LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

# CC6001NI Advanced Database System Development

## 40% Individual Coursework

**Student Name: 20048842**

**London Met ID:**

**College ID:** NP01CP4210068

**Assignment Due Date: 15th March**

**Assignment Submission Date: 15th March**

**Word Count: 2900**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded*

TABLE OF CONTENTS

## List of Figures

## List of Tables

## Introduction

Database systems are essential in the current digital era for successfully managing and organizing data for organizations of all sizes. Businesses are implementing database systems to store, handle, and retrieve data more effectively in response to the rising demand for automated systems. The assignment for this course is to develop and implement a web-based database application for a company that wants to set up an online voting system to select the employee of the month.

The primary objective of this assignment is to evaluate the students' ability to think critically and solve practical problems while designing and developing a database system. Students must use Oracle SQL Developer Data Modeler and ASP.NET with C# to analyze, design, and execute a database system for a specific business case study as part of their coursework.

The project expects us to create a functioning, properly documented system that satisfies the required deliverables in order to complete this course's requirements. The creation of a set of fully normalized system tables, creating an entity relationship diagram with Oracle SQL Developer Data Modeler, creating a list of attributes for each entity in the Data Dictionary, creating the database tables with Oracle SQL Developer Data Modeler, and implementing a web-based database application with ASP.NET and C# are a few of these.

## Textual Analysis

### Employee and Department Relationship:

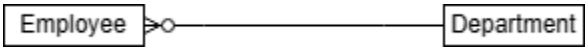



*Figure 1: Employee and Department Relationship:*

Description: Employee and Department have a one-to-many relationship, where one department can have many employees, but each employee can only belong to one department.

### Employee and Voting Record Relationship:

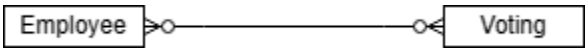



*Figure 2: Employee and Voting Record Relationship:*

Description: Employee and Vote have a many-to-many relationship, where one employee can cast multiple votes, and one candidate can receive multiple votes.

### Employee and Role Record Relationship:

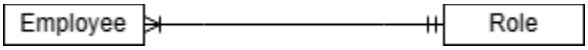



*Figure 3: Employee and Role Record Relationship:*

Employee and Role have a one-to-many relationship, where one employee can have only one role in their assigned department, but each role can belong to multiple employees.

## Case Diagram ERD

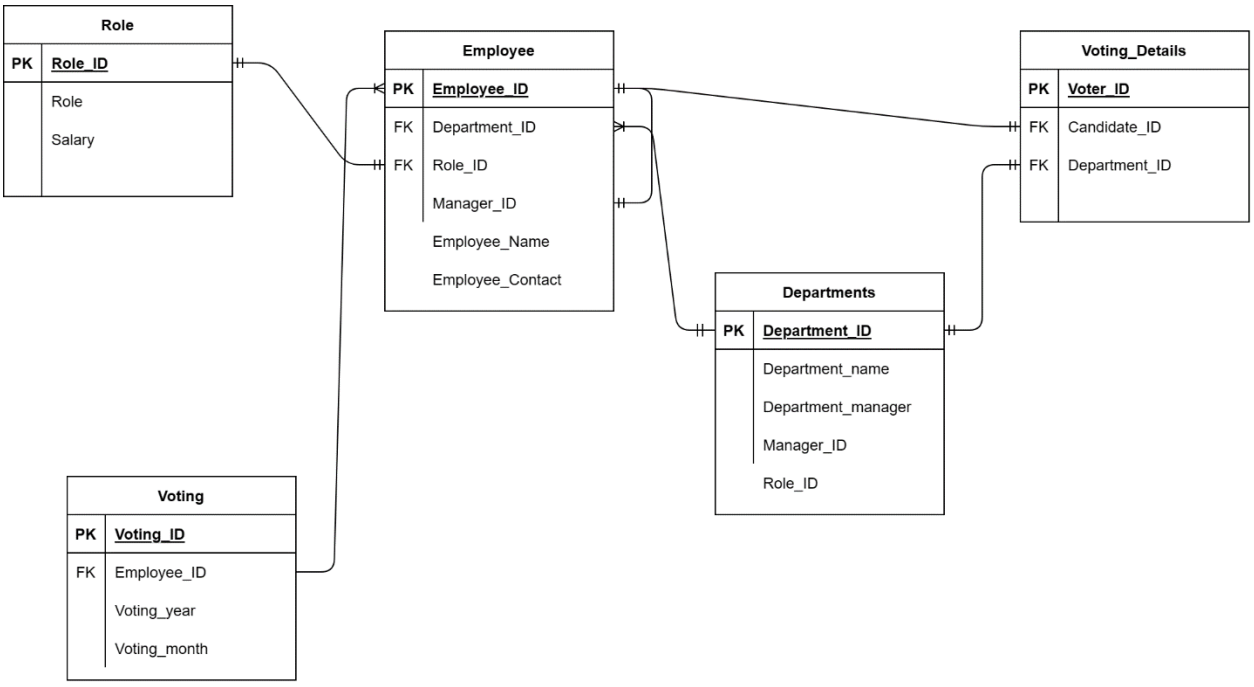



*Figure 4: Case Diagram ERD*

## Integration and Assumption

The assumptions made and the bridge entities constructed are:

• Job: The many to many connections between employees and departments were the core idea used in the development of this bridge entity. By giving employees and departments roles that are shared by both of the entities, this entity was established to strengthen the relationship.

• Role: The relationship between departments and jobs was the main assumption used to create this bridge entity. Many Roles could exist in many departments, thus role entity was created.

• Voting details: This bridge entity was created to remove the relationship between employees and voting tables, as many employees could vote for many other candidates.

## Normalization

UNF

Employee (S.N., EmployeeName, DateofBirth, Contact, {Email}, {Address}, Department)

Repeating groups ==> email and address

1NF

---Removing repeating group---

Employee-1: (S.N, EmployeeName, DateofBirth, Contact, Department, Dept_id)

Email-1: S.N*, Email

Address-1: S.N (FK), Address

---Creating new attribute Dept_id in Employee table---

2NF

---Removing partial dependency---

Since, Employee table has no composite primary key, there is no partial dependency. Hence, the table is already in 2NF.

Since, Email and Address table does not contain any non-key attributes, the table is already in 2NF.

Employee-1: (S.N, EmployeeName, DateofBirth, Contact, Department, Dept_id)

Email-1: S.N*, Email

Address-1: S.N (FK), Address

3NF

---Removing transitive dependency---

For Employee table

S.N. -> Employee_name, DOB, Contact

sn --> Dept_id -> Dept_name

Since, transitive dependency is found. New table department will be formed as such.

Employee-3 (S.N, Employee_name, DateofBirth, Contact, Dept_id*)

Department-3 (Dept_id, Dept_name)

For table Email:

Since, Email table does not contain any non-key attributes, the table is already in 3NF.

For table Address:

Since, Address table does not contain any non-key attributes, the table is already in 3NF.

Final Tables:

Employee (S.N, Employee_name, DateofBirth, Contact, Dept_id*)

Department (Dept_id, Dept_name)

Email (S.N*, Email)

Address (S.N*, address

Normalization of Figure 2

UNF

Voter (voter_id, voter_name, {voting_year, {voting_month, candidate_id,

candidate_name, candidate_department}})

Repeating groups = voting_year

Repeating group under voting_year = voting_month, candidate_id, candidate_name, candidate_department

1NF

---Removing repeating group---

Voter-1 (voter_id, voter_name)

Voter-Year-1 (voter_id*, voting_year)

Voting-Details-1 (voting_month, candidate_id, candidate_name,

department_id, candidate_department, voter_id*, voting_year*)

New attribute department_id was added

2NF

For Voters table:

Since, Voter table has no composite primary key, there is no partial dependency. Hence, the table is already in 2NF.

For Voter-year table:

Since, Voter_year table does not contain any non-key attributes, the table is already in 2NF.

For table Voting_detail:

voter_id -> X voting_year -> X voting_month -> X

voter_id, voting_year -> X voting_year, voting_month -> X voter_id, voting_month -> X

voter_id, voting_year, voting_month -> candidate_id, candidate_name, department_id, candidate_department

Since, no partial dependency was found, the table Voting_detail is in 2NF.

Voter-1 (voter_id, voter_name)

Voter-Year-1 (voter_id*, voting_year)

Voting-Details-1 (voting_month, candidate_id, candidate_name,

department_id, candidate_department, voter_id*, voting_year*)

3NF:

Removing transitive dependency For table Voter:

Since, there is only one non-key attribute, the table is already in 3NF. For table Voter_year:

Since, Voter_year table does not contain any non-key attributes, the table is already in 3NF.

candidate_id -> candidate_name department_id -> candidate_department

voter_id*, voting_year*, voting_month -> candidate_id --> can_department_id, can_name, can_department

canid --> depid --> depname

Since, transitive dependencies were found. Creating new tables:

Voter-1 (voter_id, voter_name)

Voter-Year-1 (voter_id*, voting_year)

Voting-Details-3 (voting_month, candidate_id*, voter_id*, voting_year*)

Candidate-3 (candidate_id, candidate_name, department_id*)

Department-3 (department_id, candidate_department

# Final ERD

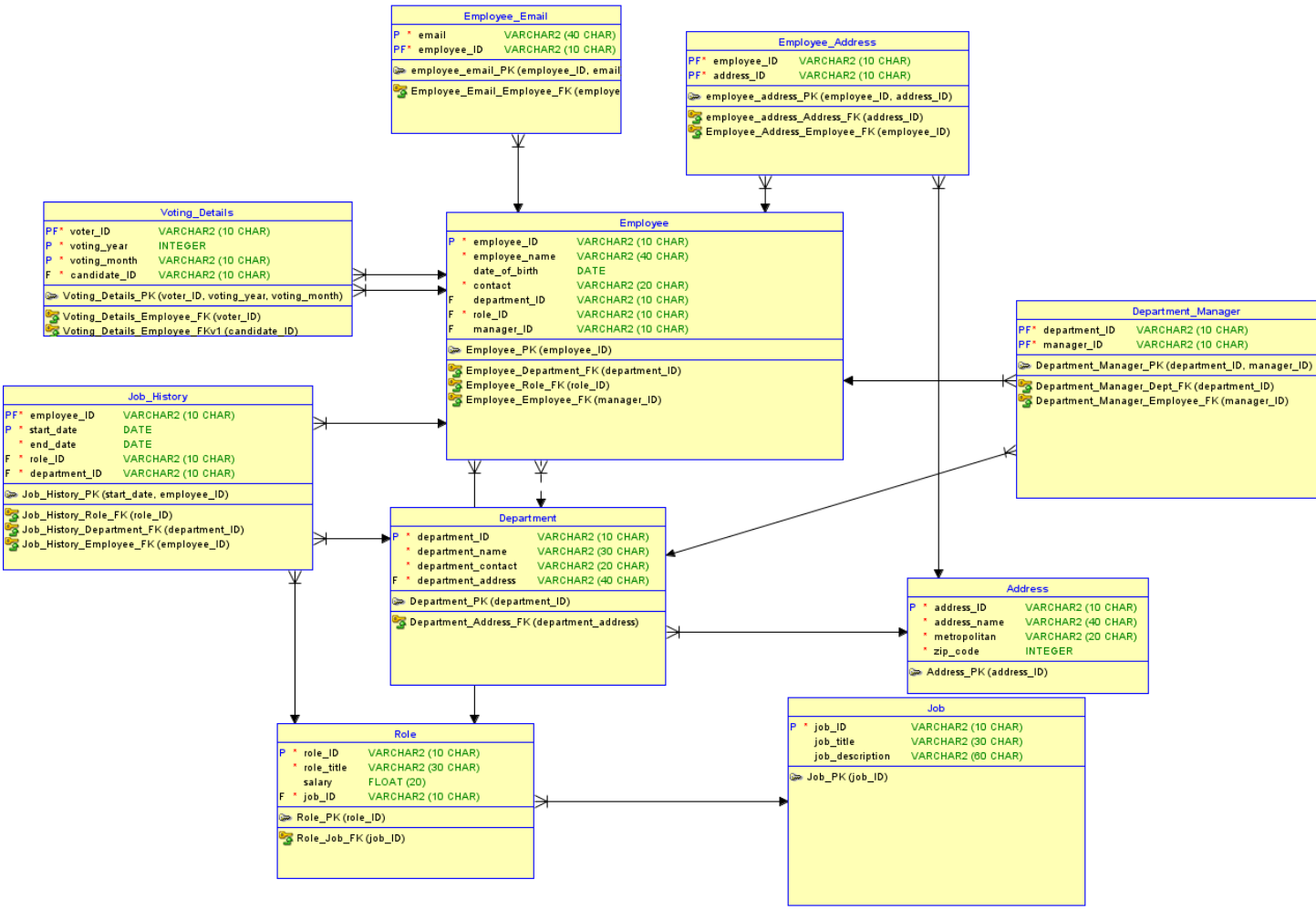



*Figure 5: Final ERD*

## Data Dictionary

### Address

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| address_ID | Varchar | 10 | Primary Key | | | To uniquely identify each address | A001 |
| address_name | Varchar | 40 | Not Null | | | To store the address' name | 123 Main St |
| metropolitan | Varchar | 20 | Not Null | | | To store metropolitan | New York City |
| zip_code | Integer | | | | | To store the address' zip code | 10001 |

*Table 1: Address*

### Job

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| job_ID | Varchar | 10 | Primary Key | | | To uniquely identify Each job | J001 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| job_title | Varchar | 30 | | | | To store the job's name | Software Engineer |
| job_description | Varchar | 60 | Default("Works in the company") | | | To store job's desciption | Develop and maintain software solutions |

*Table 2: Job*

**Role**

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| role_ID | Varchar | 10 | Primary Key | | | To uniquely identify each role | R001 |
| role_title | Varchar | 30 | Not Null | | | To store the role's name | Software Engineer |
| salary | Float | 20 | Check(salary >= 0) | | | To store role's salary | 100000 |
| job_ID | Varchar | 10 | Foreign Key, Not Null | Job | job_ID | To store role's job ID | J001 |

