DBMS Keys

Why should we learn about Keys ?

❑ Keys play an important role in the relational database.

❑ It is used to **uniquely identify** any record or row of data from the table.

❑ It is also used **to establish** and **identify** relationships **across** multiple tables.

Examples of Keys

STUDENT

ID

Student_Name

Course_Enrolled_In

Roll_Number

E-mail_Address

Phone_Number

Hall_Attached_To

PERSON

Name

DOB

Father’s_Name

E-mail_Address

Phone_Number

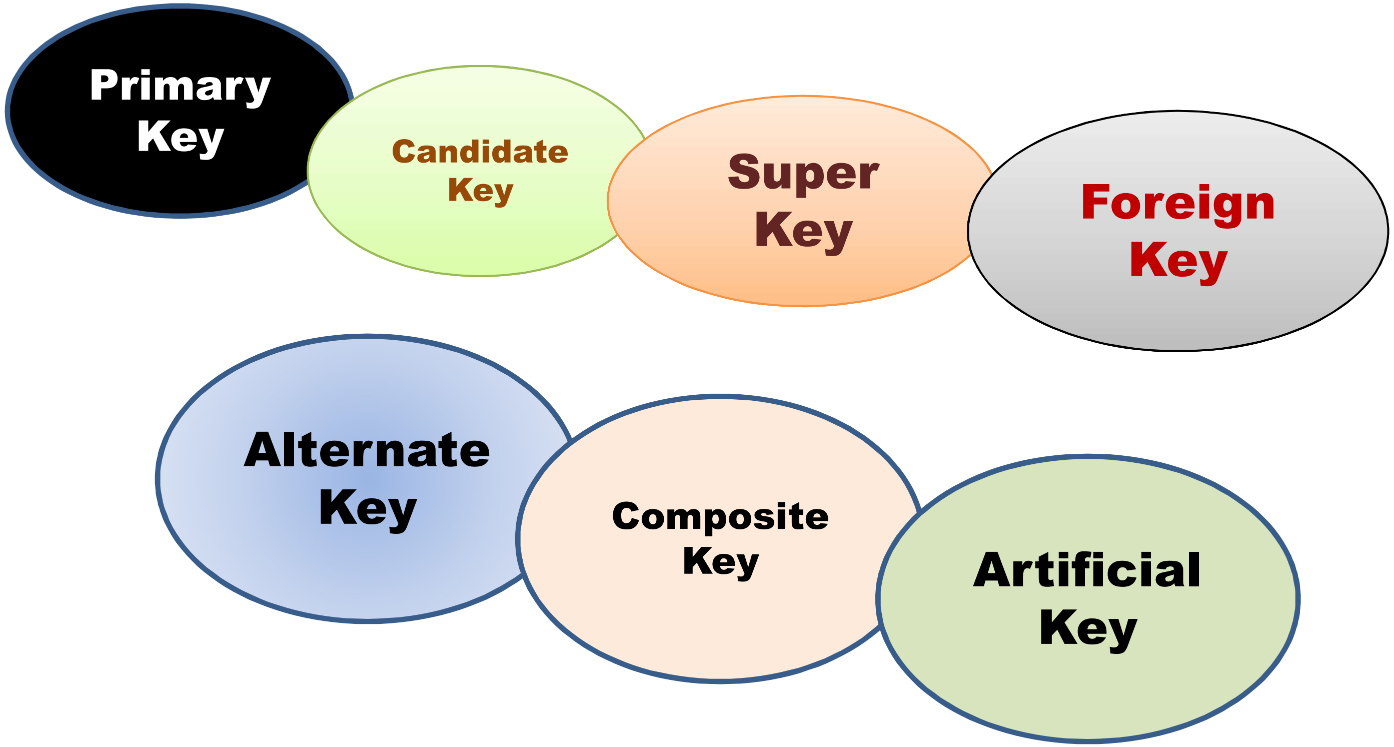
Driving_License_Number

Passport_Number

Passport_of_Country

SSN

Types of Keys



