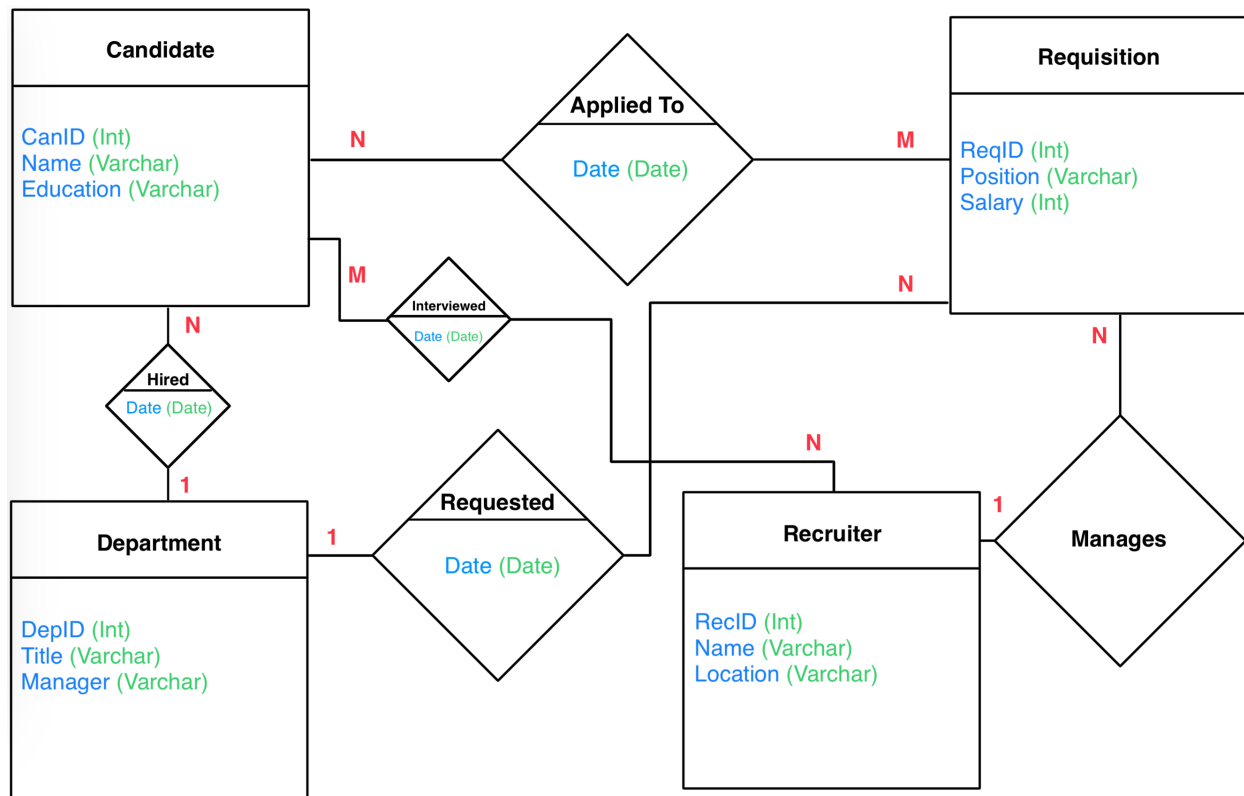# Assignment 7: Step 5 of Your PDA (Personal Database Application)

Candidates (**canID**, canName, education)
Requisitions (**reqID**, depID, recID, title, salary, opnDate, endDate)
Departments (**depID**, depName, manager)
Recruiters (**recID**, recName, location)
AppliedTo (**canID**, **reqID**, **appDate**)
Interviewed (**canID**, **recID, intDate**)

Loading

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| HR                 |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use HR;
Database changed
```

```
mysql> show tables;
+--------------+
| Tables_in_hr |
+--------------+
| AppliedTo    |
| Candidates   |
| Departments  |
| Interviewed  |
| Recruiters   |
| Requisitions |
+--------------+
6 rows in set (0.00 sec)

mysql> describe AppliedTo;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| canID   | varchar(20) | NO   | PRI | NULL    |       |
| reqID   | varchar(20) | NO   | PRI | NULL    |       |
| appDate | date        | NO   | PRI | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> describe Candidates;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| canID     | varchar(20) | NO   | PRI | NULL    |       |
| canName   | varchar(20) | NO   |     | NULL    |       |
| education | varchar(20) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> describe Departments;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| depID   | varchar(20) | NO   | PRI | NULL    |       |
| depName | varchar(20) | NO   |     | NULL    |       |
| manager | varchar(20) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> describe Interviewed;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| canID   | varchar(20) | NO   | PRI | NULL    |       |
| recID   | varchar(20) | NO   | PRI | NULL    |       |
| intDate | date        | NO   | PRI | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> describe Recruiters;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| recID    | varchar(20) | NO   | PRI | NULL    |       |
| recName  | varchar(20) | NO   |     | NULL    |       |
| location | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql> describe Requisitions;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| reqID   | varchar(20) | NO   | PRI | NULL    |       |
| depID   | varchar(20) | NO   | MUL | NULL    |       |
| recID   | varchar(20) | NO   | MUL | NULL    |       |
| title   | varchar(20) | YES  |     | NULL    |       |
| salary  | int         | YES  |     | NULL    |       |
| opnDate | date        | NO   |     | NULL    |       |
| endDate | date        | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Assignment