*Table 3: Role*

**Department_Manager**

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| department_ID | Varchar | 10 | Primary Key, Foreign Key | Department | department_ID | To store id of the manager's department id | D001 |
| manager_ID | Varchar | 30 | Primary Key, Foreign Key | Employee | manager_ID | To store id of the manager id of which department | E001 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | the manager belongs to | |

*Table 4: Department_Manager*

**Employee**

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| employee_ID | Varchar | 10 | Primary Key | | | To uniquely identify each employee | E001 |
| employee_name | Varchar | 40 | Not Null | | | To store the employee's name | John Doe |
| date_of_birth | Date | | | | | To store employee's date of birth | Jane Smith |
| contact | Varchar | 20 | Unique | | | To store employee's contact | 555-1234 |
| department_id | Varchar | 10 | Foreign Key, Not Null | Department | department_ID | To store employee 's departemnet id | D001 |
| role_ID | Varchar | 10 | Foreign Key, Not Null | Role | role_ID | To store employee's role id | R001 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| manager_ID | Varchar | 10 | Foreign Key | Manager | manager_ID | To store employee's manager id | E0002 |

*Table 5: Employee*

**Employee_Address**

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| employee_ID | Varchar | 10 | Foreign Key, Not Null | Employee | employee_ID | To store id of the employee id of who the email belongs to | E001 |
| address_ID | Varchar | 10 | Foreign Key, Not Null | Address | address_ID | To store id of the employee's address id | A001 |

*Table 6: Employee_Address*

**Employee_Email**

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| Email | Varchar | 10 | Primary Key | | | To store id of the manager | john.doe@example.com |
| employee_ID | Varchar | 10 | Primary Key, | Employee | employee_ID | To store id of the | E001 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Foreign Key | | | employee id of who the email belongs to | |

*Table 7: Employee_Email*

**Voting_Details**

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| voter_ID | Varchar | 10 | Primary Key, Foreign Key | Employee | Employee_ID | To uniquely identify each voter | E001 |
| voting_year | Integer | | Primary Key | | | To store the voting's year | 2023 |
| voting_month | Varchar | 10 | Primary Key | | | To store voting's month | January |
| candidate_ID | Varchar | 10 | Unique | | | To store candidate's id | E002 |

*Table 8: Voting_Details*

**Job_History**

| Column Name | Data Type | Size | Constraint | Reference Table | Reference Column | Description | Example Data |
|---|---|---|---|---|---|---|---|
| employee_ID | Varchar | 10 | Primary Key, Foreign Key | Employee | Employee_ID | To uniquely identify each employee | E001 |
| start_date | Integer | | Not Null | | | To store the employee's start date | 01-JAN-22 |
| end_date | Varchar | 10 | Not Null | | | To store employee's end date | 31-DEC-22 |
| role_ID | Varchar | 10 | Unique | | | To store employee's role id | R001 |
| department_ID | | | Not Null | | | To store employee's department id | D001 |

*Table 9: Job History*

## Script

**Job**

```
CREATE TABLE job (
    job_id          VARCHAR2(10 CHAR) NOT NULL,
    job_title       VARCHAR2(30 CHAR),
    job_description VARCHAR2(60 CHAR)
);

ALTER TABLE job ADD CONSTRAINT job_pk PRIMARY KEY ( job_id );
ALTER TABLE job MODIFY job_description DEFAULT 'Works in the company';
```

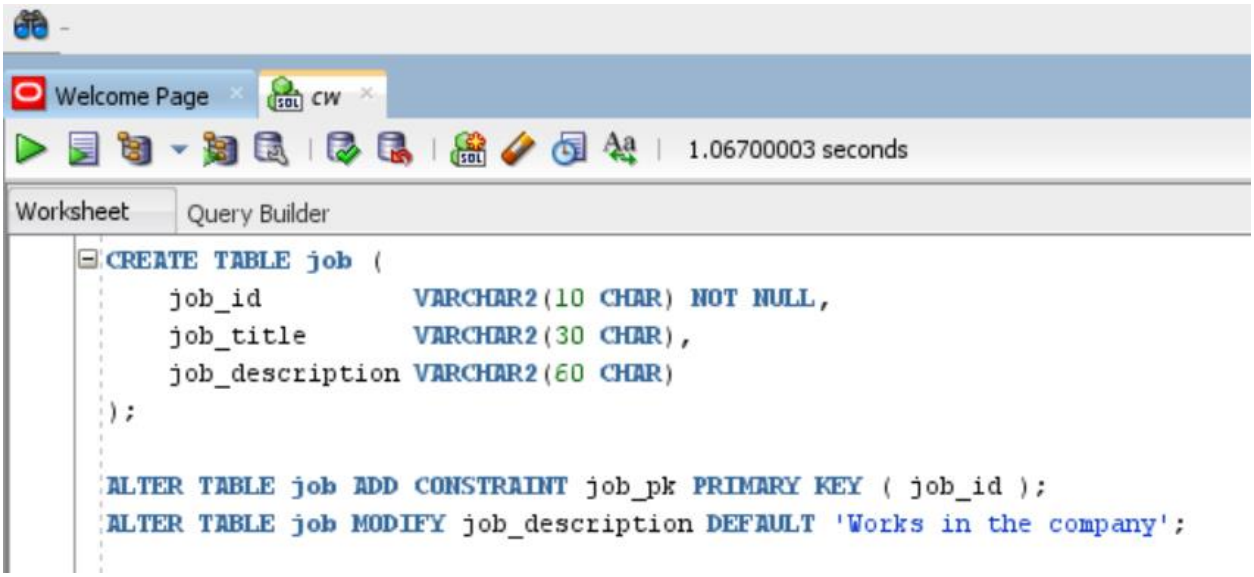



*Figure 6: Creation of "Job" Table*

**Address**

CREATE TABLE address (

address_id VARCHAR2(10 CHAR) NOT NULL,

address_name VARCHAR2(40 CHAR) NOT NULL,

metropolitan VARCHAR2(20 CHAR) NOT NULL,

zip_code INTEGER NOT NULL

);

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

```
CREATE TABLE address (
    address_id   VARCHAR2(10 CHAR) NOT NULL,
    address_name VARCHAR2(40 CHAR) NOT NULL,
    metropolitan VARCHAR2(20 CHAR) NOT NULL,
    zip_code     INTEGER NOT NULL
);

ALTER TABLE address ADD CONSTRAINT address_pk PRIMARY KEY ( address_id );
```

*Figure 7: Creation of "Address" Table*

**Department**

CREATE TABLE department (

department_id VARCHAR2(10 CHAR) NOT NULL,

department_name VARCHAR2(30 CHAR) NOT NULL,

department_contact VARCHAR2(20 CHAR) NOT NULL,

department_address VARCHAR2(40 CHAR) NOT NULL

);

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

ALTER TABLE department

ADD CONSTRAINT department_address_fk FOREIGN KEY ( department_address )

REFERENCES address ( address_id );

```
CREATE TABLE department (
    department_id      VARCHAR2(10 CHAR) NOT NULL,
    department_name    VARCHAR2(30 CHAR) NOT NULL,
    department_contact VARCHAR2(20 CHAR) NOT NULL,
    department_address VARCHAR2(40 CHAR) NOT NULL
);

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

ALTER TABLE department
    ADD CONSTRAINT department_address_fk FOREIGN KEY ( department_address )
        REFERENCES address ( address_id );
```

*Figure 8: Creation of "Address" Table*

**Role**

CREATE TABLE role (

role_id VARCHAR2(10 CHAR) NOT NULL,

role_title VARCHAR2(30 CHAR) NOT NULL,

salary FLOAT(20),

job_id VARCHAR2(10 CHAR) NOT NULL

);

ALTER TABLE role ADD CONSTRAINT role_pk PRIMARY KEY ( role_id );

ALTER TABLE role

ADD CONSTRAINT role_job_fk FOREIGN KEY ( job_id )

REFERENCES job ( job_id );

ALTER TABLE salary

ADD CHECK (salary>=0);

```
CREATE TABLE role (
    role_id    VARCHAR2(10 CHAR) NOT NULL,
    role_title VARCHAR2(30 CHAR) NOT NULL,
    salary     FLOAT(20),
    job_id     VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE role ADD CONSTRAINT role_pk PRIMARY KEY ( role_id );

ALTER TABLE role
    ADD CONSTRAINT role_job_fk FOREIGN KEY ( job_id )
        REFERENCES job ( job_id );

ALTER TABLE role ADD CHECK (salary>=0);
```

*Figure 9: Creation of "Role" table*

**Employee**

CREATE TABLE employee (

```
    employee_id   VARCHAR2(10 CHAR) NOT NULL,
    employee_name VARCHAR2(40 CHAR) NOT NULL,
    date_of_birth DATE,
    contact       VARCHAR2(20 CHAR) NOT NULL,
    department_id VARCHAR2(10 CHAR) NOT NULL,
    role_id       VARCHAR2(10 CHAR) NOT NULL,
    manager_id    VARCHAR2(10 CHAR) NOT NULL
);

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

ALTER TABLE employee
    ADD CONSTRAINT employee_department_fk FOREIGN KEY ( department_id )
        REFERENCES department ( department_id );

ALTER TABLE employee
    ADD CONSTRAINT employee_employee_fk FOREIGN KEY ( manager_id )
        REFERENCES employee ( employee_id );

ALTER TABLE employee
    ADD CONSTRAINT employee_role_fk FOREIGN KEY ( role_id )
        REFERENCES role ( role_id );
```

ALTER TABLE employee

ADD UNIQUE (contact);

```
CREATE TABLE employee (
    employee_id   VARCHAR2(10 CHAR) NOT NULL,
    employee_name VARCHAR2(40 CHAR) NOT NULL,
    date_of_birth DATE,
    contact       VARCHAR2(20 CHAR) NOT NULL,
    department_id VARCHAR2(10 CHAR) NOT NULL,
    role_id       VARCHAR2(10 CHAR) NOT NULL,
    manager_id    VARCHAR2(10 CHAR) NOT NULL
);

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

ALTER TABLE employee
    ADD CONSTRAINT employee_department_fk FOREIGN KEY ( department_id )
        REFERENCES department ( department_id );

ALTER TABLE employee
    ADD CONSTRAINT employee_employee_fk FOREIGN KEY ( manager_id )
        REFERENCES employee ( employee_id );
```

*Figure 10: Creation of "Employee" table 1*

```
ALTER TABLE employee
    ADD CONSTRAINT employee_role_fk FOREIGN KEY ( role_id )
        REFERENCES role ( role_id );

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

*Figure 11: Creation of "Employee" table 2*

**Job_History**

CREATE TABLE job_history (

```
    employee_id   VARCHAR2(10 CHAR) NOT NULL,
    start_date    DATE NOT NULL,
    end_date      DATE NOT NULL,
    role_id       VARCHAR2(10 CHAR) NOT NULL,
    department_id VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE job_history ADD CONSTRAINT job_history_pk PRIMARY KEY ( start_date, employee_id );

ALTER TABLE job_history
    ADD CONSTRAINT job_history_department_fk FOREIGN KEY ( department_id )
        REFERENCES department ( department_id );

ALTER TABLE job_history
    ADD CONSTRAINT job_history_employee_fk FOREIGN KEY ( employee_id )
        REFERENCES employee ( employee_id );

ALTER TABLE job_history
    ADD CONSTRAINT job_history_role_fk FOREIGN KEY ( role_id )
        REFERENCES role ( role_id );
```

```
CREATE TABLE job_history (
    employee_id   VARCHAR2(10 CHAR) NOT NULL,
    start_date    DATE NOT NULL,
    end_date      DATE NOT NULL,
    role_id       VARCHAR2(10 CHAR) NOT NULL,
    department_id VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE job_history ADD CONSTRAINT job_history_pk PRIMARY KEY ( start_date,
                                                                    employee_id );

ALTER TABLE job_history
    ADD CONSTRAINT job_history_department_fk FOREIGN KEY ( department_id )
        REFERENCES department ( department_id );

ALTER TABLE job_history
    ADD CONSTRAINT job_history_employee_fk FOREIGN KEY ( employee_id )
        REFERENCES employee ( employee_id );

ALTER TABLE job_history
    ADD CONSTRAINT job_history_role_fk FOREIGN KEY ( role_id )
        REFERENCES role ( role_id );
```

*Figure 12: Creation of "Job_History" table*

**Department_Manager**

CREATE TABLE department_manager (

department_id VARCHAR2(10 CHAR) NOT NULL,

manager_id VARCHAR2(10 CHAR) NOT NULL

);

ALTER TABLE department_manager ADD CONSTRAINT department_manager_pk PRIMARY KEY ( department_id,

manager_id );

ALTER TABLE department_manager

ADD CONSTRAINT department_manager_dept_fk FOREIGN KEY ( department_id )

REFERENCES department ( department_id );

ALTER TABLE department_manager

ADD CONSTRAINT department_manager_employee_fk FOREIGN KEY ( manager_id )

REFERENCES employee ( employee_id );

```
CREATE TABLE department_manager (
    department_id VARCHAR2(10 CHAR) NOT NULL,
    manager_id    VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE department_manager ADD CONSTRAINT department_manager_pk PRIMARY KEY ( department_id,
                                                                                  manager_id );

ALTER TABLE department_manager
    ADD CONSTRAINT department_manager_dept_fk FOREIGN KEY ( department_id )
        REFERENCES department ( department_id );

ALTER TABLE department_manager
    ADD CONSTRAINT department_manager_employee_fk FOREIGN KEY ( manager_id )
        REFERENCES employee ( employee_id );
```

*Figure 13: Creation of "Department_Manager" table*

**Employee_Address**

CREATE TABLE employee_address (

employee_id VARCHAR2(10 CHAR) NOT NULL,

address_id  VARCHAR2(10 CHAR) NOT NULL

);

ALTER TABLE employee_address ADD CONSTRAINT employee_address_pk PRIMARY KEY ( employee_id,

address_id );

ALTER TABLE employee_address

ADD CONSTRAINT employee_address_address_fk FOREIGN KEY ( address_id )

REFERENCES address ( address_id );

ALTER TABLE employee_address

ADD CONSTRAINT employee_address_employee_fk FOREIGN KEY ( employee_id )

REFERENCES employee ( employee_id );