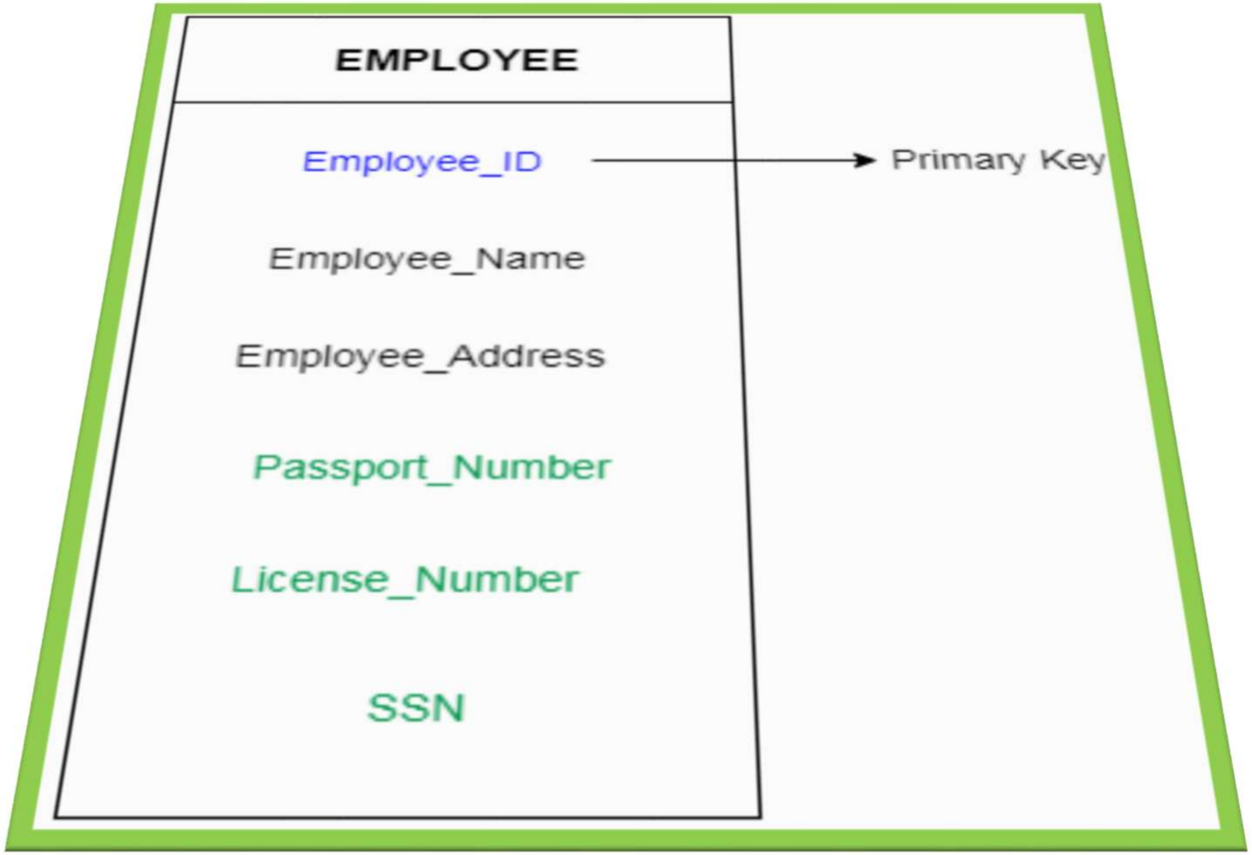
Primary Key

❑ It is the first key used to identify one and only one instance of an entity uniquely. An entity can contain multiple keys, as we saw in the PERSON table. The key which is most suitable from those lists becomes a primary key.

❑ In the EMPLOYEE table, ID can be the primary key since it is unique for each employee. In the EMPLOYEE table, we can even select License_Number and Passport_Number as primary keys since they are also unique.

❑ For each entity, the primary key selection is based on requirements and developers.

Primary Key (contd..)



