

Domas Budrys Assignment 7 Script

Create Table:

```
CREATE TABLE Employee (  
EMP_ID          INT          NOT NULL,  
EMP_FNAME       VARCHAR(255) NOT NULL,  
EMP_LNAME       VARCHAR(255) NOT NULL,  
EMP_STREET      VARCHAR(255) NULL,  
EMP_CITY        VARCHAR(255) NULL,  
EMP_STATE       VARCHAR(255) NULL,  
EMP_ZIP         VARCHAR(255) NULL,  
EMP_START_DATE  DATE         NOT NULL
```

```
CONSTRAINT Employee_PK          PRIMARY KEY (EMP_ID)
```

```
);
```

```
CREATE TABLE Area (  
AREA_ID          INT          NOT NULL,  
AREA_NAME        VARCHAR(255) NOT NULL,  
AREA_SUPERVISOR_EMPLOYEE_ID INT          NOT NULL,
```

```
CONSTRAINT Area_PK          PRIMARY KEY (AREA_ID)
```

```
);
```

```
CREATE TABLE [Table](  
TABLE_ID          INT          NOT NULL,  
AREA_ID           INT          NOT NULL,  
TABLE_SEATS       INT          NOT NULL,
```

```
CONSTRAINT Table_PK          PRIMARY KEY (TABLE_ID),
```

```
CONSTRAINT Table_FK1        FOREIGN KEY (AREA_ID)  
                             REFERENCES Area(AREA_ID)
```

```
);
```


Question 1:

```
CREATE PROCEDURE Customer_Waited @waited VARCHAR(100)
AS

IF NOT EXISTS(

SELECT C.CUST_ID, C.CUST_LAST_NAME, C.CUST_NUMBER_OF_GUESTS
FROM Customer AS C
JOIN Seating AS S
ON C.CUST_ID = S.CUST_ID

JOIN Assignment AS A
ON S.TABLE_ID = A.TABLE_ID

JOIN Employee AS E
ON A.EMP_ID = E.EMP_ID

WHERE E.EMP_ID = @waited
)
PRINT 'Employee ID does not exists'

ELSE IF NOT EXISTS (

SELECT C.CUST_ID, C.CUST_LAST_NAME, C.CUST_NUMBER_OF_GUESTS
FROM Customer AS C
JOIN Seating AS S
ON C.CUST_ID = S.CUST_ID

JOIN Assignment AS A
ON S.TABLE_ID = A.TABLE_ID

WHERE A.EMP_ID = @waited
)
```

PRINT 'Have not waited any tables'

ELSE

SELECT C.CUST_ID, C.CUST_LAST_NAME, C.CUST_NUMBER_OF_GUESTS
FROM Customer AS C
JOIN Seating AS S
ON C.CUST_ID = S.CUST_ID

JOIN Assignment AS A
ON S.TABLE_ID = A.TABLE_ID

WHERE A.EMP_ID = @waited

Exec Customer_Waited 1

Question 2:

```
CREATE PROCEDURE Seat_Number @number INT
AS

IF EXISTS (
SELECT *
FROM [Table] AS T

WHERE T.TABLE_SEATS < @number
)

PRINT 'There is no table with that many seats'

ELSE

SELECT T.TABLE_ID, A.AREA_ID, A.AREA_NAME
FROM [Table] AS T
JOIN Area AS A
ON T.AREA_ID = A.AREA_ID

WHERE T.TABLE_SEATS >= @number
```

Exec Seat_Number 1

Question 3:

```
CREATE PROCEDURE Cust_Emp @Cust INT, @Emp INT
AS
```

```
IF NOT EXISTS (
SELECT S.SEATING_DATE
FROM Seating AS S
JOIN Customer AS C
ON C.CUST_ID = S.CUST_ID
```

```
JOIN Assignment AS A
ON S.TABLE_ID = A.TABLE_ID
```

```
JOIN Employee AS E
ON A.EMP_ID = E.EMP_ID
```

```
WHERE C.CUST_ID = @Cust
AND
E.EMP_ID = @Emp
)
PRINT 'Employee have not reserved this customer yet'
```

```
ELSE
```

```
SELECT S.SEATING_DATE
FROM Seating AS S
JOIN Customer AS C
ON C.CUST_ID = S.CUST_ID
```

```
JOIN Assignment AS A
ON S.TABLE_ID = A.TABLE_ID
```

```
JOIN Employee AS E
ON A.EMP_ID = E.EMP_ID
```

```
WHERE C.CUST_ID = @Cust
AND
E.EMP_ID = @Emp
```

```
Exec Cust_Emp 1, 1
```

Trigger 1:

```
CREATE TRIGGER Seating_Table_Error ON Seating
FOR INSERT, UPDATE
AS
```

```
    DECLARE @seat_num INT
```

```
    IF EXISTS (
    SELECT *
    FROM Seating AS S
    JOIN [Table] AS T
    On S.TABLE_ID = T.TABLE_ID
```

```
    WHERE T.TABLE_ID < @seat_num
```

```
)
```

```
BEGIN
```

```
    ROLLBACK
```

```
    PRINT 'Cannot be assigned'
```

```
END
```

```
GO
```

Trigger 2:

```
CREATE TRIGGER Max_Guests ON Customer  
FOR INSERT  
AS
```

```
    DECLARE @guest_num INT
```

```
    IF EXISTS (  
        SELECT *  
        FROM Customer AS C  
        JOIN Seating AS S  
        On S.CUST_ID = C.CUST_ID
```

```
        WHERE C.CUST_NUMBER_OF_GUESTS > @guest_num
```

```
    )
```

```
    BEGIN
```

```
        ROLLBACK
```

```
        PRINT 'Cannot be inserted, table does not have enough seats'
```

```
    END
```

```
    GO
```