

DBMS Keys  
Mahmoud El-Haj  
13/01/2020

The following is to help you understand the DBMS Keys mentioned in the 2nd lecture (2. Relational Model)

Why do we need keys:

- Keys are the essential elements of any relational database.
- Keys are used to identify tuples in a relation R.
- Keys are also used to establish the relationship among the tables in a schema.

Type of keys discussed below: **Superkey, Candidate Key, Primary Key** (for Foreign Key please refer to the lecture slides (2. Relational Model)).

- **Superkey (SK)** of a relation R:
  - Is a set of attributes SK of R with the following condition:
    - No two tuples in any valid relation state  $r(R)$  will have the same value for SK
      - That is, for any distinct tuples  $t_1$  and  $t_2$  in  $r(R)$ ,  $t_1[SK] \neq t_2[SK]$
  - Every relation has at least one default superkey: the set of all its attributes
  - Basically superkey is nothing but a key. It is a super set of keys where all possible keys are included (see example below).
  - An attribute or a set of attributes that can be used to identify a tuple (row) of data in a Relation (table) is a Superkey.
- **Candidate Key** of R (all superkeys that can be candidate keys):
  - A "minimal" superkey
  - That is, a (candidate) key is a superkey K such that removal of any attribute from K results in a set of attributes that IS NOT a superkey (does not possess the superkey uniqueness property) (see example below).
  - A Candidate Key is a Superkey but not necessarily vice versa
  - Candidate Key: Are keys which can be a primary key.
  - If a relation schema has more than one key, each of the keys is called a candidate key.
  - Candidate key is the minimal subset of a superkey.
  - A candidate key is one of a number of keys, which are superkeys having no redundant attributes.
  - If any proper subset of a superkey is a superkey then that key cannot be a candidate key.

So to explain, any key is a superkey and any superkey where a subset of that superkey IS NOT another superkey can become a candidate key (therefore a candidate key is a minimal superkey).

- **Primary Key (PK):**
  - One of the candidate keys of a relation is chosen as the primary key – This is the candidate key whose values are used to identify tuples in a relation.
  - Primary key attributes are underlined.
  - Primary key cannot contain NULL value.
- You identify Superkeys then candidate keys (minimal superkeys) to select your primary key.

The following table helps you understand the difference between a superkey and a candidate key:

BASIS FOR COMPARISON	SUPERKEY	CANDIDATE KEY
Basic	A single attribute or a set of attributes that uniquely identifies all attributes in a relation is Superkey.	A proper subset of a Superkey, which is also a Superkey is a candidate key (remember if a subset of the candidate key is a Superkey then it is not a candidate key).
One in other	It is not compulsory that all Superkeys will be candidate keys.	All candidate keys are Superkeys.
Selection	The set of Superkeys forms the base for selection of candidate keys.	The set of candidate keys form the base for selection of a single primary key.
Count	There are comparatively more Superkeys in a relation.	There are comparatively less candidate keys in a relation.

----- Scroll to next page for an example -----

### Example:

Students

StudentID	REG_ID	Name	Major	Email
1	CS-2019-55	Amy	Computer Science	amyOne@xyz.com
2	CS-2018-12	Salma	Computer Science	salma@xyz.com
3	Phy-2019-65	Amy	Physics	amyTwo@xyz.com
4	Eng-2019-23	Angelina	Engineering	angelina@xyz.com

Identifying Keys: you start by identifying all keys that can be superkeys, then find which can be candidate keys, then later select a primary key from the candidate keys.

Keys:

**Superkeys** (each of the following keys can be used to identify a tuple in the relation Students):

- StudentID
- REG\_ID
- Email
- StudentID + REG\_ID
- REG\_ID + Email
- Email + StudentID
- StudentID + REG\_ID + Email

**Candidate Keys** (some of the superkeys identified above can become candidate keys and some cannot):

For example **REG\_ID + Email + StudentID** cannot be a candidate key as **StudentID + Email** (which is a subset) is a Superkey.

Similarly: **StudentID + REG\_ID** cannot be a candidate key as **StudentID** and **REG\_ID** can each act as a superkey individually.

Therefore only the following can be candidate keys:

- StudentID
- REG\_ID
- Email

### Primary Key:

Any of the candidate keys chosen above can become a primary key. We need to think of what would make more sense when searching for students. Should we use StudentID, REG\_ID or the student's email?

We think that the **REG\_ID** makes more sense to act as a primary key. It of course depends on the design and description of the Students Relation.