```
CREATE TABLE employee_address (
    employee_id VARCHAR2(10 CHAR) NOT NULL,
    address_id  VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE employee_address ADD CONSTRAINT employee_address_pk PRIMARY KEY ( employee_id,
                                                                              address_id );

ALTER TABLE employee_address
    ADD CONSTRAINT employee_address_address_fk FOREIGN KEY ( address_id )
        REFERENCES address ( address_id );

ALTER TABLE employee_address
    ADD CONSTRAINT employee_address_employee_fk FOREIGN KEY ( employee_id )
        REFERENCES employee ( employee_id );
```

*Figure 14: Creation of "Employee_Address" table*

**Employee_Email**

CREATE TABLE employee_email (

email VARCHAR2(40 CHAR) NOT NULL,

employee_id VARCHAR2(10 CHAR) NOT NULL

);

ALTER TABLE employee_email ADD CONSTRAINT employee_email_pk PRIMARY KEY ( employee_id,

email );

ALTER TABLE employee_email

ADD CONSTRAINT employee_email_employee_fk FOREIGN KEY ( employee_id )

REFERENCES employee ( employee_id );

```
CREATE TABLE employee_email (
    email       VARCHAR2(40 CHAR) NOT NULL,
    employee_id VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE employee_email ADD CONSTRAINT employee_email_pk PRIMARY KEY ( employee_id,
                                                                          email );

ALTER TABLE employee_email
    ADD CONSTRAINT employee_email_employee_fk FOREIGN KEY ( employee_id )
        REFERENCES employee ( employee_id );
```

*Figure 15: Creation of "Voting_Details" table*

**Voting_Details**

CREATE TABLE voting_details (

```
    voter_id     VARCHAR2(10 CHAR) NOT NULL,
    voting_year  INTEGER NOT NULL,
    voting_month VARCHAR2(10 CHAR) NOT NULL,
    candidate_id VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE voting_details
    ADD CONSTRAINT voting_details_pk PRIMARY KEY ( voter_id,
                                                   voting_year,
                                                   voting_month );

ALTER TABLE voting_details
    ADD CONSTRAINT voting_details_employee_fk FOREIGN KEY ( voter_id )
        REFERENCES employee ( employee_id );

ALTER TABLE voting_details
    ADD CONSTRAINT voting_details_employee_fkv1 FOREIGN KEY ( candidate_id )
        REFERENCES employee ( employee_id );
```

```
CREATE TABLE voting_details (
    voter_id     VARCHAR2(10 CHAR) NOT NULL,
    voting_year  INTEGER NOT NULL,
    voting_month VARCHAR2(10 CHAR) NOT NULL,
    candidate_id VARCHAR2(10 CHAR) NOT NULL
);

ALTER TABLE voting_details
    ADD CONSTRAINT voting_details_pk PRIMARY KEY ( voter_id,
                                                   voting_year,
                                                   voting_month );

ALTER TABLE voting_details
    ADD CONSTRAINT voting_details_employee_fk FOREIGN KEY ( voter_id )
        REFERENCES employee ( employee_id );

ALTER TABLE voting_details
    ADD CONSTRAINT voting_details_employee_fkv1 FOREIGN KEY ( candidate_id )
        REFERENCES employee ( employee_id );
```

*Figure 16: Creation of "Voting_Details" table*

## Insert Statement

### Job

```
INSERT INTO job (job_id, job_title, job_description)
VALUES ('J001', 'Software Engineer', 'Develop and maintain software solutions');

INSERT INTO job (job_id, job_title, job_description)
VALUES ('J002', 'Marketing Specialist', 'Create and execute marketing campaigns');

INSERT INTO job (job_id, job_title, job_description)
VALUES ('J003', 'Accountant', 'Manage financial records and reporting');

INSERT INTO job (job_id, job_title, job_description)
VALUES ('J004', 'Human Resources Coordinator', 'Assist with hiring and employee relations');

INSERT INTO job (job_id, job_title, job_description)
VALUES ('J005', 'CSR', 'Assist customers with inquiries and complaints');
```

*Figure 17: Insert data into "Job" table*

### Address

```
INSERT INTO address (address_id, address_name, metropolitan, zip_code)
VALUES ('A001', '123 Main St', 'New York City', 10001);

INSERT INTO address (address_id, address_name, metropolitan, zip_code)
VALUES ('A002', '456 Elm St', 'San Francisco', 94109);

INSERT INTO address (address_id, address_name, metropolitan, zip_code)
VALUES ('A003', '789 Maple Ave', 'Chicago', 60611);

INSERT INTO address (address_id, address_name, metropolitan, zip_code)
VALUES ('A004', '101 Oak St', 'Los Angeles', 90025);

INSERT INTO address (address_id, address_name, metropolitan, zip_code)
VALUES ('A005', '222 Pine St', 'Seattle', 98101);
```

*Figure 18: Insert data into "Job" table*

### Department

```
INSERT INTO department (department_id, department_name, department_contact, department_address)
VALUES ('D001', 'Engineering', 'John Smith', 'A001');

INSERT INTO department (department_id, department_name, department_contact, department_address)
VALUES ('D002', 'Marketing', 'Jane Doe', 'A002');

INSERT INTO department (department_id, department_name, department_contact, department_address)
VALUES ('D003', 'Finance', 'Bob Johnson', 'A003');

INSERT INTO department (department_id, department_name, department_contact, department_address)
VALUES ('D004', 'Human Resources', 'Sara Lee', 'A004');

INSERT INTO department (department_id, department_name, department_contact, department_address)
VALUES ('D005', 'Customer Service', 'Joe Smith', 'A005');
```

*Figure 19: Insert data into "Department" table*

**Role**

```
INSERT INTO role (role_id, role_title, salary, job_id)
VALUES ('R001', 'Software Engineer', 100000.00, 'J001');

INSERT INTO role (role_id, role_title, salary, job_id)
VALUES ('R002', 'Marketing Manager', 80000.00, 'J002');

INSERT INTO role (role_id, role_title, salary, job_id)
VALUES ('R003', 'Accountant', 75000.00, 'J003');

INSERT INTO role (role_id, role_title, salary, job_id)
VALUES ('R004', 'HR Coordinator', 60000.00, 'J004');

INSERT INTO role (role_id, role_title, salary, job_id)
VALUES ('R005', 'CSR', 40000.00, 'J005');
```

*Figure 20: Insert data into "Role" table*

**Employee**

```
INSERT INTO employee (employee_id, employee_name, date_of_birth, contact, department_id, role_id, manager_id)
VALUES ('E001', 'John Doe', TO_DATE('1990-05-01', 'YYYY-MM-DD'), '555-1234', 'D001', 'R001', 'E001');

INSERT INTO employee (employee_id, employee_name, date_of_birth, contact, department_id, role_id, manager_id)
VALUES ('E002', 'Jane Smith', TO_DATE('1985-10-15', 'YYYY-MM-DD'), '555-5678', 'D001', 'R001', 'E001');

INSERT INTO employee (employee_id, employee_name, date_of_birth, contact, department_id, role_id, manager_id)
VALUES ('E003', 'Bob Johnson', TO_DATE('1992-03-22', 'YYYY-MM-DD'), '555-9876', 'D003', 'R003', 'E001');

INSERT INTO employee (employee_id, employee_name, date_of_birth, contact, department_id, role_id, manager_id)
VALUES ('E004', 'Sara Lee', TO_DATE('1988-12-05', 'YYYY-MM-DD'), '555-1111', 'D004', 'R004', 'E001');

INSERT INTO employee (employee_id, employee_name, date_of_birth, contact, department_id, role_id, manager_id)
VALUES ('E005', 'Joe Smith', TO_DATE('1995-06-30', 'YYYY-MM-DD'), '555-2222', 'D005', 'R005', 'E001');
```

*Figure 21: Insert data into "Employee" table*

### Department_Manager

```
INSERT INTO department_manager (department_id, manager_id)
VALUES ('D001', 'E001');

INSERT INTO department_manager (department_id, manager_id)
VALUES ('D002', 'E002');

INSERT INTO department_manager (department_id, manager_id)
VALUES ('D003', 'E003');

INSERT INTO department_manager (department_id, manager_id)
VALUES ('D004', 'E004');

INSERT INTO department_manager (department_id, manager_id)
VALUES ('D005', 'E005');
```

*Figure 22: Insert data into "Department_Manager" table*

### Job_History

```
INSERT INTO job_history (employee_id, start_date, end_date, role_id, department_id)
VALUES ('E001', TO_DATE('2022-01-01', 'YYYY-MM-DD'), TO_DATE('2022-12-31', 'YYYY-MM-DD'), 'R001', 'D001');

INSERT INTO job_history (employee_id, start_date, end_date, role_id, department_id)
VALUES ('E001', TO_DATE('2021-01-01', 'YYYY-MM-DD'), TO_DATE('2021-12-31', 'YYYY-MM-DD'), 'R001', 'D001');

INSERT INTO job_history (employee_id, start_date, end_date, role_id, department_id)
VALUES ('E002', TO_DATE('2022-02-01', 'YYYY-MM-DD'), TO_DATE('2022-07-31', 'YYYY-MM-DD'), 'R002', 'D002');

INSERT INTO job_history (employee_id, start_date, end_date, role_id, department_id)
VALUES ('E003', TO_DATE('2022-03-01', 'YYYY-MM-DD'), TO_DATE('2022-09-30', 'YYYY-MM-DD'), 'R003', 'D003');

INSERT INTO job_history (employee_id, start_date, end_date, role_id, department_id)
VALUES ('E004', TO_DATE('2022-04-01', 'YYYY-MM-DD'), TO_DATE('2022-05-31', 'YYYY-MM-DD'), 'R004', 'D004');
```

*Figure 23: Insert data into "Job_History"*

```
INSERT INTO employee_address (employee_id, address_id)
VALUES ('E001', 'A001');

INSERT INTO employee_address (employee_id, address_id)
VALUES ('E002', 'A002');

INSERT INTO employee_address (employee_id, address_id)
VALUES ('E003', 'A003');

INSERT INTO employee_address (employee_id, address_id)
VALUES ('E004', 'A004');

INSERT INTO employee_address (employee_id, address_id)
VALUES ('E005', 'A005');
```

*Figure 24: Insert data into "employee_address" table*

**Employee_email**

```
INSERT INTO employee_email (email, employee_id)
VALUES ('john.doe@example.com', 'E001');

INSERT INTO employee_email (email, employee_id)
VALUES ('jane.doe@example.com', 'E002');

INSERT INTO employee_email (email, employee_id)
VALUES ('bob.smith@example.com', 'E003');

INSERT INTO employee_email (email, employee_id)
VALUES ('sara.williams@example.com', 'E004');

INSERT INTO employee_email (email, employee_id)
VALUES ('adam.johnson@example.com', 'E005');
```

*Figure 25: Insert data into "employee_email" table*

### Voting_Details

```
INSERT INTO voting_details (voter_id, voting_year, voting_month, candidate_id)
VALUES ('E001', 2023, 'January', 'E002');

INSERT INTO voting_details (voter_id, voting_year, voting_month, candidate_id)
VALUES ('E002', 2023, 'January', 'E001');

INSERT INTO voting_details (voter_id, voting_year, voting_month, candidate_id)
VALUES ('E003', 2023, 'January', 'E002');

INSERT INTO voting_details (voter_id, voting_year, voting_month, candidate_id)
VALUES ('E004', 2023, 'January', 'E003');

INSERT INTO voting_details (voter_id, voting_year, voting_month, candidate_id)
VALUES ('E005', 2023, 'January', 'E002');
```

*Figure 26: Insert data into "voting_details" table*

## Select Statement

### Job

```
SELECT * FROM job;
```

Script Output

Task completed in 2.84 seconds

| JOB_ID | JOB_TITLE | JOB_DESCRIPTION |
|---|---|---|
| J001 | Software Engineer | Develop and maintain software solutions |
| J002 | Marketing Specialist | Create and execute marketing campaigns |
| J003 | Accountant | Manage financial records and reporting |
| J004 | Human Resources Coordinator | Assist with hiring and employee relations |
| J005 | CSR | Assist customers with inquiries and complaints |
| J006 | Game Development | Develop games |

6 rows selected.

*Figure 27: Rows of Job table*

## Address

```
SELECT * FROM address;
```

Script Output

Task completed in 2.84 seconds

| ADDRESS_ID | ADDRESS_NAME | METROPOLITAN | ZIP_CODE |
|---|---|---|---|
| A001 | 123 Main St | New York City | 10001 |
| A002 | 456 Elm St | San Francisco | 94109 |
| A003 | 789 Maple Ave | Chicago | 60611 |
| A004 | 101 Oak St | Los Angeles | 90025 |
| A005 | 222 Pine St | Seattle | 98101 |

*Figure 28: Rows of Address table*

```
SELECT * FROM department;
```

Script Output

Task completed in 2.84 seconds

| DEPARTMENT | DEPARTMENT_NAME | DEPARTMENT_CONTACT | DEPARTMENT_ADDRESS |
|---|---|---|---|
| D001 | Engineering | John Smith | A001 |
| D002 | Marketing | Jane Doe | A002 |
| D003 | Finance | Bob Johnson | A003 |
| D004 | Human Resources | Sara Lee | A004 |
| D005 | Customer Service | Joe Smith | A005 |

*Figure 29: Rows of department table*

```
SELECT * FROM role;
```

Script Output | Task completed in 2.84 seconds

| ROLE_ID | ROLE_TITLE | SALARY | JOB_ID |
|---|---|---|---|
| R001 | Software Engineer | 100000 | J001 |
| R002 | Marketing Manager | 80000 | J002 |
| R003 | Accountant | 75000 | J003 |
| R004 | HR Coordinator | 60000 | J004 |
| R005 | CSR | 40000 | J005 |

*Figure 30: Rows of role table*

## Employee

