

# DBMS KEYS

## **Keys in DBMS.**

- **DBMS has Five types of Keys in it and they all has different functionality.**
- **The Keys are as follows:**
  - 1. Super Key**
  - 2. Candidate Key**
  - 3. Primary Key**
  - 4. Foreign Key**
  - 5. Composite Key**

keys.

# SUPER KEY

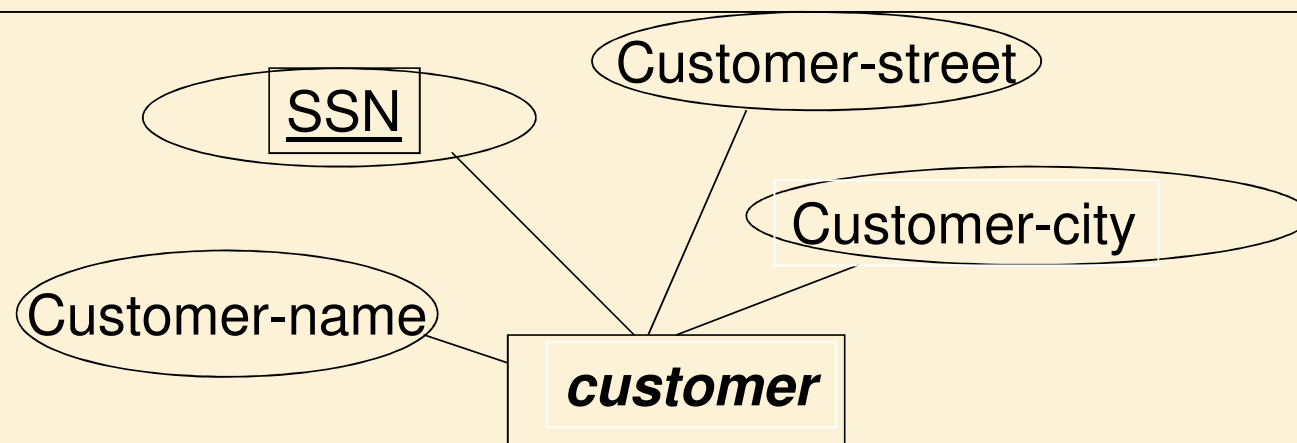
- *Super Key* is a set of attributes whose set of values can uniquely identify an entity instance in the entity set.
- A *Super Key* can contains one or more than one attributes.
- *Super Key* is the broadest definition of unique identifiers of an entity in an entity set.

## SUPER KEY

The combination of “SSN” and “Name” is a super key of the following entity set ***customer***.

Because:

The value of attributes “SSN” and “Name”, such as 558-36-1234 and Smith, can uniquely identify that particular customer in ***customer*** entity set, which is the pool of all customers.



**Book (BookId, BookName, Author)**

**So in this table we can have**

- ⊃ (BookId)**
- ⊃ (BookId, BookName)**
- ⊃ (BookId, BookName, Author)**
- ⊃ (BookId, Author)**
- ⊃ (BookName, Author)**

Emp_SSN	Emp_Number	Emp_Name
123456789	226	Steve
999999321	227	Ajeet
888997212	228	Chaitanya
777778888	229	Robert

**Super keys:**

{Emp\_SSN}

{Emp\_Number}

{Emp\_SSN, Emp\_Number}

{Emp\_SSN, Emp\_Name}

{Emp\_SSN, Emp\_Number, Emp\_Name}

{Emp\_Number, Emp\_Name}

- ❖ We are unsurprisingly very interested in the most economical combination(s) of attributes that can uniquely identify any particular entity.
- ❖ Therefore, we introduce ***Candidate Key*** next.

# CANDIDATE KEY

- *Candidate key* is a set of one or more attributes whose set of values can uniquely identify an entity instance in the entity set.
- Any attribute in the candidate key cannot be omitted without destroying the uniqueness property of the *Candidate key*.
- It is minimal *Super Key*.

