

Name: Loc Nguyen

Q1. Given the following relational database schema:

FLIGHT = (FlightN, FromCity, ToCity, Date, DepartureTime, ArrivalTime) //. You may use <, >, !=, or = between any two dates or between any two times. Also, you may assume the attribute Date = arrival date= departure date and that ToCity and FromCity are in the same time zone.

TICKET = (TicketN, FlightN, Cost, Completed) //Completed may assume the values 'Yes' or NULL, Null means the flight hasn't been completed.

PASSENGER = (Name, TicketN)

Write DDL statements to create the above tables and use appropriate data types for the attributes. The DDL statement must include at least the following constraints:

Every Primary Key;

Every Foreign Key;

For every Foreign Key constraint, the referential integrity constraints are:

ON DELETE SET NULL or DEFAULT whatever it is appropriate;

ON UPDATE SET NULL or CASCADE whatever it is appropriate;

Any necessary constraints on the attributes' values.

Solution:

```
CREATE TABLE FLIGHT ( FlightN          VARCHAR(7) PRIMARY KEY,
                        FromCity        VARCHAR(3) NOT NULL,
                        ToCity          VARCHAR(3) NOT NULL,
                        Date            DATE NOT NULL,
                        DepartureTime    TIME NOT NULL,
                        ArrivalTime      TIME NOT NULL
                        );

CREATE TABLE TICKET ( TicketN          INT PRIMARY KEY,
                        FlightN          INT,
                        FOREIGN KEY (FlightN) References FLIGHT(FlightN)
                        ON DELETE SET NULL ON UPDATE CASCADE,
                        Cost             FLOAT NOT NULL CHECK(Cost > 0),
                        Completed        VARCHAR(4) CHECK (Completed in
                        ('Yes', NULL))
                        );

CREATE TABLE PASSENGER ( Name          VARCHAR(20) PRIMARY KEY,
                           TicketN      INT,
                           FOREIGN KEY (TicketN) References TICKET(TicketN)
                           ON DELETE SET NULL ON UPDATE CASCADE
                           );
```

Q2. Consider the following relational schema, DDL statements and tables. Show the status of the above tables after each of the following operations:

EMPLOYEE(EmployeeID, EmployeeName, SupervisorID, DepartmentID)

PROJECT (ProjectID, EmployeeID)

DEPARTMENT(Department ID, DepartmentName)

CREATE TABLE EMPLOYEE

(EmployeeID INT PRIMARY KEY,
EmployeeName VARCHAR(50) NOT NULL,
SupervisorID INT DEFAULT 9,
DepartmentID INT,
FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (EmployeeID)
ON UPDATE SET DEFAULT ON DELETE CASCADE
FOREIGN KEY (DepartmentID)REFERENCES DEPARTMENT(DepartmentID);

CREATE TABLE PROJECT (

ProjectID INT PRIMARY KEY,
EmployeeID INT DEFAULT 9,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE (EmployeeID)
ON DELETE SET NULL ON UPDATE CASCADE);

CREATE TABLE DEPARTMENT(

DepartmentID INT PRIMARY KEY,
DepartmentName VARCHAR(50)

);

a. Inserting a new employee whose EmployeeID, EmployeeName, SupervisorID, DepartmentID are 12, F, 6,23. If it is rejected, explain.

Solution: Rejected because it violates foreign key constraint on ‘DepartmentID’ since DepartmentID = 23 doesn’t exist in DEPARTMENT table

EMPLOYEE

| EmployeeID | EmployeeName | SupervisorID | DepartmentID |
|------------|--------------|--------------|--------------|
| 6 | A | 7 | 6 |
| 7 | B | 1 | 6 |
| 1 | C | 8 | 7 |
| 8 | D | 9 | 3 |
| 9 | E | NULL | 6 |

PROJECT

| ProjectID | EmployeeID |
|-----------|------------|
| 1 | 6 |
| 2 | 7 |
| 3 | 1 |
| 4 | 8 |
| 5 | 7 |
| 6 | 1 |

DEPARTMENT

| DepartmentID | DepartmentName |
|--------------|----------------|
| 3 | X |
| 6 | Y |
| 7 | Z |

b. Inserting a new employee whose EmployeeID, EmployeeName, SupervisorID, DepartmentID are 12, F, Null, 3. If it is rejected, explain.

```
CREATE TABLE EMPLOYEE
( EmployeeID INT PRIMARY KEY,
  EmployeeName VARCHAR(50) NOT NULL,
  SupervisorID INT DEFAULT 9,
  DepartmentID INT,
  FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (EmployeeID)
  ON UPDATE SET DEFAULT ON DELETE CASCADE
  FOREIGN KEY (DepartmentID) REFERENCES DEPARTMENT (DepartmentID);
```

```
CREATE TABLE PROJECT (
ProjectID INT PRIMARY KEY,
EmployeeID INT DEFAULT 9,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE (EmployeeID)
ON DELETE SET NULL ON UPDATE CASCADE);
```

```
CREATE TABLE DEPARTMENT(
DepartmentID INT PRIMARY KEY,
DepartmentName VARCHAR(50)
```

