	Temporary	NYC	USA
Edit Delete	Add4	Kusunti 04	432-112	Permanent	Lalitipur	Nepal
Edit Delete	Add5	Park Avenue 05	665-321	Temporary	Boston	USA

Figure 52 Address table before



Add Address

Address Id: Add6

Street No: Jawalakhel09

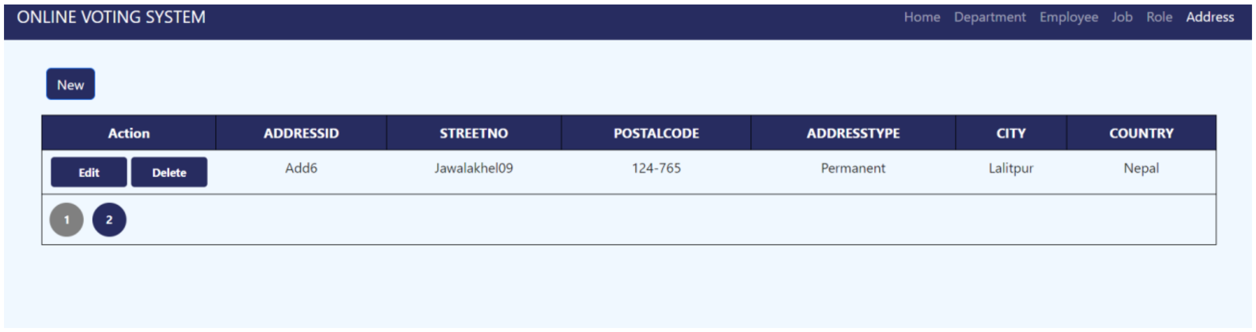
Postal Code: 124-765

Address Type: Permanent

City: Lalitpur

Country: Nepal

Figure 53 Adding new address



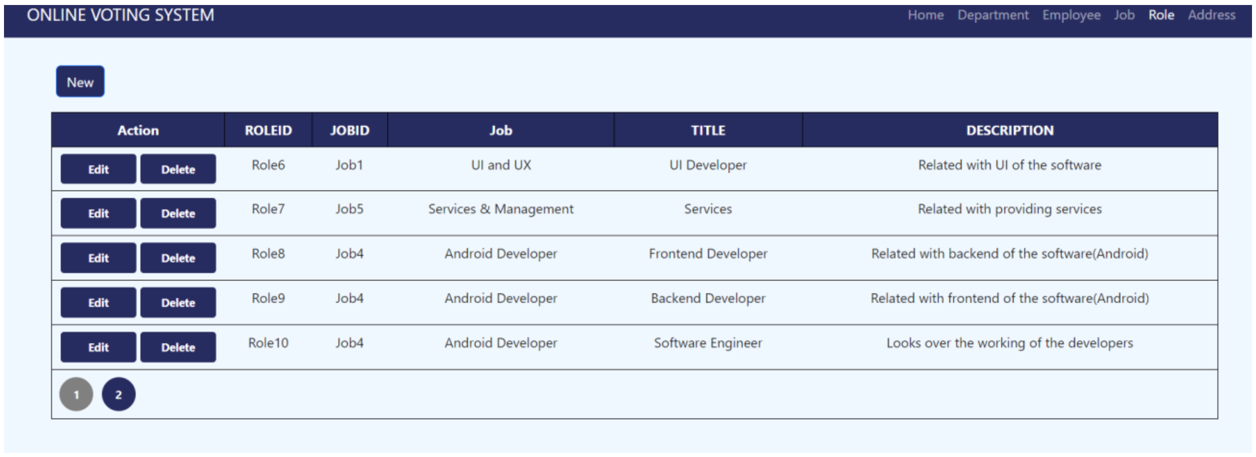
Action	ADDRESSID	STREETNO	POSTALCODE	ADDRESSTYPE	CITY	COUNTRY
Edit Delete	Add6	Jawalakhel09	124-765	Permanent	Lalitpur	Nepal

Figure 54 Address added to Address table (Test successful)