EMPLOYEE

Employee_ID

Employee_Name

Employee_Address

Passport_Number

License_Number

SSN

Primary Key

Candidate Key

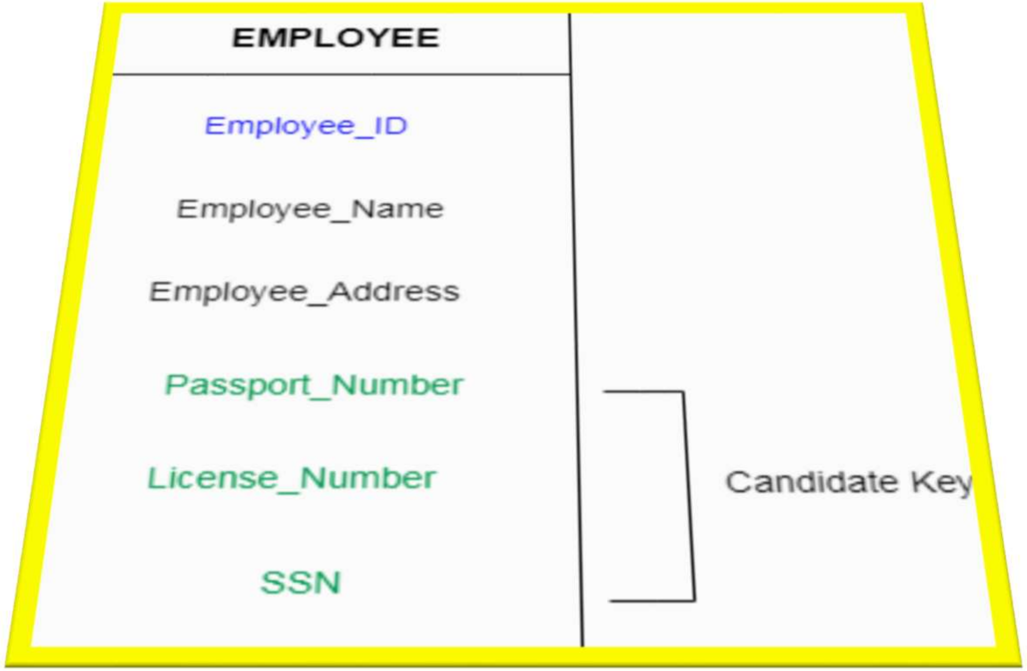
- A candidate key is an attribute or set of attributes that can uniquely identify a tuple.

- Except for the primary key, the remaining attributes are considered a candidate key.

- The candidate keys are as strong as the primary key.

Candidate Key (contd..)

- Here Employee_ID is best suited for the primary key. The rest of the attributes, like SSN, Passport_Number, License_Number, etc., are considered a candidate key.


EMPLOYEE

Employee_ID

Employee_Name

Employee_Address

Passport_Number

License_Number

SSN

Candidate Key

Super Key

- ❑ Super key is **an attribute set** that can uniquely identify a tuple.

- ❑ A super key is a superset of a candidate key.

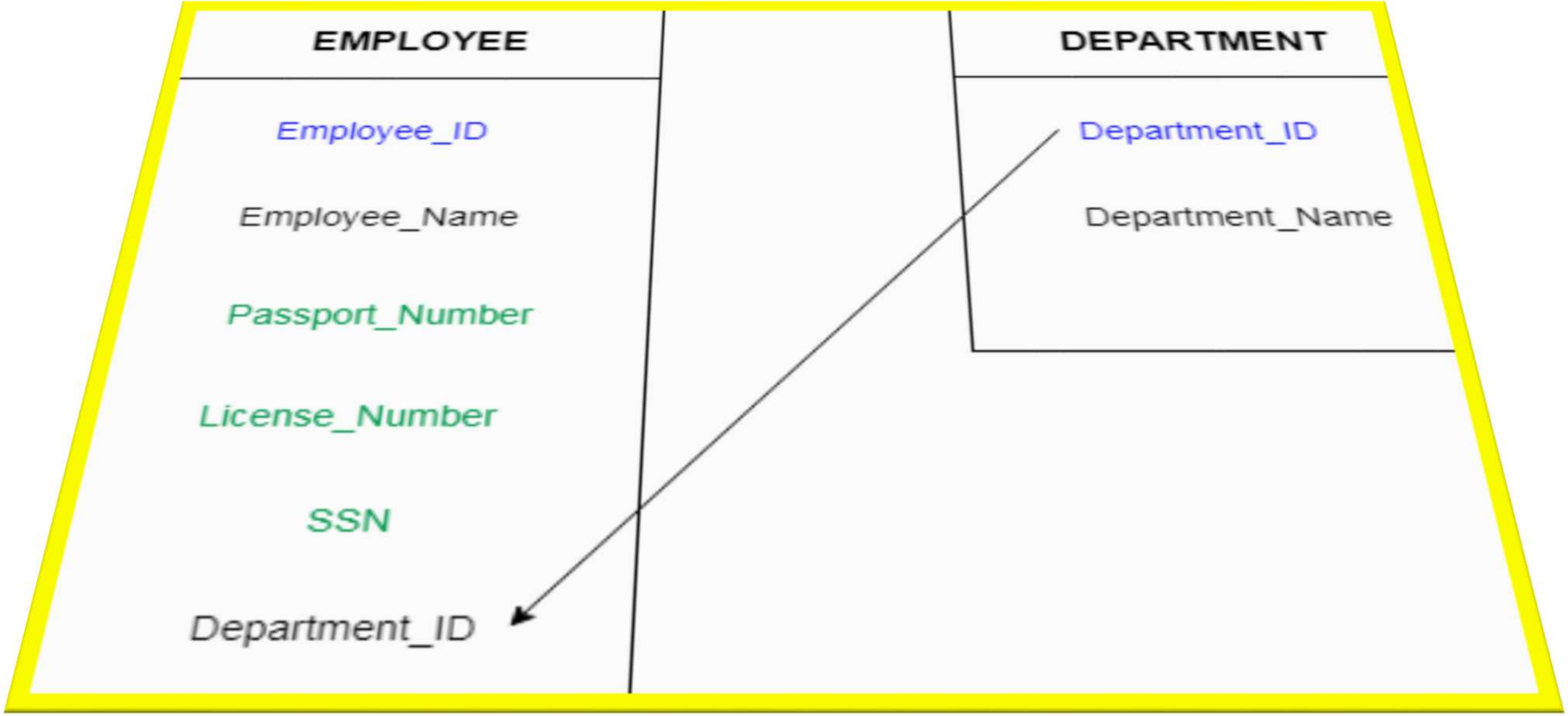
- ❑ **For Example**: In the preceding slide(s) EMPLOYEE table, for(EMPLOEE_ID, EMPLOYEE_NAME), the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key.