);

Solution: Accepted. Tables updated
EMPLOYEE

| EmployeeID | EmployeeName | SupervisorID | DepartmentID |
|------------|--------------|--------------|--------------|
| 6 | A | 7 | 6 |
| 7 | B | 1 | 6 |
| 1 | C | 8 | 7 |
| 8 | D | 9 | 3 |
| 9 | E | NULL | 6 |
| 12 | F | NULL | 3 |

PROJECT

| ProjectID | EmployeeID |
|-----------|------------|
| 1 | 6 |
| 2 | 7 |
| 3 | 1 |
| 4 | 8 |
| 5 | 7 |
| 6 | 1 |

DEPARTMENT

| DepartmentID | DepartmentName |
|--------------|----------------|
| 3 | X |
| 6 | Y |
| 7 | Z |

c. Deleting the employee whose ID= 1.If it is rejected, explain.

```
CREATE TABLE EMPLOYEE
( EmployeeID      INT      PRIMARY KEY,
  EmployeeName  VARCHAR(50)  NOT NULL,
  SupervisorID  INT  DEFAULT 9,
  DepartmentID  INT,
  FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (EmployeeID)
  ON UPDATE SET DEFAULT ON DELETE CASCADE
  FOREIGN KEY (DepartmentID)REFERENCES DEPARTMENT(DepartmentID);
```

```
CREATE TABLE PROJECT (
ProjectID  INT      PRIMARY KEY,
EmployeeID INT  DEFAULT 9,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE (EmployeeID)
ON DELETE SET NULL ON UPDATE CASCADE);
```

```
CREATE TABLE DEPARTMENT(
DepartmentID INT PRIMARY KEY,
DepartmentName VARCHAR(50)
```

);

Solution: Accepted. Updated table below. EmployeeID = 1 row will be deleted and NULL value is set in the PROJECT table. SupervisorID will be set to 9 as default on update.

EMPLOYEE

| EmployeeID | EmployeeName | SupervisorID | DepartmentID |
|------------|--------------|--------------|--------------|
| 6 | A | 7 | 6 |
| 7 | B | 9 | 6 |
| 8 | D | 9 | 3 |
| 9 | E | NULL | 6 |
| 12 | F | NULL | 3 |

PROJECT

| ProjectID | EmployeeID |
|-----------|------------|
| 1 | 6 |
| 2 | 7 |
| 3 | NULL |
| 4 | 8 |
| 5 | 7 |
| 6 | NULL |

DEPARTMENT

| DepartmentID | DepartmentName |
|--------------|----------------|
| 3 | X |
| 6 | Y |
| 7 | Z |

d. Inserting a new department with ID = 21 and name= W. If it is rejected, explain.

```
CREATE TABLE EMPLOYEE
( EmployeeID      INT      PRIMARY KEY,
  EmployeeName  VARCHAR(50)  NOT NULL,
  SupervisorID  INT  DEFAULT 9,
  DepartmentID  INT,
  FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (EmployeeID)
  ON UPDATE SET DEFAULT ON DELETE CASCADE
  FOREIGN KEY (DepartmentID) REFERENCES DEPARTMENT (DepartmentID);
```

```
CREATE TABLE PROJECT (
ProjectID  INT      PRIMARY KEY,
EmployeeID INT  DEFAULT 9,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE (EmployeeID)
ON DELETE SET NULL ON UPDATE CASCADE);
```

```
CREATE TABLE DEPARTMENT(
DepartmentID INT PRIMARY KEY,
DepartmentName VARCHAR(50)
```

);

Solution: Accepted. Updated table below. Add new row to DEPARTMENT table.

EMPLOYEE

| EmployeeID | EmployeeName | SupervisorID | DepartmentID |
|------------|--------------|--------------|--------------|
| 6 | A | 7 | 6 |
| 7 | B | 9 | 6 |
| 8 | D | 9 | 3 |
| 9 | E | NULL | 6 |
| 12 | F | NULL | 3 |

PROJECT

| ProjectID | EmployeeID |
|-----------|------------|
| 1 | 6 |
| 2 | 7 |
| 3 | NULL |
| 4 | 8 |
| 5 | 7 |
| 6 | NULL |

DEPARTMENT

| DepartmentID | DepartmentName |
|--------------|----------------|
| 3 | X |
| 6 | Y |
| 7 | Z |
| 21 | W |

e. Changing the EmployeeID in the table EMPLOYEE from 7 to 10.

```
CREATE TABLE EMPLOYEE
( EmployeeID INT PRIMARY KEY,
  EmployeeName VARCHAR(50) NOT NULL,
  SupervisorID INT DEFAULT 9,
  DepartmentID INT,
  FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (EmployeeID)
  ON UPDATE SET DEFAULT ON DELETE CASCADE
  FOREIGN KEY (DepartmentID) REFERENCES DEPARTMENT (DepartmentID);
```

```
CREATE TABLE PROJECT (
ProjectID INT PRIMARY KEY,
EmployeeID INT DEFAULT 9,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE (EmployeeID)
ON DELETE SET NULL ON UPDATE CASCADE);
```

```
CREATE TABLE DEPARTMENT(
DepartmentID INT PRIMARY KEY,
DepartmentName VARCHAR(50)
```

);

Solution: Accepted. Updated table below. On update set SupervisorID to 9 as default in Employee table. In project table, employeeID will be cascade and update to 10.