```
SELECT * FROM employee;
```

Script Output | Task completed in 2.84 seconds

| EMPLOYEE_I | EMPLOYEE_NAME | DATE_OF_B | CONTACT | DEPARTMENT | ROLE_ID | MANAGER_ID |
|---|---|---|---|---|---|---|
| E001 | John Doe | 01-MAY-90 | 555-1234 | D001 | R001 | E001 |
| E002 | Jane Smith | 15-OCT-85 | 555-5678 | D001 | R001 | E001 |
| E003 | Bob Johnson | 22-MAR-92 | 555-9876 | D003 | R003 | E001 |
| E004 | Sara Lee | 05-DEC-88 | 555-1111 | D004 | R004 | E001 |
| E005 | Joe Smith | 30-JUN-95 | 555-2222 | D005 | R005 | E001 |
| E006 | Ram Smitg | 22-MAR-92 | 555-5609 | D001 | R003 | E001 |

*Figure 31: Rows of Employee table*

**Department**

```
SELECT * FROM department_manager;
```

Script Output

Task completed in 2.84 seconds

```
DEPARTMENT MANAGER_ID
---------- ----------
D001       E001
D002       E002
D003       E003
D004       E004
D005       E005
```

*Figure 32: Rows of department table*

**Employee_Address**

```
SELECT * FROM employee_address;
```

Script Output

Task completed in 2.84 seconds

```
EMPLOYEE_I ADDRESS_ID
---------- ----------
E001       A001
E002       A002
E003       A003
E004       A004
E005       A005
```

*Figure 33: Rows of Employee_Address*

**Employee_email**

```
SELECT * FROM employee_email;
```

Script Output

Task completed in 2.84 seconds

| EMAIL | EMPLOYEE_I |
|---|---|
| john.doe@example.com | E001 |
| jane.doe@example.com | E002 |
| bob.smith@example.com | E003 |
| sara.williams@example.com | E004 |
| adam.johnson@example.com | E005 |

Figure 34: Rows of employee_email

**Job_history**

```
SELECT * FROM job_history;
```

Script Output

Task completed in 2.84 seconds

| EMPLOYEE_I | START_DAT | END_DATE | ROLE_ID | DEPARTMENT |
|---|---|---|---|---|
| E001 | 01-JAN-22 | 31-DEC-22 | R001 | D001 |
| E001 | 01-JAN-21 | 31-DEC-21 | R001 | D001 |
| E002 | 01-FEB-22 | 31-JUL-22 | R002 | D002 |
| E003 | 01-MAR-22 | 30-SEP-22 | R003 | D003 |
| E004 | 01-APR-22 | 31-MAY-22 | R004 | D004 |

*Figure 35: Rows of job_history*

### Voting_details

```
SELECT * FROM voting_details;
```

Script Output | Task completed in 2.84 seconds

```
E003        01-MAR-22 30-SEP-22 R003        D003
E004        01-APR-22 31-MAY-22 R004        D004


VOTER_ID   VOTING_YEAR VOTING_MON CANDIDATE_
---------- ----------- ---------- ----------
E001              2023 January    E002
E002              2023 January    E001
E003              2023 January    E002
E004              2023 January    E003
E005              2023 January    E002
```

*Figure 36: Rows of voting_details*

# Forms

## Dashboard

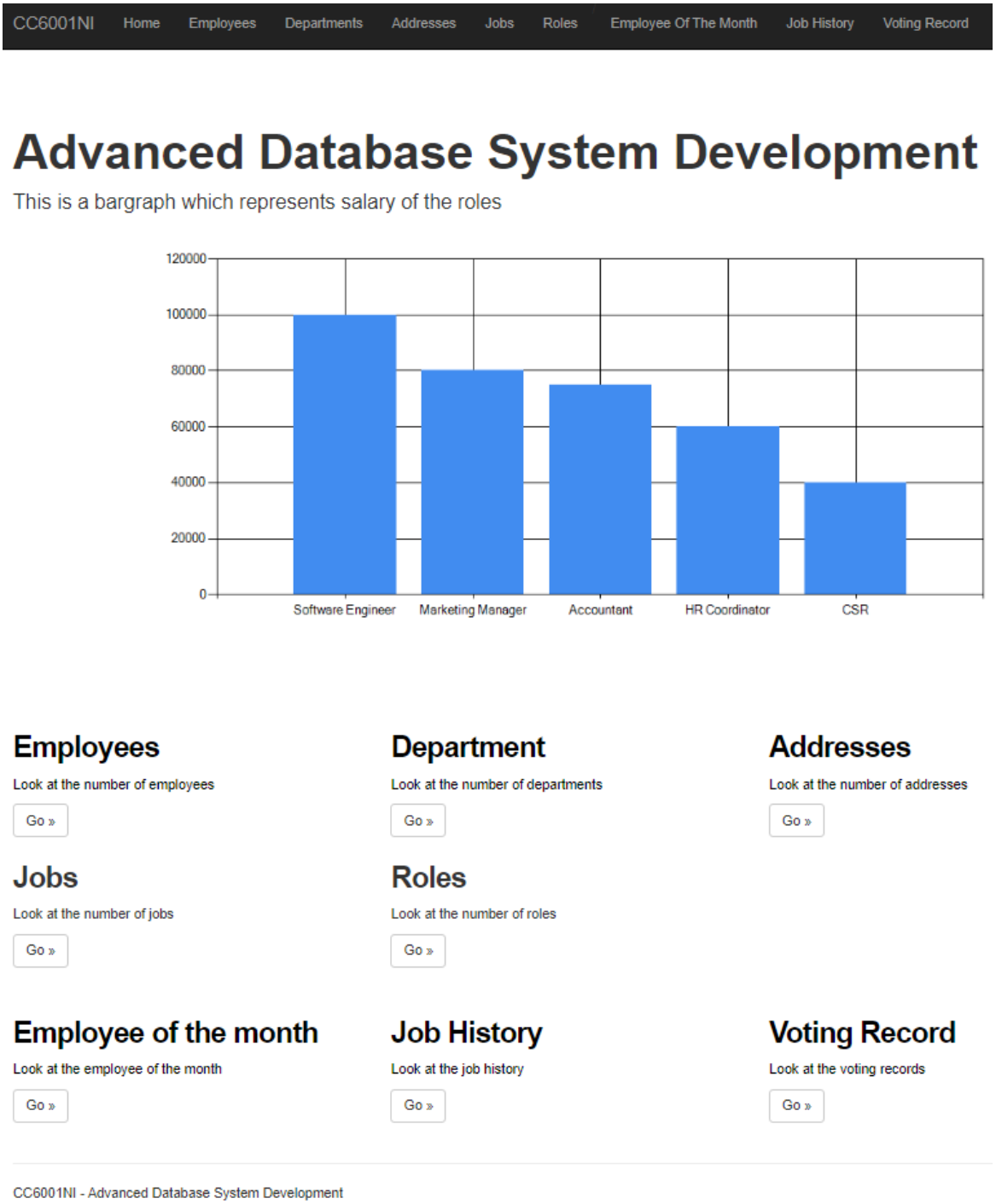



*Figure 37: Dashboard*

## SQL Queries

### Employee of the Month