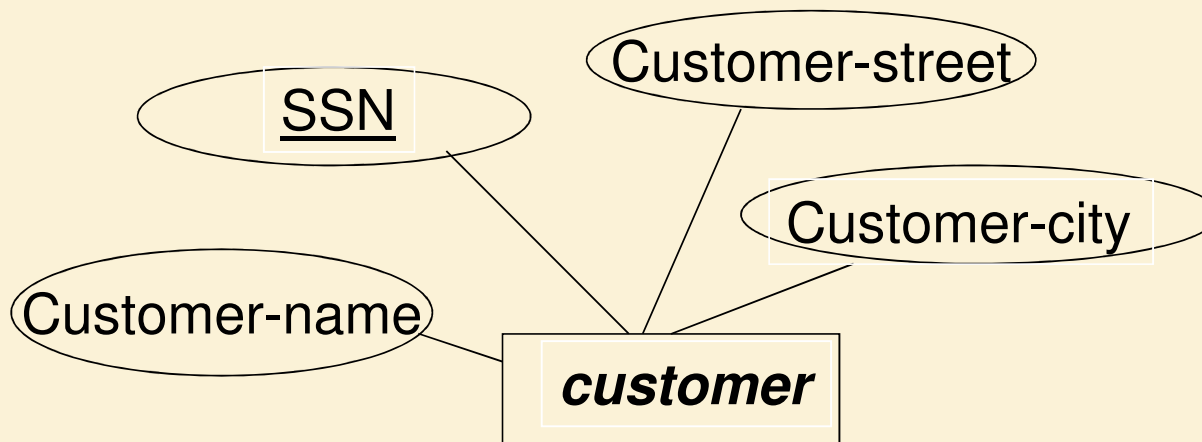


## Properties of Candidate Key.....

- While most entity sets have only one candidate key, some entity sets could have more than one candidate key.
- Candidate key could have more than one attributes.
- In building a database in a database software, the software will only allow to use one candidate key to be the unique identifier of an entity for an entity set.

## CANDIDATE KEY

- Example:
- (*SSN*, *Name*) is NOT a candidate key, because taking out “*name*” still leaves “*SSN*” which can uniquely identify an entity. “*SSN*” is a candidate key of ***customer***.
- **Example:** Both “*SSN*” and “*License #*” are candidate keys of ***Driver*** entity set.



**Super keys:**

{Emp\_SSN}  
{Emp\_Number}  
{Emp\_SSN, Emp\_Number}  
{Emp\_SSN, Emp\_Name}  
{Emp\_SSN, Emp\_Number,  
Emp\_Name}  
{Emp\_Number, Emp\_Name}

**Candidate Keys:**

As I stated above, they are the minimal super keys with no redundant attributes.

{Emp\_SSN}  
{Emp\_Number}

- Overall, ***Super Key*** is the broadest unique identifier; ***Candidate Key*** is a subset of ***Super Key***; and ***Primary Key*** is a subset of ***Candidate Key***.
- In practice, we would first look for Super Keys. Then we look for ***Candidate Keys*** based on experience and common sense.
- If there is only one ***Candidate Key***, it naturally will be designated as the ***Primary Key***.
- If we find more than one ***Candidate Key***, then we can designate any one of them as ***Primary Key***.