13.5. Test Case: To delete UI Developer from Role table

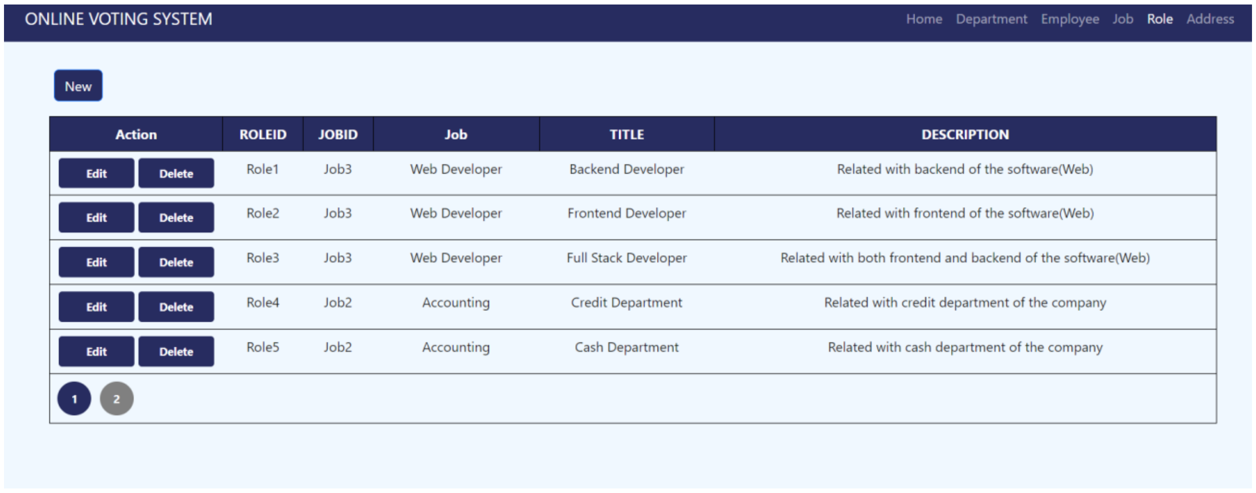
Test No	5
Objective	To delete UI Developer from Role table.
Action	Click on delete button.
Expected Result	UI Developer from Role table will be deleted.
Actual Result	UI Developer from Role table was deleted.
Conclusion	Test successful.

Table 15 To delete UI Developer from Role table



Action	ROLEID	JOBID	Job	TITLE	DESCRIPTION
Edit Delete	Role6	Job1	UI and UX	UI Developer	Related with UI of the software
Edit Delete	Role7	Job5	Services & Management	Services	Related with providing services
Edit Delete	Role8	Job4	Android Developer	Frontend Developer	Related with backend of the software(Android)
Edit Delete	Role9	Job4	Android Developer	Backend Developer	Related with frontend of the software(Android)
Edit Delete	Role10	Job4	Android Developer	Software Engineer	Looks over the working of the developers

Figure 55 Role table before deleting UI Developer


ONLINE VOTING SYSTEM

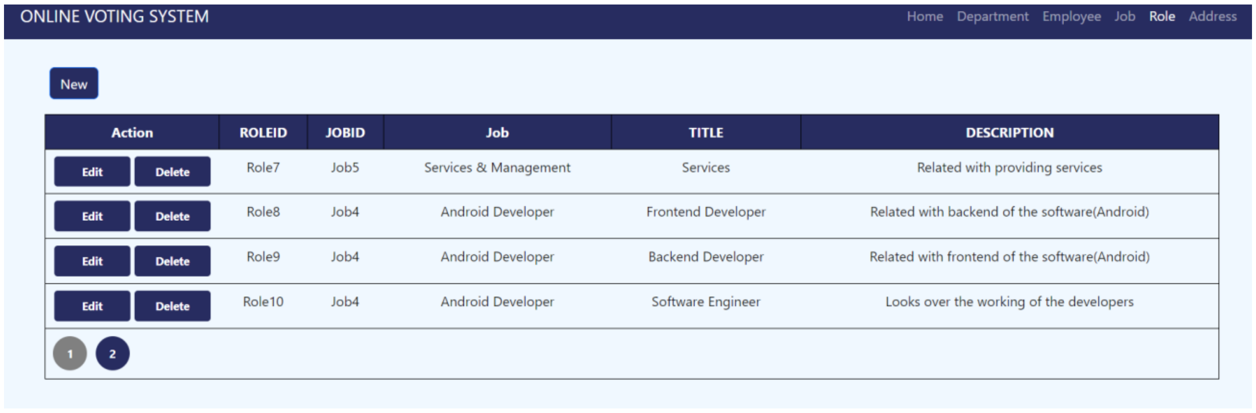
Home Department Employee Job Role Address

