

Chapter 50: Foreign Keys

Section 50.1: Foreign Keys explained

Foreign Keys constraints ensure data integrity, by enforcing that values in one table must match values in another table.

An example of where a foreign key is required is: In a university, a course must belong to a department. Code for the this scenario is:

```
CREATE TABLE Department (  
  Dept_Code      CHAR (5)      PRIMARY KEY,  
  Dept_Name      VARCHAR (20)  UNIQUE  
);
```

Insert values with the following statement:

```
INSERT INTO Department VALUES ('CS205', 'Computer Science');
```

The following table will contain the information of the subjects offered by the Computer science branch:

```
CREATE TABLE Programming_Courses (  
  Dept_Code      CHAR(5),  
  Prg_Code       CHAR(9) PRIMARY KEY,  
  Prg_Name       VARCHAR (50) UNIQUE,  
  FOREIGN KEY (Dept_Code) References Department(Dept_Code)  
);
```

(The data type of the Foreign Key must match the datatype of the referenced key.)

The Foreign Key constraint on the column Dept_Code allows values only if they already exist in the referenced table, Department. This means that if you try to insert the following values:

```
INSERT INTO Programming_Courses Values ('CS300', 'FDB-DB001', 'Database Systems');
```

the database will raise a Foreign Key violation error, because CS300 does not exist in the Department table. But when you try a key value that exists:

```
INSERT INTO Programming_Courses VALUES ('CS205', 'FDB-DB001', 'Database Systems');  
INSERT INTO Programming_Courses VALUES ('CS205', 'DB2-DB002', 'Database Systems II');
```

then the database allows these values.

A few tips for using Foreign Keys

- A Foreign Key must reference a UNIQUE (or PRIMARY) key in the parent table.
- Entering a NULL value in a Foreign Key column does not raise an error.
- Foreign Key constraints can reference tables within the same database.
- Foreign Key constraints can refer to another column in the same table (self-reference).

Section 50.2: Creating a table with a foreign key

In this example we have an existing table, SuperHeros.

This table contains a primary key ID.

We will add a new table in order to store the powers of each super hero:

```
CREATE TABLE HeroPowers
(
  ID int NOT NULL PRIMARY KEY,
  Name nvarchar(MAX) NOT NULL,
  HeroId int REFERENCES SuperHeros(ID)
)
```

The column HeroId is a **foreign key** to the table SuperHeros.