```
SELECT v.candidate_ID, e.employee_name, e.date_of_birth, e.contact FROM
(
  SELECT candidate_ID, COUNT(*) AS vote_count
  FROM voting_details
  WHERE voting_month = :month AND voting_year = :year
  GROUP BY candidate_ID
  ORDER BY vote_count DESC
) v
JOIN employee e ON e.employee_id = v.candidate_id
WHERE ROWNUM <= 3
```

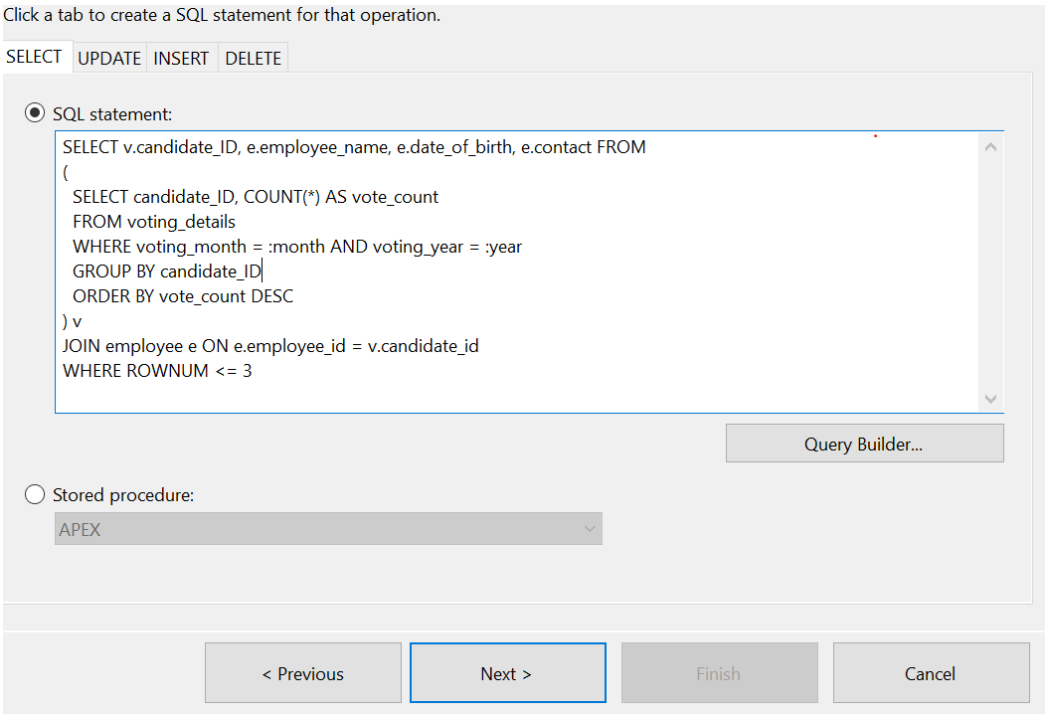



*Figure 38: SQL query for the employee of the month*

**Job History**

SELECT e.employee_ID, e.employee_name, e.date_of_birth, e.contact, d.department_ID, d.department_name, d.department_address, d.department_contact, j.job_ID, j.job_title, h.start_date, h.end_date FROM employee e JOIN department d ON e.department_ID =d.department_ID JOIN role r ON e.role_ID = r.role_ID JOIN job j ON r.job_ID = j.job_ID JOIN job_history h ON e.employee_ID = h.employee_ID WHERE e.employee_ID = :id

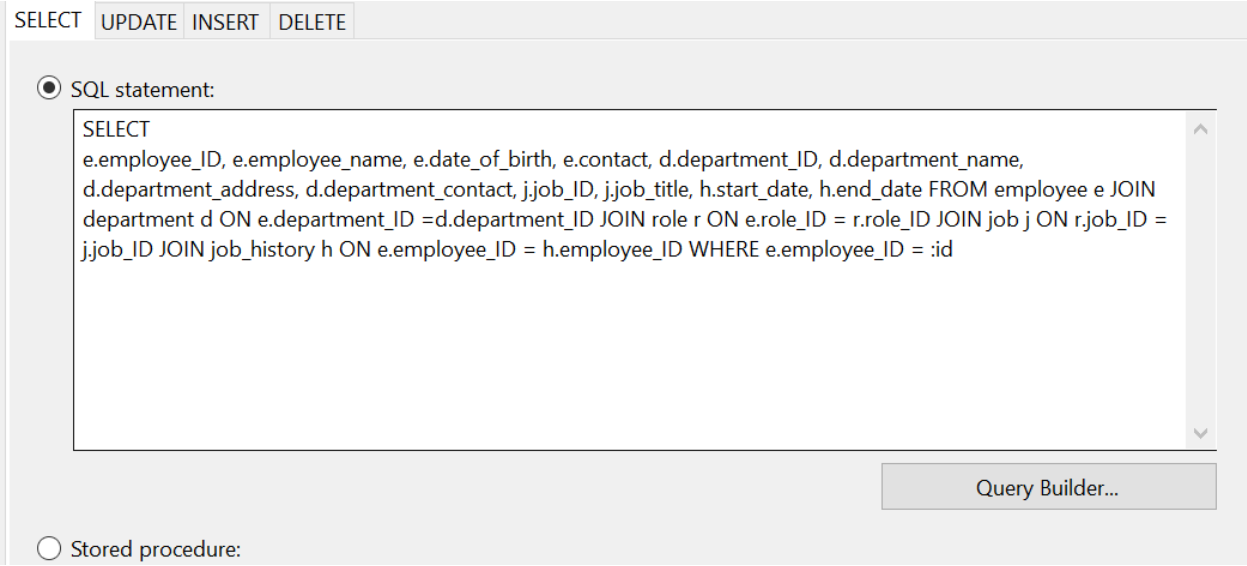



*Figure 39: SQL query for Job History*

**Voting record**

SELECT e.employee_ID, e.employee_name, v.voting_year, v.voting_month, v.candidate_ID From employee e JOIN voting_details v ON e.employee_ID = v.voter_ID WHERE v.voter_ID = :name

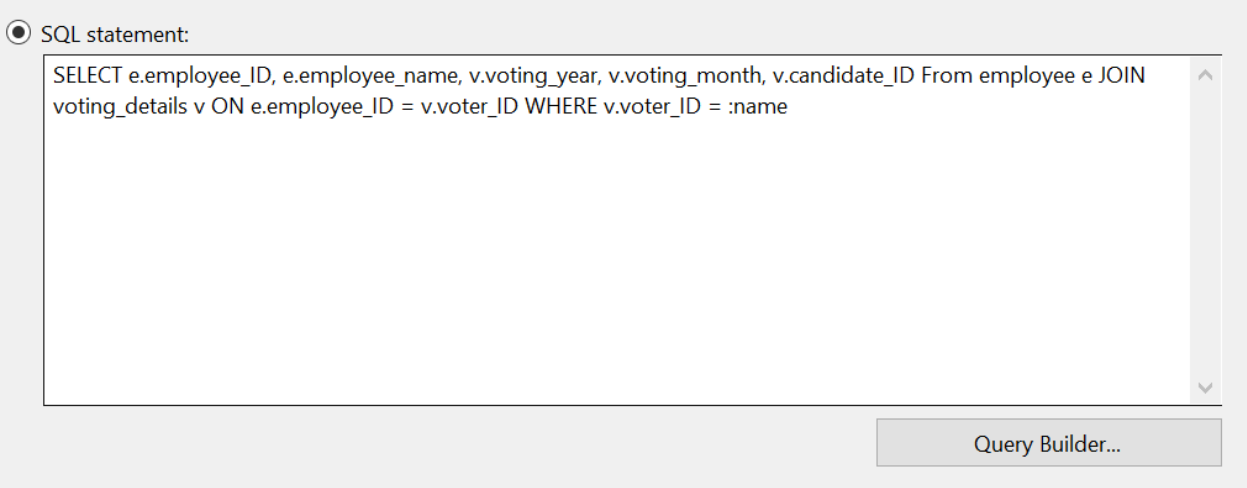



*Figure 40: SQL query for Voting record*

## Complex Forms

## Employee of the month

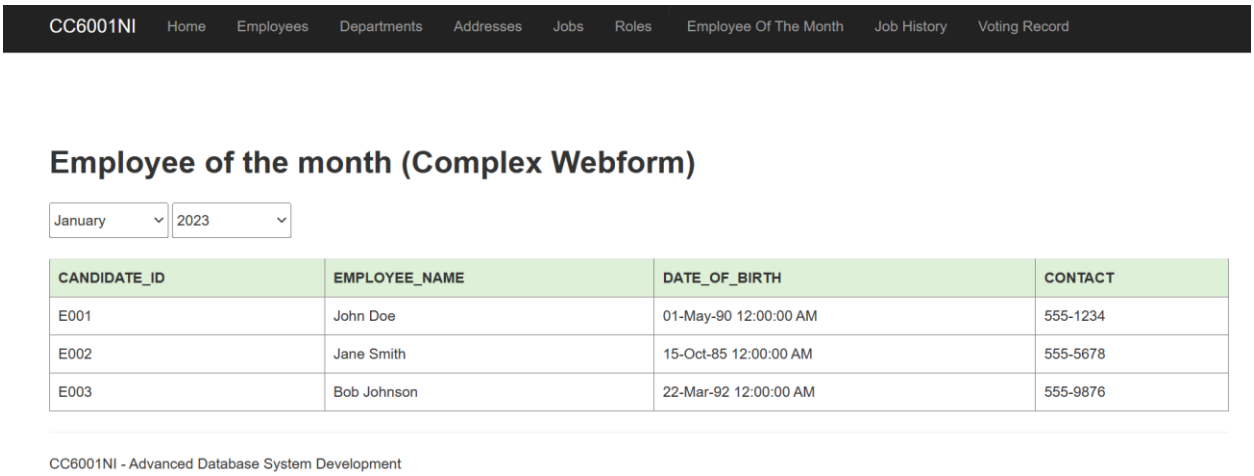



*Figure 41: Employee of the Month*

## Job History

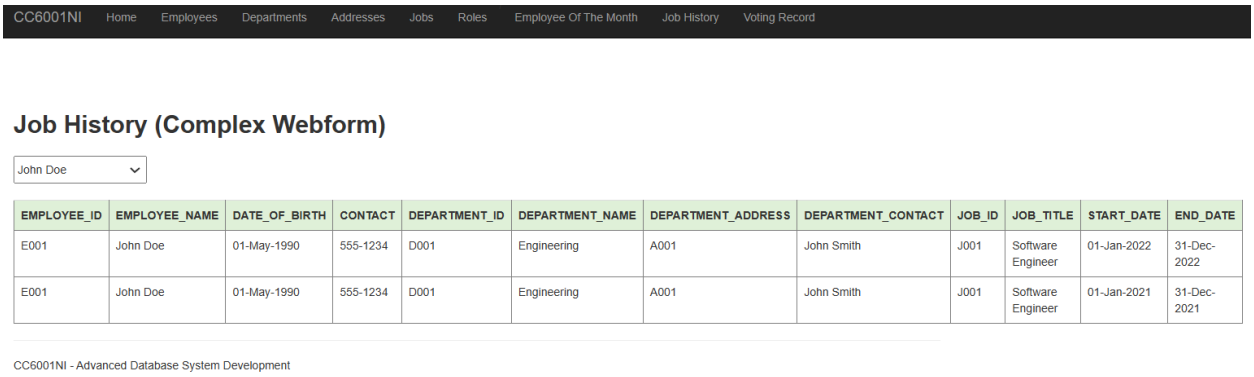



*Figure 42: Job History*

## Voting Record

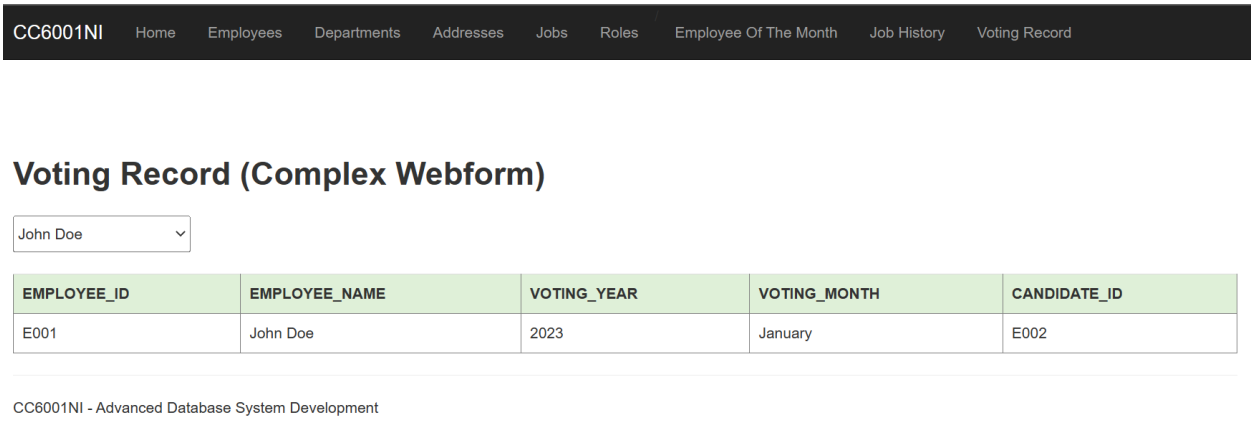



*Figure 43: Voting Record*

## Basic Webforms

## Employee Details

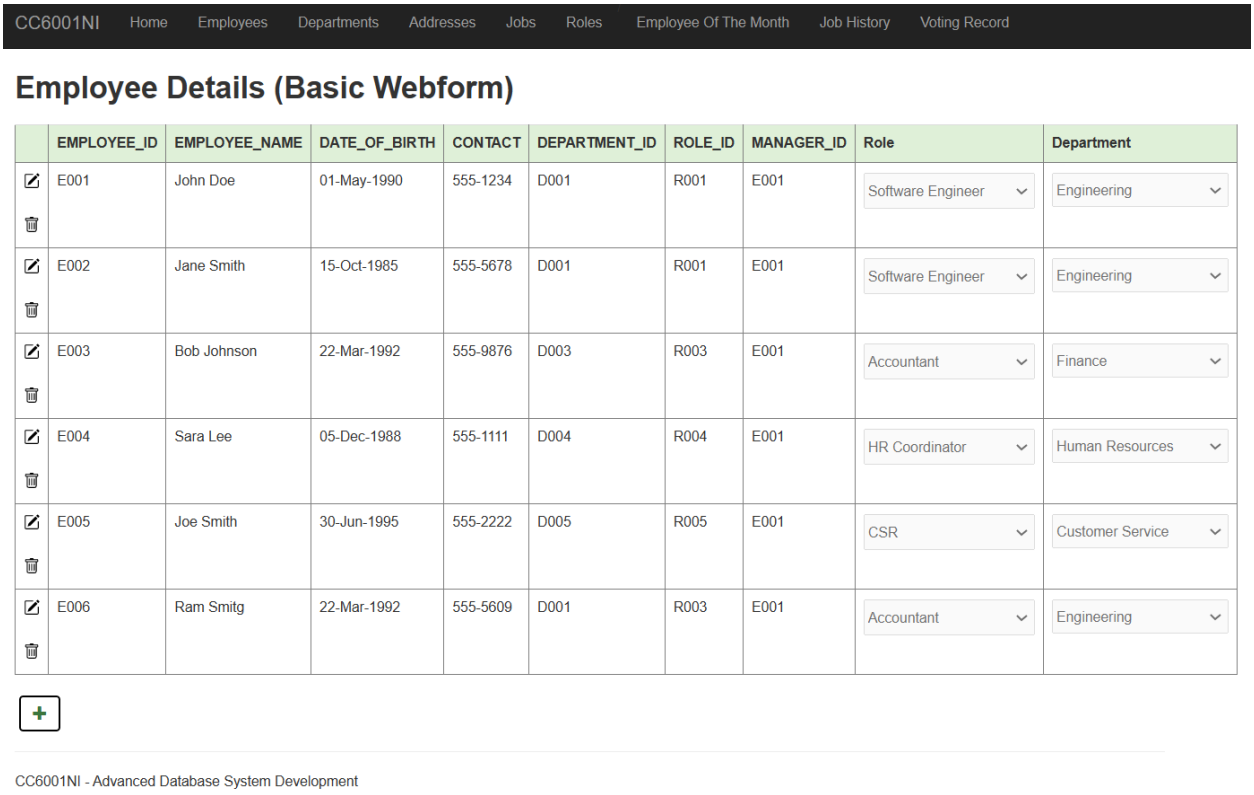



*Figure 44: Employee Details*

## Department Details

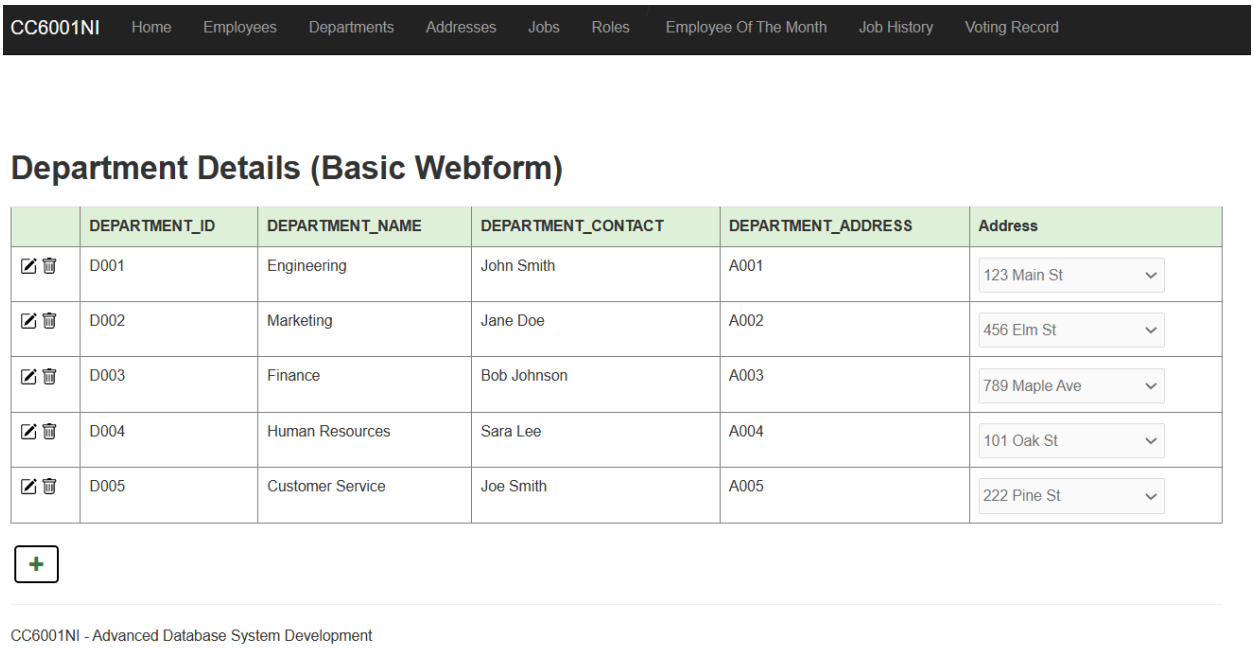



*Figure 45: Department Details*

## Address Details

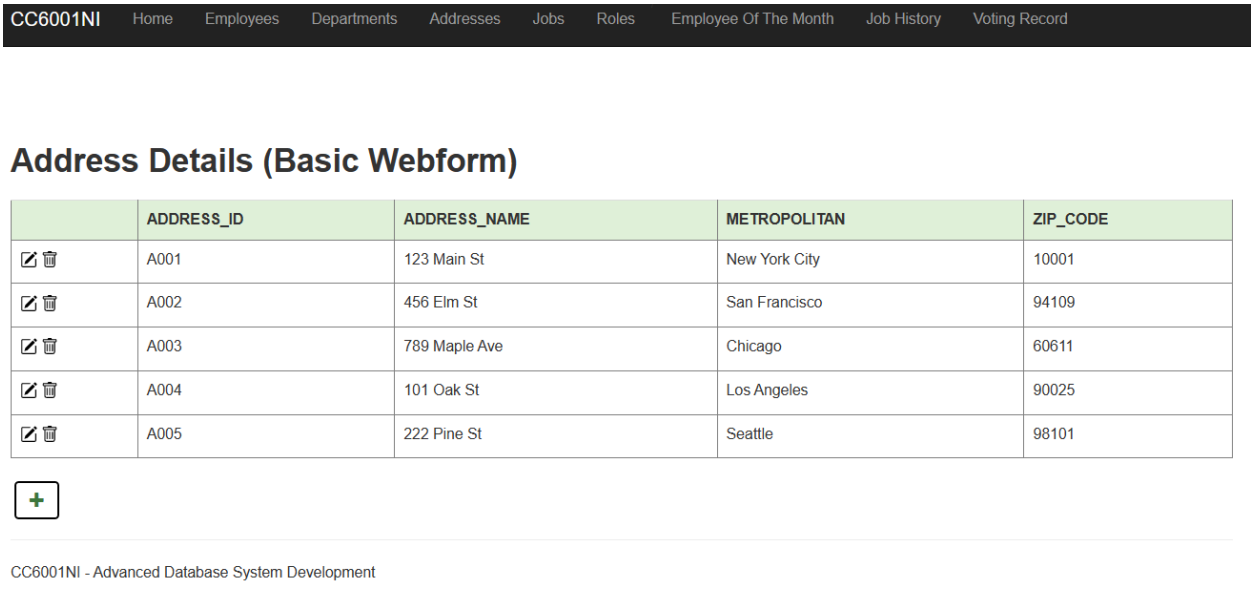
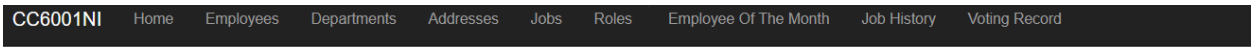



*Figure 46: Address Details*

## Job Details



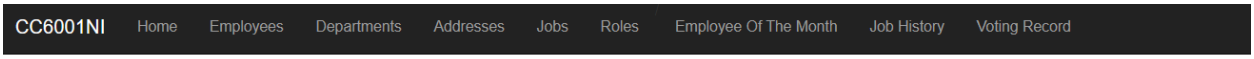

### Job Details (Basic Webform)

| | JOB_ID | JOB_TITLE | JOB_DESCRIPTION |
|---|---|---|---|
| | J001 | Software Engineer | Develop and maintain software solutions |
| | J002 | Marketing Specialist | Create and execute marketing campaigns |
| | J003 | Accountant | Manage financial records and reporting |
| | J004 | Human Resources Coordinator | Assist with hiring and employee relations |
| | J005 | CSR | Assist customers with inquiries and complaints |
| | J006 | Game Development | Develop games |

CC6001NI - Advanced Database System Development

*Figure 47: Job Details*

## Role Details




### Role Details (Basic Webform)

| | ROLE_ID | ROLE_TITLE | SALARY | JOB_ID | Job |
|---|---|---|---|---|---|
| | R001 | Software Engineer | 100000 | J001 | Software Engineer |
| | R002 | Marketing Manager | 80000 | J002 | Marketing Specialist |
| | R003 | Accountant | 75000 | J003 | Accountant |
| | R004 | HR Coordinator | 60000 | J004 | Human Resources Coo |
| | R005 | CSR | 40000 | J005 | CSR |

CC6001NI - Advanced Database System Development

*Figure 48: Role Details*

# Testing

## Employees

### Create and Read

**Employee Details (Basic Webform)**

| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| ✎ 🗑 | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| ✎ 🗑 | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| ✎ 🗑 | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| ✎ 🗑 | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| ✎ 🗑 | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |
| ✎ 🗑 | E006 | Ram Smitg | 22-Mar-1992 | 555-5609 | D001 | R003 | E001 | Accountant | Engineering |

+

*Figure 49: Employee Details*

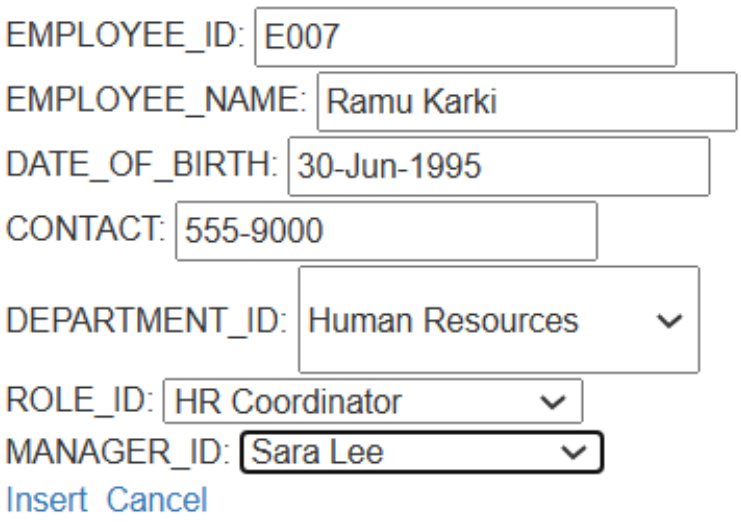



*Figure 50: Adding new to employee details*

## Update

### Employee Details (Basic Webform)

| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |
| | E006 | Ram Smitg | 22-Mar-1992 | 555-5609 | D001 | R003 | E001 | Accountant | Engineering |

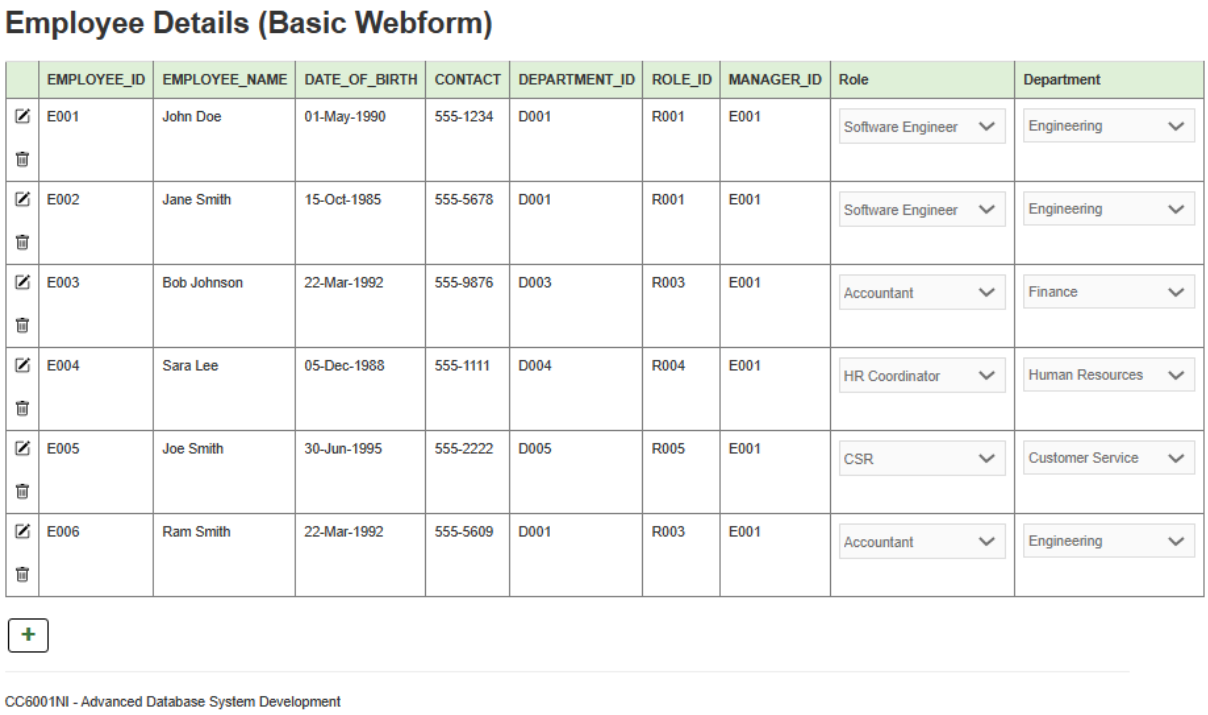
*Figure 51: Update Employee Details 1*

### Employee Details (Basic Webform)

| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |
| | E006 | Ram Smith | 22-Mar-92 12:00:00 AM | 555-5609 | D001 | R003 | E001 | Accountant | Engineering |

CC6001NI - Advanced Database System Development

*Figure 52: Update Employee Details 2*



Employee Details (Basic Webform)

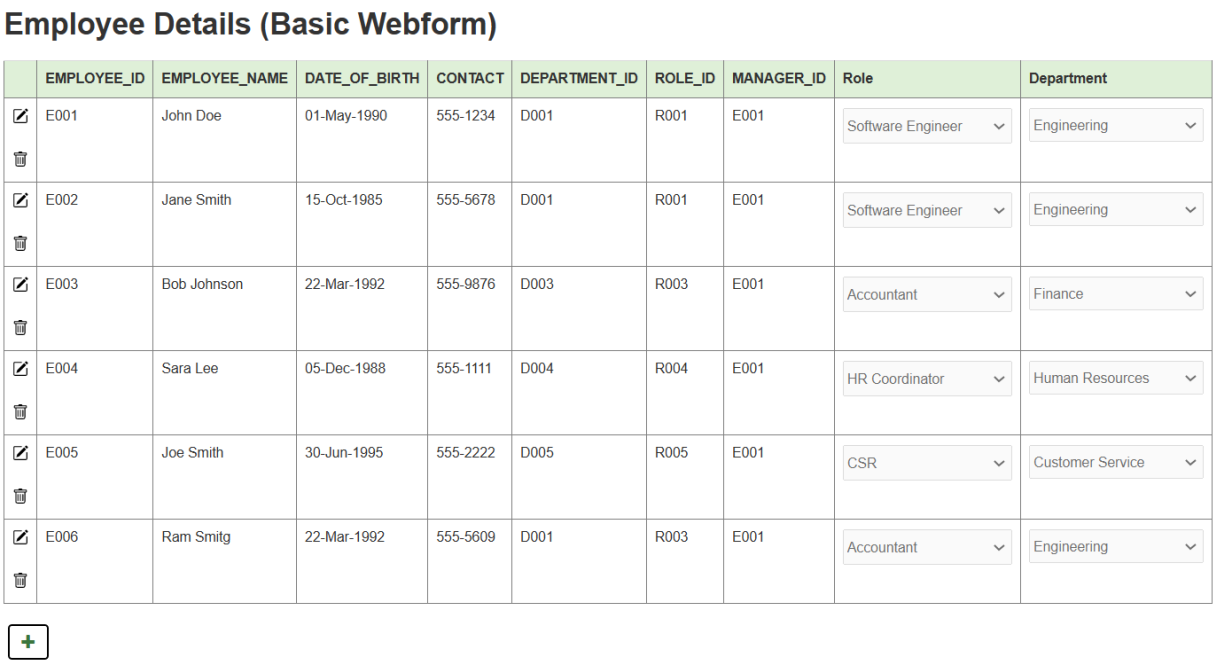
| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |
| | E006 | Ram Smith | 22-Mar-1992 | 555-5609 | D001 | R003 | E001 | Accountant | Engineering |

+

CC6001NI - Advanced Database System Development

*Figure 53: Update Employee Details 3*

## Delete



Employee Details (Basic Webform)

| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |
| | E006 | Ram Smitg | 22-Mar-1992 | 555-5609 | D001 | R003 | E001 | Accountant | Engineering |

+

*Figure 54: Delete Employee Details 1*

## Employee Details (Basic Webform)

| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |

+

CC6001NI - Advanced Database System Development

*Figure 55: Delete employee details 2*

## Departments

## Create and Read

## Department Details (Basic Webform)

| | DEPARTMENT_ID | DEPARTMENT_NAME | DEPARTMENT_CONTACT | DEPARTMENT_ADDRESS | Address |
|---|---|---|---|---|---|
| | D001 | Engineering | John Smith | A001 | 123 Main St |
| | D002 | Marketing | Jane Doe | A002 | 456 Elm St |
| | D003 | Finance | Bob Johnson | A003 | 789 Maple Ave |
| | D004 | Human Resources | Sara Lee | A004 | 101 Oak St |
| | D005 | Customer Service | Joe Smith | A005 | 222 Pine St |

+

CC6001NI - Advanced Database System Development

*Figure 56: Department Details*

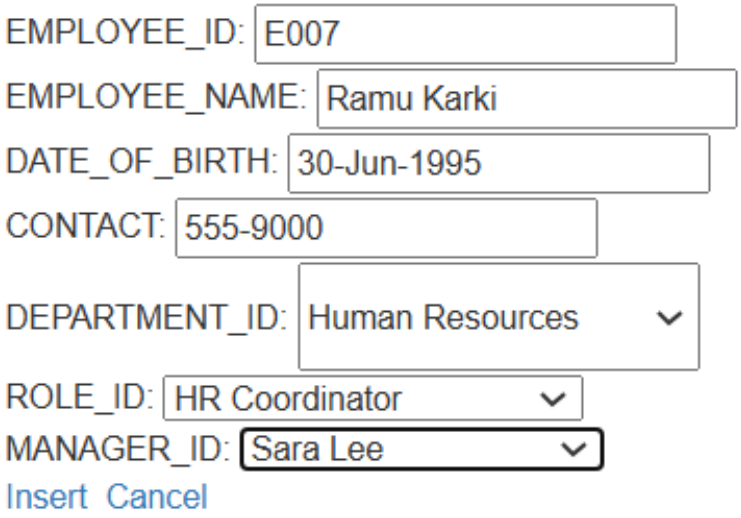



*Figure 57: Adding new to employee details*

## Update

**Employee Details (Basic Webform)**

| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |
| | E006 | Ram Smitg | 22-Mar-1992 | 555-5609 | D001 | R003 | E001 | Accountant | Engineering |

+

*Figure 58: Update Employee Details 1*

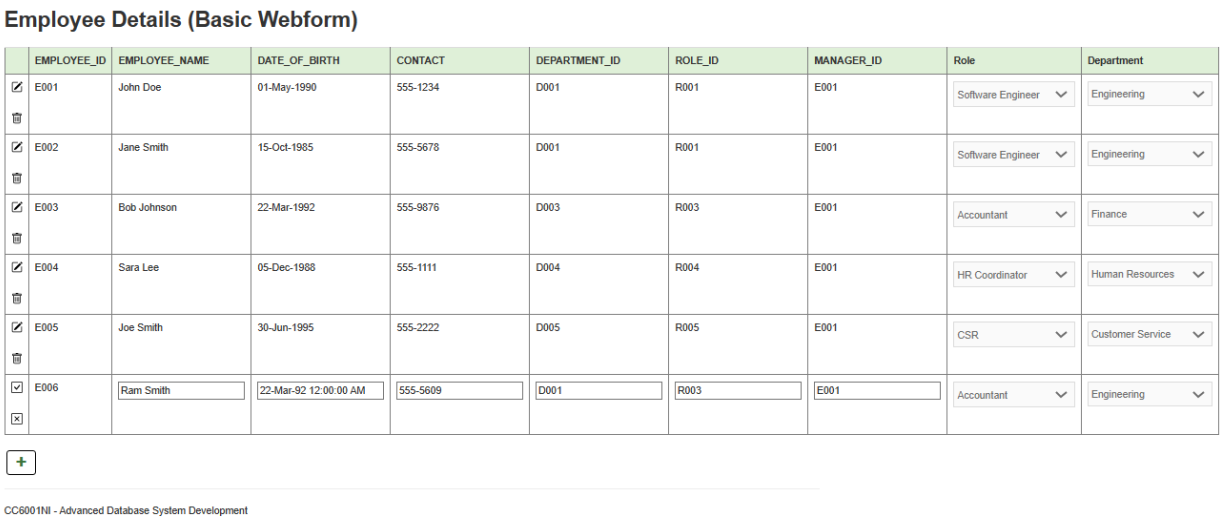



*Figure 59: Update Employee Details 2*

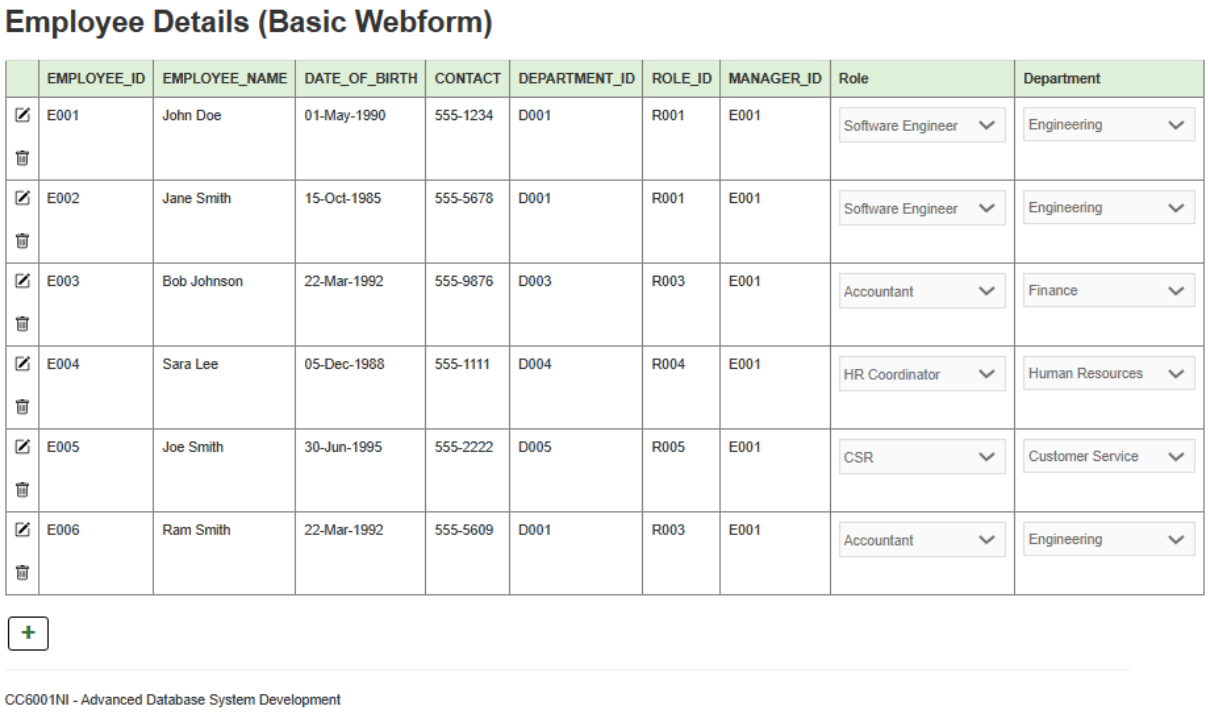



*Figure 60: Update Employee Details 3*

## Delete

### Employee Details (Basic Webform)

| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |
| | E006 | Ram Smitg | 22-Mar-1992 | 555-5609 | D001 | R003 | E001 | Accountant | Engineering |

+

*Figure 61: Delete Employee Details 1*

### Employee Details (Basic Webform)

| | EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | ROLE_ID | MANAGER_ID | Role | Department |
|---|---|---|---|---|---|---|---|---|---|
| | E001 | John Doe | 01-May-1990 | 555-1234 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E002 | Jane Smith | 15-Oct-1985 | 555-5678 | D001 | R001 | E001 | Software Engineer | Engineering |
| | E003 | Bob Johnson | 22-Mar-1992 | 555-9876 | D003 | R003 | E001 | Accountant | Finance |
| | E004 | Sara Lee | 05-Dec-1988 | 555-1111 | D004 | R004 | E001 | HR Coordinator | Human Resources |
| | E005 | Joe Smith | 30-Jun-1995 | 555-2222 | D005 | R005 | E001 | CSR | Customer Service |

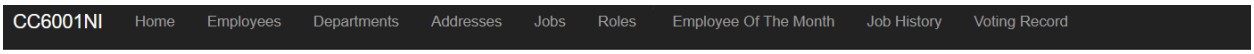
+

CC6001NI - Advanced Database System Development

*Figure 62: Delete employee details 2*

## Addresses

## Create Read




### Address Details (Basic Webform)

| | ADDRESS_ID | ADDRESS_NAME | METROPOLITAN | ZIP_CODE |
|---|---|---|---|---|
| | A001 | 123 Main St | New York City | 10001 |
| | A002 | 456 Elm St | San Francisco | 94109 |
| | A003 | 789 Maple Ave | Chicago | 60611 |
| | A004 | 101 Oak St | Los Angeles | 90025 |
| | A005 | 222 Pine St | Seattle | 98101 |

CC6001NI - Advanced Database System Development

*Figure 63: Read*

### Address Details (Basic Webform)

| | ADDRESS_ID | ADDRESS_NAME | METROPOLITAN | ZIP_CODE |
|---|---|---|---|---|
| | A001 | 123 Main St | New York City | 10001 |
| | A002 | 456 Elm St | San Francisco | 94109 |
| | A003 | 789 Maple Ave | Chicago | 60611 |
| | A004 | 101 Oak St | Los Angeles | 90025 |
| | A005 | 222 Pine St | Seattle | 98101 |

ADDRESS_ID: A006
ADDRESS_NAME: LAGANKHEL
METROPOLITAN: Lalitpur
ZIP_CODE: 80909
Insert Cancel

CC6001NI - Advanced Database System Development

*Figure 64: Create 1*

## Address Details (Basic Webform)

| | ADDRESS_ID | ADDRESS_NAME | METROPOLITAN | ZIP_CODE |
|---|---|---|---|---|
| | A001 | 123 Main St | New York City | 10001 |
| | A002 | 456 Elm St | San Francisco | 94109 |
| | A003 | 789 Maple Ave | Chicago | 60611 |
| | A004 | 101 Oak St | Los Angeles | 90025 |
| | A005 | 222 Pine St | Seattle | 98101 |
| | A006 | LAGANKHEL | Lalitpur | 80909 |

CC6001NI - Advanced Database System Development

*Figure 65: Create 2*

## Update

## Address Details (Basic Webform)

| | ADDRESS_ID | ADDRESS_NAME | METROPOLITAN | ZIP_CODE |
|---|---|---|---|---|
| | A001 | 123 Main St | New York City | 10001 |
| | A002 | 456 Elm St | San Francisco | 94109 |
| | A003 | 789 Maple Ave | Chicago | 60611 |
| | A004 | 101 Oak St | Los Angeles | 90025 |
| | A005 | 222 Pine St | Seattle | 98101 |
| | A006 | LAGANKHELA | Lalitpur | 80909 |

CC6001NI - Advanced Database System Development

*Figure 66: Update 1*

## Address Details (Basic Webform)

| | ADDRESS_ID | ADDRESS_NAME | METROPOLITAN | ZIP_CODE |
|---|---|---|---|---|
| | A001 | 123 Main St | New York City | 10001 |
| | A002 | 456 Elm St | San Francisco | 94109 |