New

Action	ROLEID	JOBID	Job	TITLE	DESCRIPTION
Edit Delete	Role1	Job3	Web Developer	Backend Developer	Related with backend of the software(Web)
Edit Delete	Role2	Job3	Web Developer	Frontend Developer	Related with frontend of the software(Web)
Edit Delete	Role3	Job3	Web Developer	Full Stack Developer	Related with both frontend and backend of the software(Web)
Edit Delete	Role4	Job2	Accounting	Credit Department	Related with credit department of the company
Edit Delete	Role5	Job2	Accounting	Cash Department	Related with cash department of the company

1 2

Figure 56 Role table after deleting UI Developer (Test Successful 1)


ONLINE VOTING SYSTEM

Home Department Employee Job Role Address

New

Action	ROLEID	JOBID	Job	TITLE	DESCRIPTION
Edit Delete	Role7	Job5	Services & Management	Services	Related with providing services
Edit Delete	Role8	Job4	Android Developer	Frontend Developer	Related with backend of the software(Android)
Edit Delete	Role9	Job4	Android Developer	Backend Developer	Related with frontend of the software(Android)
Edit Delete	Role10	Job4	Android Developer	Software Engineer	Looks over the working of the developers

1 2

Figure 57 Role table after delete UI Developer (Test Successful 2)

14. Further discussion:

A web application using C# and the ASP.NET framework was developed a thorough review of the work was obtained. It is wonderful to know that the curriculum was challenging but worthwhile and that a lot of knowledge about IDEs, database management systems, and database design tools was achieved.

A major job of understanding the significance of normalization, which is a key idea in database architecture. Building effective, scalable, and maintained high-quality databases requires avoiding data duplication and guaranteeing data integrity. Using normalization techniques can help eliminate redundant data and guarantee that the data is consistent throughout the database, which can speed up queries, improve the quality of the data, and make maintenance simpler.

Tools and techniques learned during the coursework:

14.1. Oracle SQL Developer:

Oracle SQL Developer increases productivity and makes database development work easier. You can view database objects, run pre-made or custom reports, update and debug PL/SQL code, and issue SQL commands using SQL Developer (Kumar, 2018).

14.2. Oracle Data Modeler:

Oracle Data Modeler is a tool provided by the Oracle Corporation for creating, defining, and deploying database schemas. With a graphical user interface, database designers can create and modify data models, which are visual representations of a database's structure (GUI).

14.3. Visual Studio:

Visual Studio, often known as Microsoft Visual Studio and VS, is an integrated development environment for Microsoft Windows. It is a tool for developing websites, web services, online apps, and computer programs. It has a code editor, debugger, GUI design tool, and database schema builder, and it supports the majority of the most widely used revision control systems. There are two versions available: an expensive commercial version and a free "Community" edition (Hope, 2019).

15. Conclusion:

A database management system (DBMS) is a piece of software used to manage, organize, and alter data in a database. A database, which is a collection of prepared data, is a useful tool for businesses and organizations to store and manage data.

A crucial strategy that may be used to ensure that the database is efficient and free of data abnormalities is normalization. By dividing a large database into smaller tables and building links between them, normalization can reduce data redundancy and improve data consistency.

The web-based database application for this project allows users to perform CRUD operations on data that is stored in databases. The database schema has been fully designed and optimized. Users of the application have access to a user-friendly interface that allows them to see data in various formats and add new entries to tables containing information about employees, departments, jobs, positions, and locations. Giving consumers a quick and simple way to save and analyse data is the project's main goal in order to improve productivity and decision-making.

Problems always occur when it comes to building a project. In similar terms, there were some difficulties while doing this project too. The notion and the concepts were new and were quite perplexing in some cases. But these difficulties were overcome by detailed research work and implementation of the research done. These difficulties were helped to be overcome by the teachers and their guidance.

Bibliography

Hope, C., 2019. *Computer Hope.* [Online]
Available at: https://www.computerhope.com/jargon/v/visual-studio.htm
[Accessed 14 March 2023].

Kumar, A., 2018. *udemy.* [Online]
Available at: https://www.udemy.com/course/oracle-sql-developer/
[Accessed 14 March 2023].

R, V. M., 2023. *edureka.* [Online]
Available at: https://www.edureka.co/blog/what-is-a-database/
[Accessed 13 March 2023].