# SDF II(15B11CI211)

EVEN Semester 2021


## 2nd Semester , First Year

## Jaypee Institute Of Information Technology (JIIT), Noida

# Topics Covered

- Keys
- Types of Keys
  - Super Key
  - Candidate Key
  - Primary Key
  - Foreign Key
  - Composite Key

# Keys

- Keys in DBMS is an attribute or set of attributes which helps you to identify any record or row(tuple) from a relation(table). They allow you to find the relation between two tables.
- **For example:** In STUDENT table, ID is used as a key because it is unique for each student. In PERSON table, passport_number, license_number, SSN are keys since they are unique for each person.

| STUDENT |
| --- |
| ID |
| Name |
| Address |
| Course |

| PERSON |
| --- |
| Name |
| DOB |
| Passport_Number |
| Licence_Number |
| SSN |

# Types of Keys

- DBMS has different types of Keys in it and they all has different functionality.

The Keys are as follows:

- Super Key
- Candidate Key
- Primary Key
- Foreign Key
- Composite Key

# Super Key

- **Super Key** is defined as a set of attributes within a table that can uniquely identify each record within a table.
- In the table defined, super key would include **student_id, (student_id, name), phone etc.**

## Confused?

- The first one is pretty simple as student_id is unique for every row of data, hence it can be used to identity each row uniquely.
- Next comes, (student_id, name), now name of two students can be same, but their student_id can't be same hence this combination can also be a key.
- Similarly, phone number for every student will be unique, hence again, phone can also be a key.

  So they all are super keys.

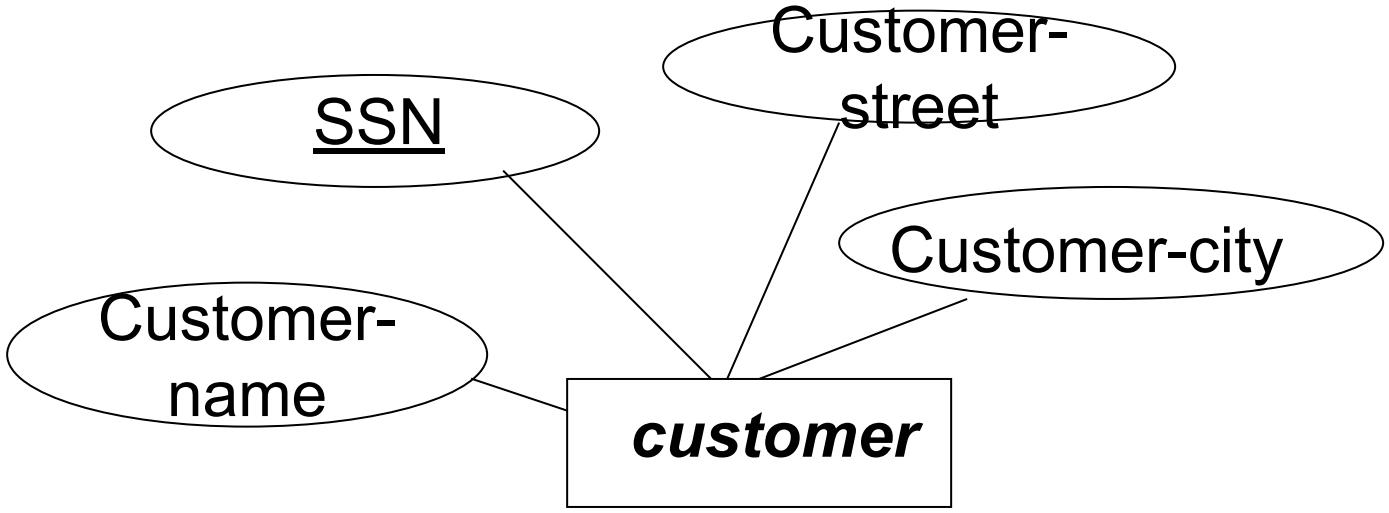
| student_id | name | phone | age |
|---|---|---|---|
| 1 | Akon | 9876723452 | 17 |
| 2 | Akon | 9991165674 | 19 |
| 3 | Bkon | 7898756543 | 18 |
| 4 | Ckon | 8987867898 | 19 |
| 5 | Dkon | 9990080080 | 17 |

# 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 can uniquely identify that particular customer in **customer** entity set, which is the pool of all customers.



# Candidate Key

- A **super key** with no redundant attribute is known as candidate key. Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is that the candidate key should not have any redundant attributes. That's the reason they are also termed as **minimal super key**.
- A candidate key can never be NULL or empty. And its value should be unique.
- There can be more than one candidate keys for a table.
- A candidate key can be a combination of more than one columns(attributes).

