Data Integrity (it is just having the right data in your db)

- **Entity Integrity –** There are no duplicate rows in a table. (Duplicate rows is when each value in a row including the PK, equals anther row)
- **Domain Integrity –** Enforces valid entries for a given column by restricting the type, the format, or the range of values. (Domain is just the acceptable and not acceptable values for a column depending on the datatype and its range)
- **Referential integrity –** Rows cannot be deleted, which are used by other records.

/*Performing the DDL(Data definition language) on the school DB tables and adding constraint to them*/

/*1)Domain Integrity*/ **Datatypes**, **Not Null**, **Unique**, **Check**, **Default**

/*2)entity integrity*/ **PK**

/*3)referential integrity*/**FK**

/*Apply on our case*/

/*creat student table*/

/*pk*/

```
use school
create table Student
(Student_ID int primary key,
 Student_Fname nvarchar(10),
 Student_Mname nvarchar(10),
 Student_lname nvarchar(10),
 Student_Fax nvarchar(10) unique,
 Student_Bdate datetime)
```

OR

/*add column to student*/

```
use school2
create table Student
(Student_ID int,
 Student_Fname nvarchar(10),
 Student_Minit nvarchar(10),
 Student_lname nvarchar(10),
 Student_Bdate datetime)
```

```
/*other way*/
```

PK:

```
alter table student
alter column Student_ID int not null


alter table student
add constraint c1 primary key(Student_ID)
```

Unique (name a constraint):

```
use school2
alter table Student
add  Student_Fax nvarchar(10) constraint uq unique
```

OR

/*other way*/

```
use school2
alter table Student
add  Student_Fax nvarchar(10)

alter table student add constraint c2 unique (Student_Fax)
```

/*create the class table*/

/*identity not a constraint*/

```
use school2
alter table student
add st_number int identity(1,1)
```

OR

```
use school2
create table Student
(Student_ID int identity(1,1) primary key,
 Student_Fname nvarchar(10),
 Student_Minit nvarchar(10),
 Student_lname nvarchar(10),
 Student_Bdate datetime)
```

/*create table Class*/

```
create table Class
(Class_Name nvarchar(10) primary key,
 Class_Floor nvarchar(10) default 'unknown')
```

OR

```
alter table class
 add constraint c2 default('unknown') for class_floor
```

/*create teacher table*/

```
create table Teacher
(Teacher_SSN int primary key,
 Teacher_Fname nvarchar(10),
 Teacher_Minit nvarchar(10),
 Teacher_Lname nvarchar(10))
```

Delete actions of rows in the parent table

If you delete one or more rows in the parent table, you can set one of the following actions:

- `ON DELETE NO ACTION`: SQL Server raises an error and rolls back the delete action on the row in the parent table.
- `ON DELETE CASCADE`: SQL Server deletes the rows in the child table that is corresponding to the row deleted from the parent table.
- `ON DELETE SET NULL`: SQL Server sets the rows in the child table to `NULL` if the corresponding rows in the parent table are deleted. To execute this action, the foreign key columns must be **nullable**.
- `ON DELETE SET DEFAULT` SQL Server sets the rows in the child table to their default values if the corresponding rows in the parent table are deleted. To execute this action, **the foreign key columns must have default definitions**. Note that a nullable column has a default value of `NULL` if no default value specified.

By default, SQL Server applies `ON DELETE NO ACTION` if you don't explicitly specify any action.

/*create the teacher/class table*/

```
create table [Teacher Class]
(Teacher_SSN int,
 Class_Name nvarchar(10),
 No_Of_Subjects int,
 constraint c4 check ( No_Of_Subjects between 1 and 3),
 constraint c8 foreign key (teacher_ssn) references teacher(teacher_ssn)on
delete cascade  on update cascade,
 constraint c9 foreign key (class_name) references class(class_name)on
delete cascade  on update cascade,
 constraint c6 primary key (Teacher_SSN,Class_Name))
```

OR

```
create table [Teacher Class]
(Teacher_SSN int foreign key references teacher(teacher_ssn)on delete
cascade  on update cascade,
 Class_Name nvarchar(10) foreign key references class(class_name)on delete
cascade  on update cascade,
 No_Of_Subjects int check ( No_Of_Subjects between 1 and 3),
 constraint c6 primary key (Teacher_SSN,Class_Name))
```

OR

```
alter table "teacher class"
add constraint c4 check( No_Of_Subjects <4 AND No_Of_Subjects >0)
```

/*create teacher telephone table*/

```
create table Teacher_Telephone
(Teacher_SSN int,
 Telephone_Number nvarchar(10),
 constraint c10 foreign key (teacher_ssn) references
teacher(teacher_ssn)on delete cascade  on  update cascade,
 constraint c7 primary key (Teacher_SSN,Telephone_Number)
)

/*Refrential*/
use school
alter table class
add Teacher_ssn int

use school
alter table class
add constraint FK_class foreign key (Teacher_ssn) references Teacher(Teacher_SSN)
```
