SQL: Structured Query language

**SQL key (Super key, Candidate key, Primary key, unique key, Composite key, foreign key)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SocialSecurityNumber** | **Name** | **Address** | **BirthDay** | **TaxIDNumber** |
| 818764259 | John | 22 washington st | 1/3/1972 | 837650987 |
| 517881756 | Raj | 1 boston road | 6/9/1964 | 767354174 |
| 788876566 | Janet | 23 avenue NYC | 4/15/1985 | 939635252 |
| 748209944 | Janet | 23 avenue NYC | 4/15/1980 | 643228844 |

**Uniquely identifying a row means being able to pin point to one row.** For example suppose I need you to delete the fourth row, if I tell you,  delete the row where the name is Janet. There are two such rows, so that means I cannot uniquely identify a row (or pin point a row) with the name alone.

But if I tell you delete the row where Name is Janet , address is 23 avenue NYC and birth day is 4/15/1980. Then you will delete the correct row. That means a combination of Name,Address and Birthday, uniquely identifies a row.  
(You can also uniquely identify a row with SocialSecurityNumber or TaxIDNumber )

**Super  key:  A combination of columns that can uniquely identify a row in a table. A super key can have more columns than it needs to uniquely identify a row.**

 For example in the above table [SocialSecurityNumber+ Name+Birthday] is a super key. Note here that SocialSecurityNumber by itself can uniquely identify a row. Name and Birthday are extra  columns that are not required to uniquely identify a row.

**Candidate  key:  A minimal super key.   (A sub set of super key, where if you remove even one column, you cannot identify a row uniquely any more)**

Assume that a particular address you **cannot** have two individuals with the same name AND date of birth.

In the example above SocialSecurityNumber,  TaxIDNumber and the combination [Name+Addresss+ Birthday] are all Candidate keys.  But if you remove Birthday from [Name+Addresss+ Birthday] then it is not a candidate key anymore.

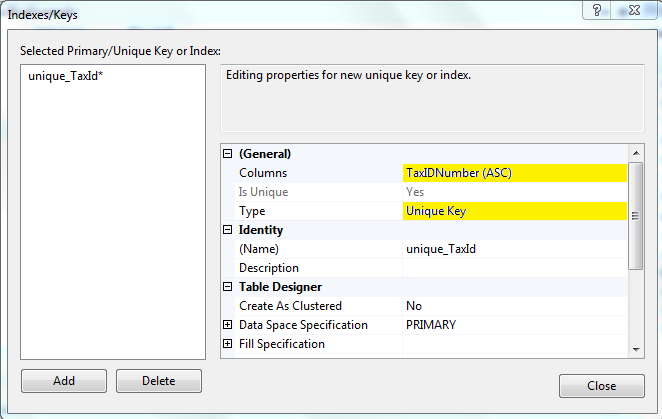
**Primary key:  The main candidate key in a table. (The main column or combination of columns that uniquely identifies a row).** In the above example,  SocialSecurityNumber is the primary key.

Note that when you set a column as a primary key in SQL server, it also automatically sets that column (or the combination of columns) as a “clustered index”. But if you want, you can remove the clustered index on the primary key. It is not mandatory to have the primary key as the clustered index.

**Unique Key: Unique is practically (but not theoretically) the same as superkey. Unique key is a constraint that you put on a table where you want to ensure that the value is not repeated between two rows.**

For example, in my SQL database table I want to make sure that no two rows have the same TaxIDNumber. To do that in SQL Server, I will right click on the table-->Design

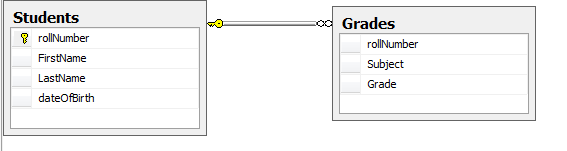
In the new window that opens up, I will right click the table-->Indexes/Keys. Click Add. Then make the selections as shown in the image below.

[](http://2.bp.blogspot.com/-L1dHyKFOAhw/TxtD45NGXyI/AAAAAAAAAMk/N1VI-P4aBWs/s1600/UniqueKey.PNG)

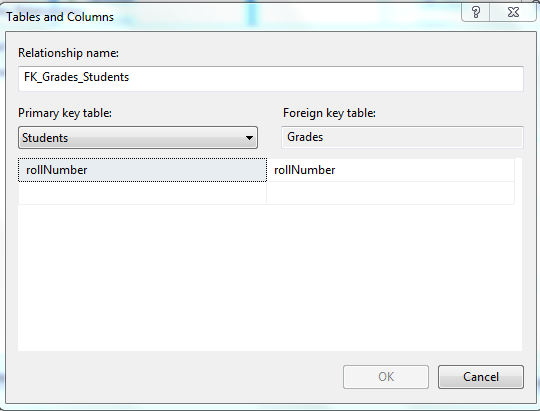
Note here that you can set a unique key on a combination of columns. You can also set multiple unique keys in a table.   
  
  
**Difference between primary and unique key:**

|  |  |
| --- | --- |
| **Primary key constraint** | **Unique key constraint** |
| A table can have only one primary key. | A table can have multiple unique keys. |
| Doesn’t allow null in any of the columns that are a part of the primary key. | Allows nulls in all the columns that are a part of the unique key constraint. |
| By default a clustered index is created on the column(s) that you set the primary key on.  (This index can be dropped later, it is not mandatory) | By default a non clustered index is created on the column(s) that you set the Unique  key on.  (This index can be dropped later, it is not mandatory) |

**Composite key (Compound key): A key that includes more than one column.**  
  
**Foreign key constraint:**This is more easily explained with an example. Consider the tables shown below.

[](http://4.bp.blogspot.com/-BGXuPTmjjgs/TxyLCxhZExI/AAAAAAAAAMs/w0z6ceARjYY/s1600/foreignKey.PNG)

The table Students hold the roll numbers for each student. The grade table holds a grade for each roll number.  To make sure  a grade for a roll number should not exist in the Grade table, if that roll number is not present in the students table, we will set a foreign key as shown below. With a foreign key set, sql server will not allow the entry of a roll number in the Grades table, unless that roll number exists in the Students table.

[](http://4.bp.blogspot.com/-i9qe81fboxY/TxyL-Q6WvBI/AAAAAAAAAM0/_oZ_tS_tFNE/s1600/foreignKeys.PNG)

Note here that rollNumber has to be defined as a primary or uniqe key in the students table for this foreign key to be possible.