# Candidate Key

- Lets take an example of table “Employee”.
- This table has three attributes: Emp_Id, Emp_Number & Emp_Name.
- Here Emp_Id & Emp_Number will be having unique values and Emp_Name have duplicate values as more than one employees have same name.
- ***How many super keys the table can have?***

| Emp_Id | Emp_Number | Emp_Name |
|---|---|---|
| E1 | JIIT1604 | Rohan |
| E2 | JIIT1605 | Mohan |
| E3 | JIIT1606 | Sohan |
| E4 | JIIT1607 | Mohan |

# Candidate Key

## Super keys

1. {Emp_Id}
2. {Emp_Number}
3. {Emp_Id, Emp_Number}
4. {Emp_Id, Emp_Name}
5. {Emp_Id, Emp_Number, Emp_Name}
6. {Emp_Number, Emp_Name}

## Lets select the candidate keys from the above set of super keys.

1. {Emp_Id} – No redundant attributes
2. {Emp_Number} – No redundant attributes
3. {Emp_Id, Emp_Number} – Redundant attribute. Either of those attributes can be a minimal super key as both of these columns have unique values.
4. {Emp_Id, Emp_Name} – Redundant attribute Emp_Name.
5. {Emp_Id, Emp_Number, Emp_Name} – Redundant attributes. Emp_Id or Emp_Number alone are sufficient enough to uniquely identify a row of Employee table.
6. {Emp_Number, Emp_Name} – Redundant attribute Emp_Name.

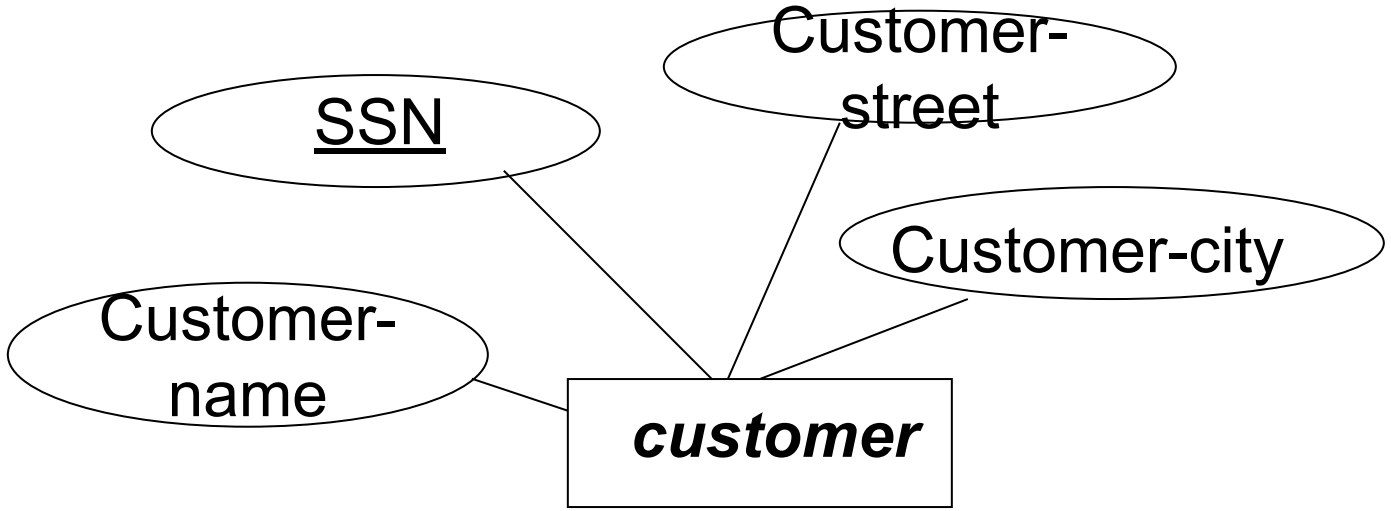
## The **candidate keys** we have selected are:

{Emp_Id}
{Emp_Number}

# Candidate key

Another 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 table**



# Primary Key

- Primary key is a candidate key that is most appropriate to become the main key for any table.
- It is a key that can uniquely identify each record in a table.
- We denote it by underlining the attribute name (column name).

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 Key

In the following table, there are three attributes: Stu_ID, Stu_Name & Stu_Age. Out of these three attributes, one attribute or a set of more than one attributes can be a primary key.