1) *Showing CREATE TABLE statements with constraints.*

```
CREATE TABLE Requisitions (
reqID varchar(20) NOT NULL,
depID varchar(20) NOT NULL,
recID varchar(20) NOT NULL,
title varchar(20),
salary int,
opnDate date NOT NULL,
endDate date NOT NULL,
Primary Key (reqID),
Foreign Key (depID) References Departments(depID),
Foreign Key (recID) References Recruiters(recID));

CREATE TABLE Interviewed (
canID varchar(20) NOT NULL,
recID varchar(20) NOT NULL,
intDate date NOT NULL,
Primary Key (canID, recID, intDate),
Foreign Key (canID) References Candidates(canID),
Foreign Key (recID) References Recruiters(recID));

CREATE TABLE AppliedTo (
canID varchar(20) NOT NULL,
reqID varchar(20) NOT NULL,
appDate date NOT NULL,
Primary Key (canID, reqID, appDate),
Foreign Key (canID) References Candidates(canID),
Foreign Key (reqID) References Requisitions(reqID));
```

Now, let's attempt to violate a foreign key constraint by inserting a row into the Requisitions table with a depID that does not exist in the Departments table.

```
mysql> INSERT INTO Requisitions (reqID, depID, recID, title, salary,
opnDate, endDate) VALUES ('Req999', 'Dep999', 'Rec999', 'Test Title',
50000, '2024-01-01', '2024-12-31');

ERROR 1452 (23000): Cannot add or update a child row: a foreign key
constraint fails (`hr`.`requisitions`, CONSTRAINT `requisitions_ibfk_1`
FOREIGN KEY (`depID`) REFERENCES `departments` (`depID`))
```

2) To demonstrate *using input parameters and show changes to the database*, we will create a stored procedure that adds a new candidate and applies them to open requisitions with a salary above a certain amount. (i.e Insert a new application for each requisition with a salary above minSalary)

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE ApplyToHighSalaryPositions(IN candidateID
VARCHAR(20), IN minSalary INT)
    -> BEGIN
    ->   INSERT INTO AppliedTo (canID, reqID, appDate)
    ->   SELECT candidateID, reqID, CURDATE()
    ->   FROM Requisitions
    ->   WHERE salary > minSalary AND endDate > CURDATE();
    -> END//
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
```

Using canID 'C11111' we apply to positions with a salary above $100000 using the procedure we created to demonstrate adding a new record to the AppliedTo table using the procedure's input parameters.

```
mysql> SELECT * FROM Candidates WHERE canID = 'C11111';
+--------+-----------+-----------+
| canID  | canName   | education |
+--------+-----------+-----------+
| c11111 | Elizabeth | MBA/MS/MA |
+--------+-----------+-----------+
1 row in set (0.00 sec)

mysql> CALL ApplyToHighSalaryPositions('C11111', 100000);
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM AppliedTo WHERE canID = 'C11111';
+--------+--------+------------+
| canID  | reqID  | appDate    |
+--------+--------+------------+
| c11111 | q14232 | 2024-01-06 |
+--------+--------+------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM Requisitions WHERE reqID = 'q14232';
+--------+-------+-------+-------+--------+------------+------------+
| reqID  | depID | recID | title | salary | opnDate    | endDate    |
+--------+-------+-------+-------+--------+------------+------------+
| q14232 | d8080 | r5027 | CFO   | 237331 | 2024-01-09 | 2024-01-16 |
+--------+-------+-------+-------+--------+------------+------------+
1 row in set (0.00 sec)
```

3) For the Candidates table, we might need to search by education. Thus, creating an index on Education and testing queries speed changes.

Indexing:

```
mysql> CREATE INDEX idx_education ON Candidates(education);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Before Indexing:

1 –
```
mysql> SELECT * FROM Candidates WHERE education = 'MBA/MS/MA';
+--------+-------------+-----------+
| canID  | canName     | education |
+--------+-------------+-----------+
| c10003 | Emily       | MBA/MS/MA |
 * * * * * * * * * * * * * * *
| c9994  | Michael     | MBA/MS/MA |
+--------+-------------+-----------+
4332 rows in set (0.01 sec)
```

2 –
```
mysql> SELECT R.title, D.depName
    -> FROM Requisitions R
    -> JOIN Departments D ON R.depID = D.depID
    -> WHERE D.depName = 'Engineering';
+---------------------+-------------+
| title               | depName     |
+---------------------+-------------+
| Financial Analyst   | Engineering |
 * * * * * * * * * * * * * * * * *
| Project Manager     | Engineering |
+---------------------+-------------+
1529 rows in set (0.02 sec)
```

After Indexing:

1 –
```
mysql> SELECT * FROM Candidates WHERE education = 'MBA/MS/MA';
+--------+-------------+-----------+
| canID  | canName     | education |
+--------+-------------+-----------+
| c10003 | Emily       | MBA/MS/MA |
 * * * * * * * * * * * * * * *
| c9994  | Michael     | MBA/MS/MA |
+--------+-------------+-----------+
4332 rows in set (0.00 sec)
```

2 –
```
mysql> SELECT R.title, D.depName
    -> FROM Requisitions R
    -> JOIN Departments D ON R.depID = D.depID
    -> WHERE D.depName = 'Engineering';
+---------------------+-------------+
| title               | depName     |
+---------------------+-------------+
| Financial Analyst   | Engineering |
 * * * * * * * * * * * * * * * * *
| Project Manager     | Engineering |
+---------------------+-------------+
1529 rows in set (0.01 sec)
```