Quote:

|  |
| --- |
| For example, Given an employee table consisting of the columns:  ***employeeID name job and  departmentID*** we could use the employeeID in combination with any or all other columns of this table to uniquely identify a row in the table. **Examples of superkeys** in this table would be **{employeeID, Name}, {employeeID, Name, job}, and {employeeID, Name, job, departmentID}.** In a real database we don't need values for all of those columns to identify a row. We only need, per our example, the set {employeeID}. This is a minimal superkey – that is, a minimal set of columns that can be used to identify a single row. **So, employeeID is a candidate key.** |

Now, if employeeID is a candidate key then why not it is the superkey. Because employeeID can also uniquely identify the tuples.  
  
(2) In your example why Roll number is not the superkey as it is uniquely identifying the tuples?  
  
(3)

Quote:

|  |
| --- |
| Any Unique key with some Non Unique key combination is called a super key of the relationship |

Is it neccessary that Unique key has to be combined with some Non Unique key to be called as a super key.

In this part, I will be briefing about different kind of keys available in database systems.  
Base example I will be using the following table to explain about database keys -  
  
EMPLOYEE [ EMPLOYEE\_ID, EMPLOYEE\_SSN\_ID, EMPLOYEE\_DEPT\_ID, EMPLOYEE\_FIRST\_NAME, EMPLOYEE\_LAST\_NAME, EMPLOYEE\_ADDRESS ]  
  
**Candidate Key**  
  
A candidate key is a combination of attributes that can be uniquely used to identify a database record without any extraneous data. Each table may have one or more candidate keys. In general, one of these candidate keys is selected as the table primary key.  
  
Example - From the above table EMPLOYEE\_ID, EMPLOYEE\_SSN\_ID, and EMPLOYEE\_DEPT\_ID can be considered as candidate keys  
  
**Primary Key**  
  
A primary key is a single column or combination of columns that uniquely defines a record. None of the columns that are part of the primary key can contain a null value. A table can have only one primary key.  
  
Example - EMPLOYEE\_ID or EMPLOYEE\_SSN\_ID can be considered as primary keys  
  
**Unique Key**  
  
A unique key or primary key [is a candidate key] to uniquely identify each row in a table. It be comprised of either a single column or multiple columns.  
  
The major difference is that for unique keys the implicit NOT NULL constraint is not automatically enforced, while for primary keys it is enforced. Thus, the values in unique key columns may or may not be NULL.  
  
**Differences between Primary Key and Unique Key**  
  
Primary Keys -  
1. It will not accept null values.         
2. There will be only one primary key in a table.         
3. Clustered index is created in Primary key.         
4. Primary key allows each row in a table to be uniquely identified and ensures that no duplicate rows exist.         
  
Unique Keys -  
1. Null values are accepted.  
2. More than one unique key will be there in a table.  
3. Non-Clustered index is created in unique key.  
4. Unique key constraint is used to prevent the duplication of key values within the rows of a table and allow null values.  
  
**Alternate Key**  
  
A candidate key that is not the primary key is called an alternate key.  
  
Example - If EMPLOYEE\_ID is considered as primary keys then EMPLOYEE\_SSN\_ID is an alternate key.  
  
**Superkey**  
  
A superkey is a combination of attributes that can be uniquely used to identify a database record. A table might have many superkeys. Candidate keys are a special subset of superkeys that do not have any extraneous information in them.  
  
A primary key is therefore a minimum superkey.  
  
Examples - Any combination of the following can be considered as a Super key  
  
- EMPLOYEE\_ID - Minimal Super Key  
  
- EMPLOYEE\_ID and EMPLOYEE\_SSN\_ID  
  
- EMPLOYEE\_ID, EMPLOYEE\_SSN\_ID and EMPLOYEE\_DEPT\_ID  
  
- EMPLOYEE\_ID, EMPLOYEE\_SSN\_ID, EMPLOYEE\_DEPT\_ID, EMPLOYEE\_FIRST\_NAME  
  
- EMPLOYEE\_SSN\_ID, EMPLOYEE\_FIRST\_NAME, EMPLOYEE\_LAST\_NAME  
  
**Foreign Key**  
  
The foreign key identifies a column or a set of columns in one (referencing) table that refers to a column or set of columns in another (referenced) table.  
  
**Composite Key**  
  
A primary key that made up of more than one attribute is known as a composite key.  
  
Example - [ EMPLOYEE\_ID and EMPLOYEE\_SSN\_ID ] can together be treated as (one of) composite keys. Another combination can be [ EMPLOYEE\_ID, EMPLOYEE\_SSN\_ID and EMPLOYEE\_DEPT\_ID ]