| | A003 | 789 Maple Ave | Chicago | 60611 |
| | A004 | 101 Oak St | Los Angeles | 90025 |
| | A005 | 222 Pine St | Seattle | 98101 |
| | A006 | LAGANKHELA | Lalitpur | 80909 |

CC6001NI - Advanced Database System Development

*Figure 67: Update 2*

## Delete

CC6001NI Home Employees Departments Addresses Jobs Roles Employee Of The Month Job History Voting Record

### Address Details (Basic Webform)

| | ADDRESS_ID | ADDRESS_NAME | METROPOLITAN | ZIP_CODE |
|---|---|---|---|---|
| | A001 | 123 Main St | New York City | 10001 |
| | A002 | 456 Elm St | San Francisco | 94109 |
| | A003 | 789 Maple Ave | Chicago | 60611 |
| | A004 | 101 Oak St | Los Angeles | 90025 |
| | A005 | 222 Pine St | Seattle | 98101 |

CC6001NI - Advanced Database System Development

*Figure 68: Deleted "A006" row*

## Jobs

## Read and Create

### Job Details (Basic Webform)

| | JOB_ID | JOB_TITLE | JOB_DESCRIPTION |
|---|---|---|---|
| | J001 | Software Engineer | Develop and maintain software solutions |
| | J002 | Marketing Specialist | Create and execute marketing campaigns |
| | J003 | Accountant | Manage financial records and reporting |
| | J004 | Human Resources Coordinator | Assist with hiring and employee relations |
| | J005 | CSR | Assist customers with inquiries and complaints |
| | J006 | Game Development | Develop games |

CC6001NI - Advanced Database System Development

*Figure 69: Read*

## Job Details (Basic Webform)

| | JOB_ID | JOB_TITLE | JOB_DESCRIPTION |
|---|---|---|---|
| | J001 | Software Engineer | Develop and maintain software solutions |
| | J002 | Marketing Specialist | Create and execute marketing campaigns |
| | J003 | Accountant | Manage financial records and reporting |
| | J004 | Human Resources Coordinator | Assist with hiring and employee relations |
| | J005 | CSR | Assist customers with inquiries and complaints |
| | J006 | Game Development | Develop games |

JOB_ID: J007
JOB_TITLE: Cleaning
JOB_DESCRIPTION: Cleaning
Insert Cancel

CC6001NI - Advanced Database System Development

*Figure 70: Create 1*

## Job Details (Basic Webform)

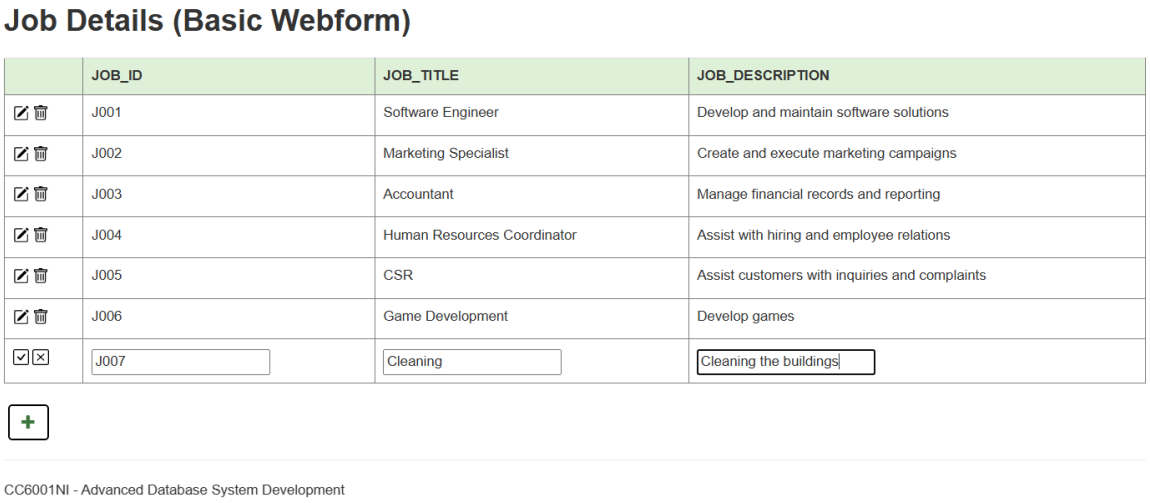
| | JOB_ID | JOB_TITLE | JOB_DESCRIPTION |
|---|---|---|---|
| | J001 | Software Engineer | Develop and maintain software solutions |
| | J002 | Marketing Specialist | Create and execute marketing campaigns |
| | J003 | Accountant | Manage financial records and reporting |
| | J004 | Human Resources Coordinator | Assist with hiring and employee relations |
| | J005 | CSR | Assist customers with inquiries and complaints |
| | J006 | Game Development | Develop games |
| | J007 | Cleaning | Cleaning |

+

CC6001NI - Advanced Database System Development

*Figure 71: Create 2*

## Update

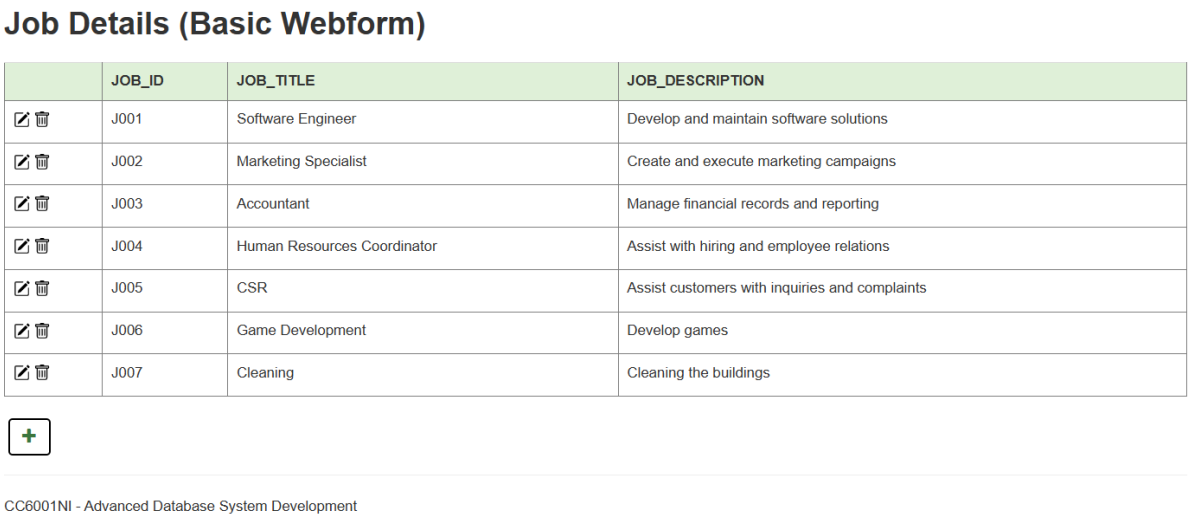


Job Details (Basic Webform)

| | JOB_ID | JOB_TITLE | JOB_DESCRIPTION |
|---|---|---|---|
| | J001 | Software Engineer | Develop and maintain software solutions |
| | J002 | Marketing Specialist | Create and execute marketing campaigns |
| | J003 | Accountant | Manage financial records and reporting |
| | J004 | Human Resources Coordinator | Assist with hiring and employee relations |
| | J005 | CSR | Assist customers with inquiries and complaints |
| | J006 | Game Development | Develop games |
| | J007 | Cleaning | Cleaning the buildings |

CC6001NI - Advanced Database System Development

*Figure 72: Update 1*



Job Details (Basic Webform)

| | JOB_ID | JOB_TITLE | JOB_DESCRIPTION |
|---|---|---|---|
| | J001 | Software Engineer | Develop and maintain software solutions |
| | J002 | Marketing Specialist | Create and execute marketing campaigns |
| | J003 | Accountant | Manage financial records and reporting |
| | J004 | Human Resources Coordinator | Assist with hiring and employee relations |
| | J005 | CSR | Assist customers with inquiries and complaints |
| | J006 | Game Development | Develop games |
| | J007 | Cleaning | Cleaning the buildings |

CC6001NI - Advanced Database System Development

*Figure 73: Update 2*

## Delete

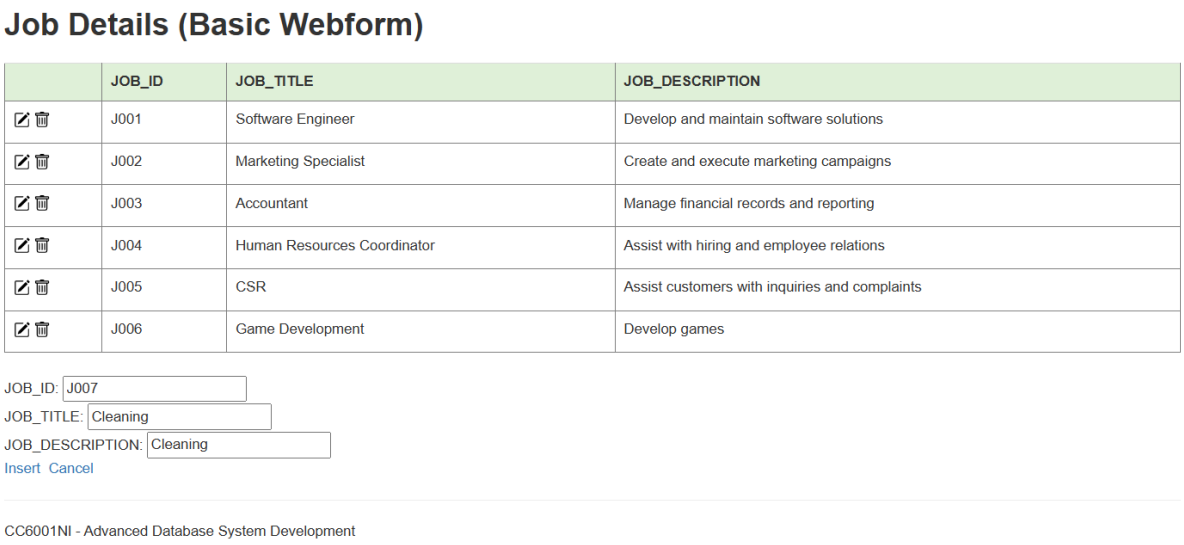



*Figure 74: Deleted "J007" row*

## Roles

## Create and Read


*Figure 75: Read*

The isssue was solvedin create and delete by binding the dropdown to the  job id.

Server Error in '/' Application.

ORA-01400: cannot insert NULL into ("CW"."ROLE"."JOB_ID")

**Description:** An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

**Exception Details:** System.Data.OracleClient.OracleException: ORA-01400: cannot insert NULL into ("CW"."ROLE"."JOB_ID")

**Source Error:**

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

**Stack Trace:**