EMPLOYEE

| EmployeeID | EmployeeName | SupervisorID | DepartmentID |
|------------|--------------|--------------|--------------|
| 6 | A | 9 | 6 |
| 10 | B | 9 | 6 |
| 8 | D | 9 | 3 |
| 9 | E | NULL | 6 |
| 12 | F | NULL | 3 |

PROJECT

| ProjectID | EmployeeID |
|-----------|------------|
| 1 | 6 |
| 2 | 10 |
| 3 | NULL |
| 4 | 8 |
| 5 | 10 |
| 6 | NULL |

DEPARTMENT

| DepartmentID | DepartmentName |
|--------------|----------------|
| 3 | X |
| 6 | Y |
| 7 | Z |
| 21 | W |

f. Insert a new project with ProjectID= 13 and EmployeeID= 10. If it is rejected, explain.

```
CREATE TABLE EMPLOYEE
( EmployeeID      INT      PRIMARY KEY,
  EmployeeName  VARCHAR(50)  NOT NULL,
  SupervisorID  INT  DEFAULT 9,
  DepartmentID  INT,
  FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (EmployeeID)
  ON UPDATE SET DEFAULT ON DELETE CASCADE
  FOREIGN KEY (DepartmentID)REFERENCES DEPARTMENT(DepartmentID);
```

```
CREATE TABLE PROJECT (
ProjectID  INT      PRIMARY KEY,
EmployeeID INT  DEFAULT 9,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE (EmployeeID)
ON DELETE SET NULL ON UPDATE CASCADE);
```

```
CREATE TABLE DEPARTMENT(
DepartmentID INT PRIMARY KEY,
DepartmentName VARCHAR(50)
```

```
);
```

Solution: Accepted. Insert new row in Project table. Updated table below
EMPLOYEE

| EmployeeID | EmployeeName | SupervisorID | DepartmentID |
|------------|--------------|--------------|--------------|
| 6 | A | 9 | 6 |
| 10 | B | 9 | 6 |
| 8 | D | 9 | 3 |
| 9 | E | NULL | 6 |
| 12 | F | NULL | 3 |

PROJECT

| ProjectID | EmployeeID |
|-----------|------------|
| 1 | 6 |
| 2 | 10 |
| 3 | NULL |
| 4 | 8 |
| 5 | 10 |
| 6 | NULL |
| 13 | 10 |

DEPARTMENT

| DepartmentID | DepartmentName |
|--------------|----------------|
| 3 | X |
| 6 | Y |
| 7 | Z |
| 21 | W |

g. Deleting the project with the ProjectID= 1. If it is rejected , explain.

CREATE TABLE EMPLOYEE

```
( EmployeeID      INT      PRIMARY KEY,
  EmployeeName  VARCHAR(50)  NOT NULL,
  SupervisorID  INT  DEFAULT 9,
  DepartmentID  INT,
  FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (EmployeeID)
  ON UPDATE SET DEFAULT ON DELETE CASCADE
  FOREIGN KEY (DepartmentID)REFERENCES DEPARTMENT(DepartmentID);
```

CREATE TABLE PROJECT (

```
ProjectID  INT      PRIMARY KEY,
EmployeeID INT  DEFAULT 9,
FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE (EmployeeID)
ON DELETE SET NULL ON UPDATE CASCADE);
```

CREATE TABLE DEPARTMENT(

```
DepartmentID INT PRIMARY KEY,
DepartmentName VARCHAR(50)
```

);

Solution: Accepted. Row deleted in Project table. Table updated.

EMPLOYEE

| EmployeeID | EmployeeName | SupervisorID | DepartmentID |
|------------|--------------|--------------|--------------|
| 6 | A | 9 | 6 |
| 10 | B | 9 | 6 |
| 8 | D | 9 | 3 |
| 9 | E | NULL | 6 |
| 12 | F | NULL | 3 |

PROJECT

| ProjectID | EmployeeID |
|-----------|------------|
| 2 | 10 |
| 3 | NULL |
| 4 | 8 |
| 5 | 10 |
| 6 | NULL |
| 13 | 10 |

DEPARTMENT

| DepartmentID | DepartmentName |
|--------------|----------------|
| 3 | X |
| 6 | Y |
| 7 | Z |
| 21 | W |