- ❑ The super key would be EMPLOYEE-ID (**EMPLOYEE_ID**, **EMPLOYEE-NAME**), etc.

Foreign Key

- ❑ Foreign keys are the column of the table used to point to the primary key of another table.
- ❑ Every employee works in a specific department in a company, and employee and department are two different entities. So we can't store the department's information in the employee table. That's why we link these two tables through the primary key of one table.
- ❑ We add the primary key of the DEPARTMENT table, Department_Id, as a new attribute in the EMPLOYEE table.
- ❑ In the EMPLOYEE table, Department_Id is the foreign key, and both the tables are related.

Foreign Key (contd..)



EMPLOYEE

Employee_ID

Employee_Name

Passport_Number

License_Number

SSN

Department_ID

DEPARTMENT

Department_ID

Department_Name

Alternate Key

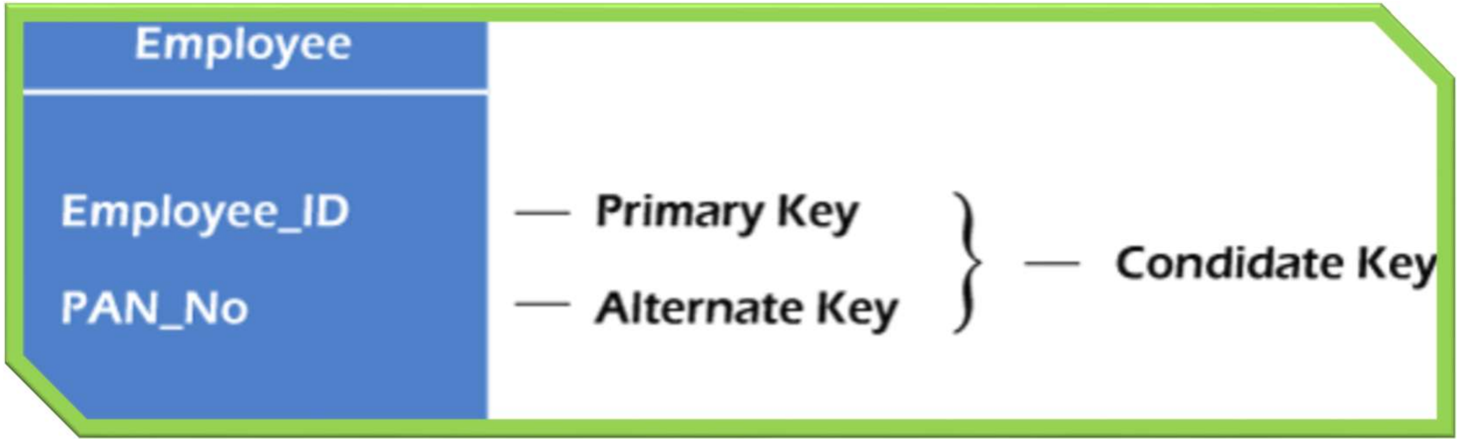
- ❑ There may be *one* **or** *more* attributes **or** *a combination* of attributes that uniquely identify each tuple in a relation. These attributes or combinations of the attributes are called the **candidate keys**.

- ❑ One key is chosen as the **primary key** from these candidate keys, and the remaining candidate key, if it exists, is termed the **alternate key**.

- ❑ In short, *the total number of the alternate keys is the total number of candidate keys minus the primary key*. The alternate key may or may not exist. If there is only one candidate key in a relation, it does not have an alternate key.