```
[OracleException (0x80131938): ORA-01400: cannot insert NULL into ("CW"."ROLE"."JOB_ID")
]
   System.Data.OracleClient.OracleConnection.CheckError(OciErrorHandle errorHandle, Int32 rc) +408
   System.Data.OracleClient.OracleCommand.Execute(OciStatementHandle statementHandle, CommandBehavior behavior, Boolean needRowid, OciRowidDescriptor& rowidDescriptor, ArrayList& resultParameterOrdinals) +1844
   System.Data.OracleClient.OracleCommand.ExecuteNonQueryInternal(Boolean needRowid, OciRowidDescriptor& rowidDescriptor) +687
   System.Data.OracleClient.OracleCommand.ExecuteNonQuery() +144
   System.Web.UI.WebControls.SqlDataSourceView.ExecuteDbCommand(DbCommand command, DataSourceOperation operation) +579
   System.Web.UI.DataSourceView.Insert(IDictionary values, DataSourceViewOperationCallback callback) +179
   System.Web.UI.WebControls.FormView.HandleInsert(String commandArg, Boolean causesValidation) +502
   System.Web.UI.WebControls.FormView.HandleEvent(EventArgs e, Boolean causesValidation, String validationGroup) +725
   System.Web.UI.Control.RaiseBubbleEvent(Object source, EventArgs args) +50
   System.Web.UI.WebControls.FormViewRow.OnBubbleEvent(Object source, EventArgs e) +135
   System.Web.UI.Control.RaiseBubbleEvent(Object source, EventArgs args) +50
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +5440
```

**Version Information:** Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.9037.0

*Figure 76: Error encountered*

## Update

### Role Details (Basic Webform)

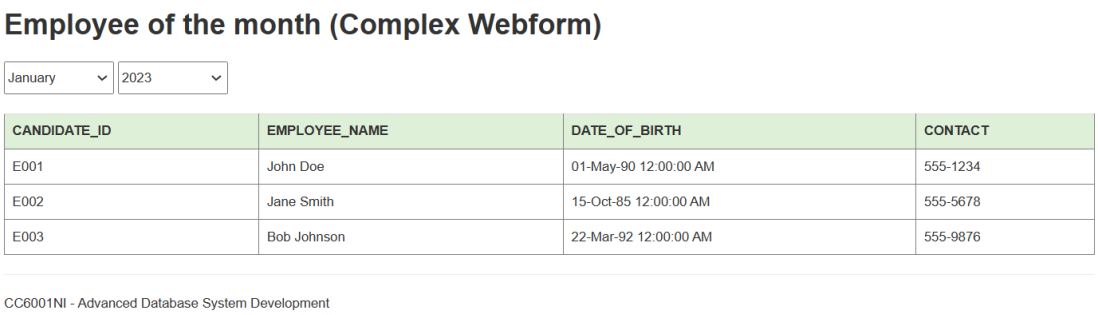
| | ROLE_ID | ROLE_TITLE | SALARY | JOB_ID | Job |
|---|---|---|---|---|---|
| | R001 | Software Engineer | 100000 | J001 | Software Engineer |
| | R002 | Marketing Manager | 80000 | J002 | Marketing Specialist |
| | R003 | Accountant | 75000 | J003 | Accountant |
| | R004 | HR Coordinator | 60000 | J004 | Human Resources Cc |
| | R005 | CSR | 400005 | J005 | CSR |

+

CC6001NI - Advanced Database System Development

*Figure 77: Updated*

## Employee of the Month



Employee of the month (Complex Webform)

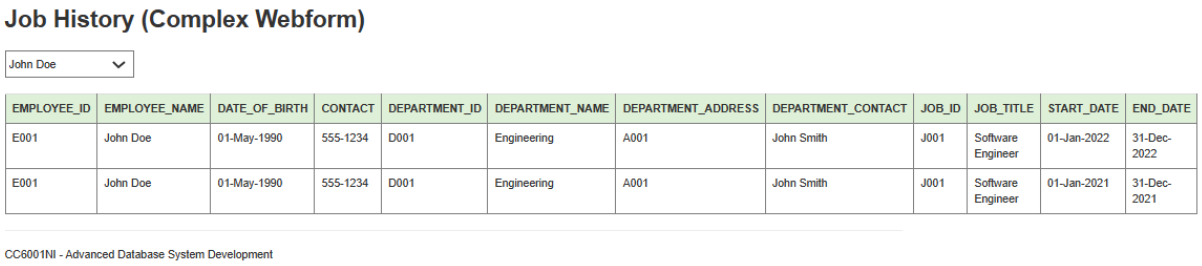
January 2023

| CANDIDATE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT |
|---|---|---|---|
| E001 | John Doe | 01-May-90 12:00:00 AM | 555-1234 |
| E002 | Jane Smith | 15-Oct-85 12:00:00 AM | 555-5678 |
| E003 | Bob Johnson | 22-Mar-92 12:00:00 AM | 555-9876 |

CC6001NI - Advanced Database System Development

*Figure 78: Accoding to date and month*

## Job Details



Job History (Complex Webform)

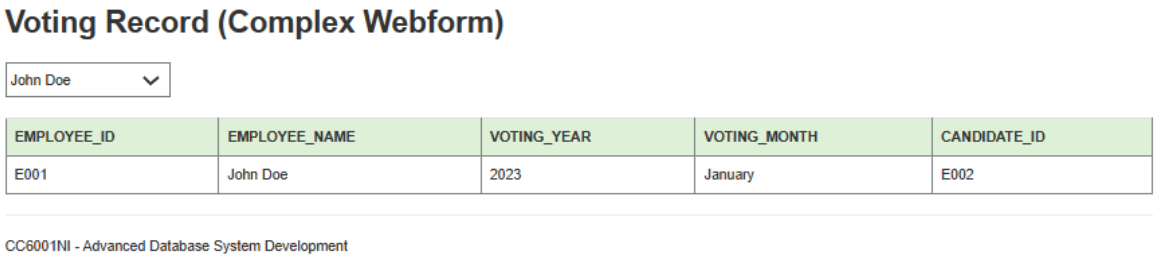
John Doe

| EMPLOYEE_ID | EMPLOYEE_NAME | DATE_OF_BIRTH | CONTACT | DEPARTMENT_ID | DEPARTMENT_NAME | DEPARTMENT_ADDRESS | DEPARTMENT_CONTACT | JOB_ID | JOB_TITLE | START_DATE | END_DATE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| E001 | John Doe | 01-May-1990 | 555-1234 | D001 | Engineering | A001 | John Smith | J001 | Software Engineer | 01-Jan-2022 | 31-Dec-2022 |
| E001 | John Doe | 01-May-1990 | 555-1234 | D001 | Engineering | A001 | John Smith | J001 | Software Engineer | 01-Jan-2021 | 31-Dec-2021 |

CC6001NI - Advanced Database System Development

*Figure 79: According to name*

## Voting Record



Voting Record (Complex Webform)

John Doe

| EMPLOYEE_ID | EMPLOYEE_NAME | VOTING_YEAR | VOTING_MONTH | CANDIDATE_ID |
|---|---|---|---|---|
| E001 | John Doe | 2023 | January | E002 |

CC6001NI - Advanced Database System Development

*Figure 80: According to name*

## Further Discussion

Based on the coursework described above, it was an engaging and valuable experience to work on a project that involved utilizing SQL developer, data modeler, Oracle, ASP.NET, and a database management web application. During the initial stages of the project, we used data modeler to create a conceptual model of the database schema. This helped us to visualize the different tables, their relationships, and the attributes associated with each table. It also allowed us to make changes and adjustments to the database design before implementation.

Once the conceptual model was finalized, we moved onto implementation using Oracle and SQL developer. This involved creating tables, defining relationships between them, and populating them with data. We also created views and stored procedures to facilitate data retrieval and manipulation.The database was graphically designed using the help of data modeler and to provide a user interface for interacting with the database, we utilized ASP.NET to create a web application. This allowed users to view and modify the data in the database using a user-friendly interface. The application was also designed to be secure and scalable, ensuring that it could handle a large number of users and data inputs.

Throughout the project, we utilized various tools to manage the database, including Oracle, SQL developer, data modeler, ASP .NET, visual studio. These tools allowed us to monitor database performance, optimize queries, and make modifications to the database schema when needed. Overall, working on this project was a rewarding experience that provided us with valuable insights into database design and management. The use of tools such as SQL developer, data modeler, Oracle, ASP.NET, and a database management web application allowed us to create a robust and scalable solution that met the needs of the users.