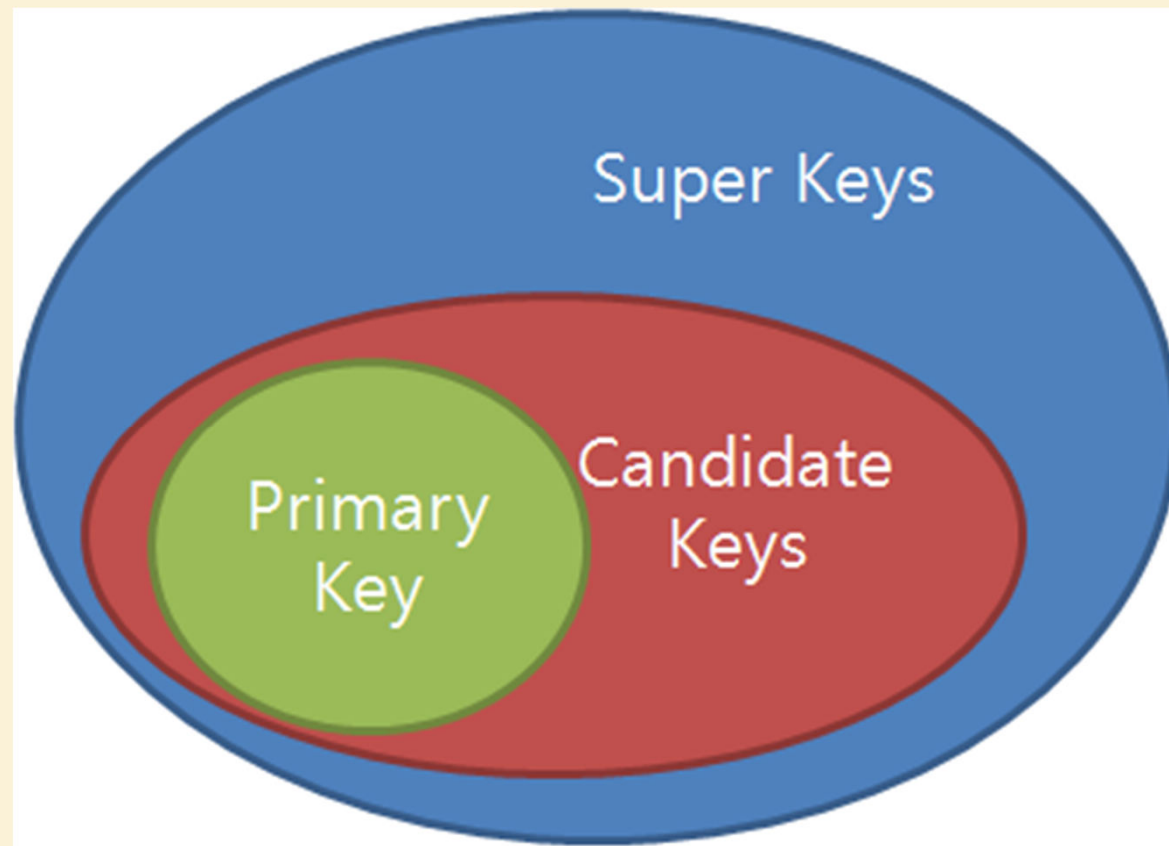
# PRIMARY KEY

- The *Primary Key* is an attribute or a set of attributes **that uniquely** identify a specific instance of an entity.
- Every entity in the data model must have a primary key whose values uniquely identify instances of the entity.

# Properties Of Primary Keys....

**To qualify as a primary key for an entity, an attribute must have the following properties:**

- It must have a non-null value for each instance of the entity
- The value must be unique for each instance of an entity
- The values must not change or become null during the life of each entity instance



❖ *Primary* and *Foreign* keys are the most basic components on which *relational* theory is based. Each entity must have a attribute or attributes, the primary key, whose values uniquely identify each instance of the entity. Every child entity must have an attribute, the foreign key, that completes the association with the parent entity.



# FOREIGN KEY

- A *Foreign key* is an attribute that completes a relationship by identifying the parent entity.
- *Foreign keys* provide a method for maintaining integrity in the data (called referential integrity) and for navigating between different instances of an entity.
- Every relationship in the model must be supported by a foreign key.

## Properties Of Foreign Key....

- Every dependent and category (subtype) entity in the model must have a foreign key for each relationship in which it participates.
- Foreign keys are formed in dependent and subtype entities by migrating the entire primary key from the parent or generic entity. If the primary key is composite, it may not be split.

<u>studentId</u>	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042

Foreign Keys

Relationship

Primary Keys

<u>courseId</u>	courseName
A004	Accounts
C002	Computing
P301	History
S042	Short Course

# COMPOSITE KEY

- When a primary key is created from a combination of 2 or more columns, the primary key is called a composite key.
- Each column may not be unique by itself within the database table but when combined with the other column(s) in the composite key, the combination is unique.

# ALTERNATE KEYS

Primary  
key

Alternate Keys

Roll_No	Name	Branch	City
01	Deepak	Computers	Bhiwani
02	Mukesh	Electronics	Rohtak
03	Teena	Mechanical	Bhiwani
04	Deepti	Chemical	Rohtak
05	Monika	Civil	Delhi

For the following table Identify super key, primary key, candidate keys

[illegible]