-----Attribute Stu_Name alone cannot be a primary key as more than one students can have same name, Attribute Stu_Age alone cannot be a primary key as more than one students can have same age, Attribute **Stu_Id** alone is a primary key as each student has a unique id that can identify the student record in the table.
-----{Stu_Id, Stu_Name} collectively can identify the tuple in the, but we do not choose it as primary key because Stu_Id alone is enough to uniquely identifies rows in a table and we always go for minimal set.

**Table Name: STUDENT**

| Stu_Id | Stu_Name | Stu_Age |
|---|---|---|
| 101 | Steve | 23 |
| 102 | John | 24 |
| 103 | Robert | 28 |
| 104 | Steve | 29 |
| 105 | Carl | 29 |

# Primary Key

- Primary keys are not necessarily to be a single attribute (column). It can be a set of more than one attributes. Consider this table ORDER, this table keeps the daily record of the purchases made by the customer. This table has three attributes: Customer_ID, Product_ID & Order_Quantity.
- Customer_ID alone cannot be a primary key as a single customer can place more than one order thus more than one rows of same Customer_ID value.
- Product_ID alone cannot be a primary key as more than one customers can place a order for the same product thus more than one rows with same product id.
- Order_Quantity alone cannot be a primary key as more than one customers can place the order for the same quantity.
- Since none of the attributes alone were able to become a primary key, but {Customer_ID, Product_ID} together can identify the rows uniquely in the table so this set is the primary key for this table.

| Customer_ID | Product_ID | Order_Quantity |
|---|---|---|
| 1011 | 9023 | 10 |
| 1122 | 9023 | 15 |
| 1099 | 9031 | 20 |
| 1177 | 9031 | 18 |
| 1011 | 9111 | 50 |

- 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**.
- 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**.

# Foreign Key

- Foreign Key is used to generate the relationship between the tables. It is a field in database table that acts as a Primary Key in another table.
- The **table** with the **foreign key** is called the child **table**, and the **table** with the primary **key** is called the **referenced** or parent **table**.
- Data should be entered in the foreign key column with great care, as wrongly entered data can invalidate the relationship between the two tables.

# Foreign Key

The relation which is being referenced is called referenced relation and the corresponding attribute is called referenced attribute and the relation which refers to the referenced relation is called referencing relation and the corresponding attribute is called referencing attribute.

The referenced attribute of the referenced relation should be the primary key for it. For Example, **STUD_NO** in STUDENT_COURSE is a foreign key to STUD_NO in STUDENT relation. Here **STUD_NO** is a referenced attribute in STUDENT table.

**STUDENT**

| STUD_NO | STUD_NAME | STUD_PHONE | STUD_STATE | STUD_COUNTRY | STUD_AGE |
|---|---|---|---|---|---|
| 1 | RAM | 9716271721 | Haryana | India | 20 |
| 2 | RAM | 9898291281 | Punjab | India | 19 |
| 3 | SUJIT | 7898291981 | Rajsthan | India | 18 |
| 4 | SURESH | | Punjab | India | 21 |

**Table 1**

**STUDENT_COURSE**

| STUD_NO | COURSE_NO | COURSE_NAME |
|---|---|---|
| 1 | C1 | DBMS |
| 2 | C2 | Computer Networks |
| 1 | C2 | Computer Networks |

**Table 2**

Foreign Key can be NULL as well as may contain duplicate tuples. For Example, STUD_NO in STUDENT_COURSE relation is not unique. However, the STUD_NO in STUDENT relation is a primary key and it needs to be always unique and it cannot be null.

# Composite Key

- If any single attribute of a table is not capable of being the key i.e it cannot identify a row uniquely, then we combine two or more attributes to form a key. This is known as a composite key.

- Any key such as super key, primary key, candidate key etc. can be called composite key if it has more than one attributes.

# Composite Key

Lets consider a table Sales. This table has four columns (attributes) – cust_Id, order_Id, product_code & product_count. None of these columns **alone** can play a role of key in this table. So, the key should be having more than one attributes. Key here is **{cust_id, product_code}.**

This is a composite key as it is made up of more than one attributes.

**Table – Sales**

| cust_Id | order_Id | product_code | product_count |
|---|---|---|---|
| C01 | O001 | P007 | 23 |
| C02 | O123 | P007 | 19 |
| C02 | O123 | P230 | 82 |
| C01 | O001 | P890 | 42 |

# References


- https://beginnersbook.com/
- https://www.studytonight.com/
- https://www.tutorialspoint.com/
- https://www.c-sharpcorner.com/
- https://www.javatpoint.com/
- https://www.w3schools.com/
- https://www.guru99.com/

# Thank you