

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.

```
create table FLIGHT
(
    FlightN int PRIMARY KEY,
    FromCity varchar(50) NOT NULL,
    ToCity varchar(50) NOT NULL,
    Date DATE,
    DepartureTime TIME,
    ArrivalTime TIME,
    CHECK (ArrivalTime > DepartureTime)
);

create table TICKET
(
    TicketN int PRIMARY KEY,
    FlightN int,
    Cost decimal(10, 2),
    Completed varchar(3),
    FOREIGN KEY(FlightN) REFERENCES FLIGHT(FlightN) ON DELETE SET NULL,
    CHECK (Completed = 'Yes')
);

create table PASSENGER
(
    Name varchar(50),
    TicketN int,
    FOREIGN KEY(TicketN) REFERENCES TICKET(TicketN) ON DELETE 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.

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

Answer: Since there is no 23 in Department ID's primary key, the entry will be rejected.

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)
```

```
);
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

Answer: The entry is accepted because the DepartmentID of 3 is a valid primary key, therefore it can be accepted as a foreign key in the entry. Also, the SupervisorID foreign key can be set to NULL.

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)
```

```
);
EMPLOYEE
```

EmployeeID	EmployeeName	SupervisorID	DepartmentID
8	D	9	3
9	E	NULL	6

PROJECT

ProjectID	EmployeeID
1	NULL
2	NULL
3	NULL
4	8
5	NULL
6	NULL

DEPARTMENT

DepartmentID	DepartmentName
3	X
6	Y
7	Z

Answer: The deletion is accepted. Since FOREIGN KEY (SupervisorID) REFERENCES EMPLOYEE (EmployeeID) ... ON DELETE CASCADE, it will delete employee 7 since its SupervisorID was 1, this will also affect employee 6 since employee 7 was its SupervisorID. These 3 employees will be deleted. All projects related to employees 6, 7, and 1 will be set to NULL.

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)
```

```
);
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
21	W

Answer: The entry is accepted; the primary key does not exist so it can be added as a new instance. Also, there is no rules on foreign keys so nothing else will change.

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)
```

```
);
EMPLOYEE
```

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

PROJECT

ProjectID	EmployeeID
1	8
2	10
3	1
4	8
5	10
6	1

DEPARTMENT

DepartmentID	DepartmentName
3	X
6	Y
7	Z

Answer: The update is accepted. Since FOREIGN KEY(SupervisorID) REFERENCES EMPLOYEE(EmployeeID) ON UPDATE SET DEFAULT and since SupervisorID INT DEFAULT 9, Employee 6's supervisor (which was 7 and updated to 10) will be set to the default, which is 9.

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)
```

```
);
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

Answer: The entry is rejected since there is no EmployeeID primary key of 10 to be used as a foreign key for the project.

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)
```

```
);
```

```
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
2	7
3	1
4	8
5	7
6	1

```
DEPARTMENT
```

DepartmentID	DepartmentName
3	X
6	Y
7	Z

Answer: The deletion is accepted since ProjectID is a primary key with no restrictions.