Alternate Key (contd..)

❑Let us take an example, where Employee relation has two attributes (e.g. Employee_Id & PAN_No) that can be considered as **candidate keys.** In this example, if **Employee_ID** is selected as the **Primary key**, in that case the other candidate key, **PAN_No** shall be considered as the **Alternate key**.



Composite Key

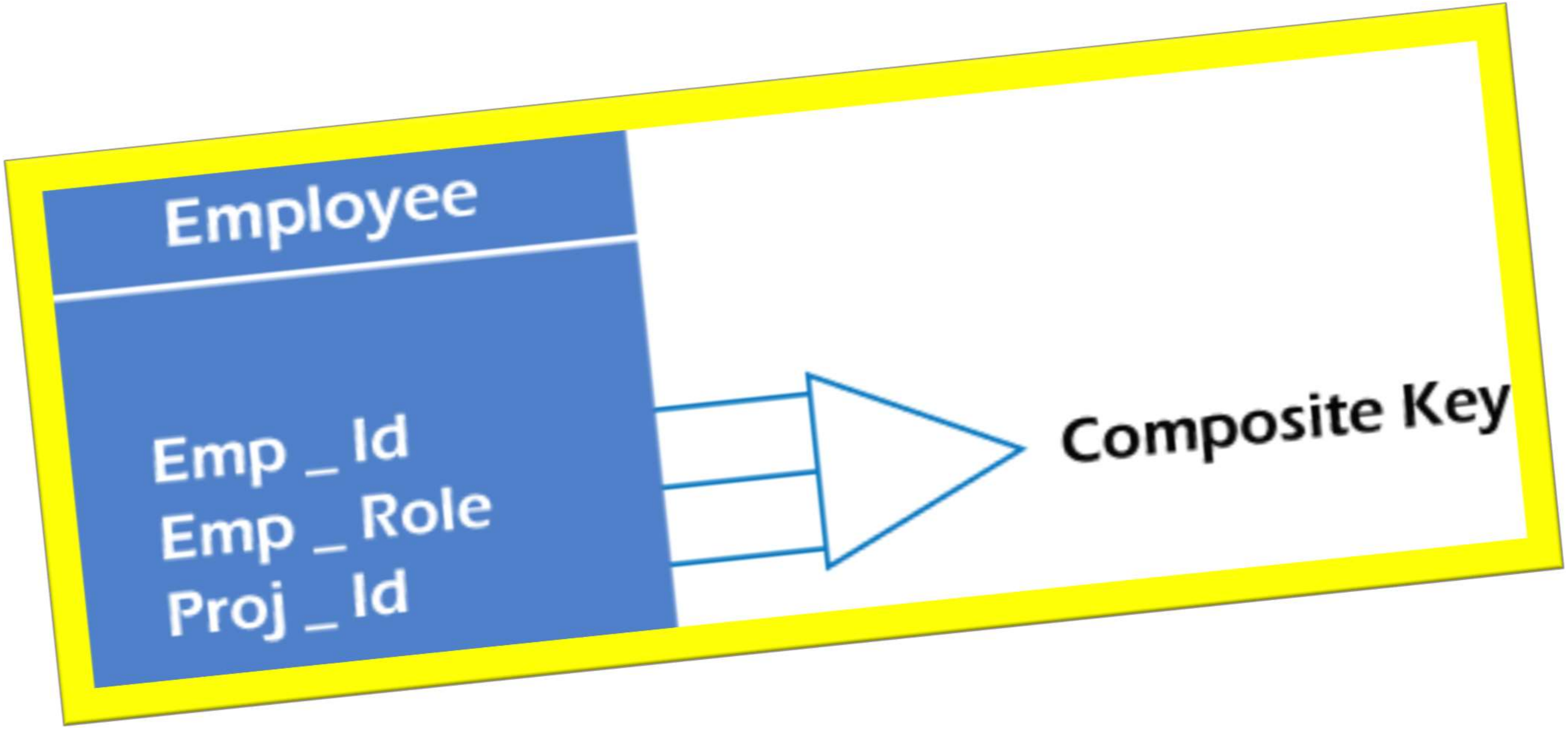
- Whenever a primary key consists of more than one attribute, it is known as a composite key.

- This key is also known as Concatenated Key.

- Let us consider example of employee relations. Assuming that an employee may get assigned multiple roles, and hence an employee may work on multiple projects simultaneously.

- Consequently, the primary key will be composed of all three attributes, namely **Emp_ID**, **Emp_role**, and **Proj_ID** in combination. So these attributes act as a *composite key* since the primary key comprises more than one attribute.

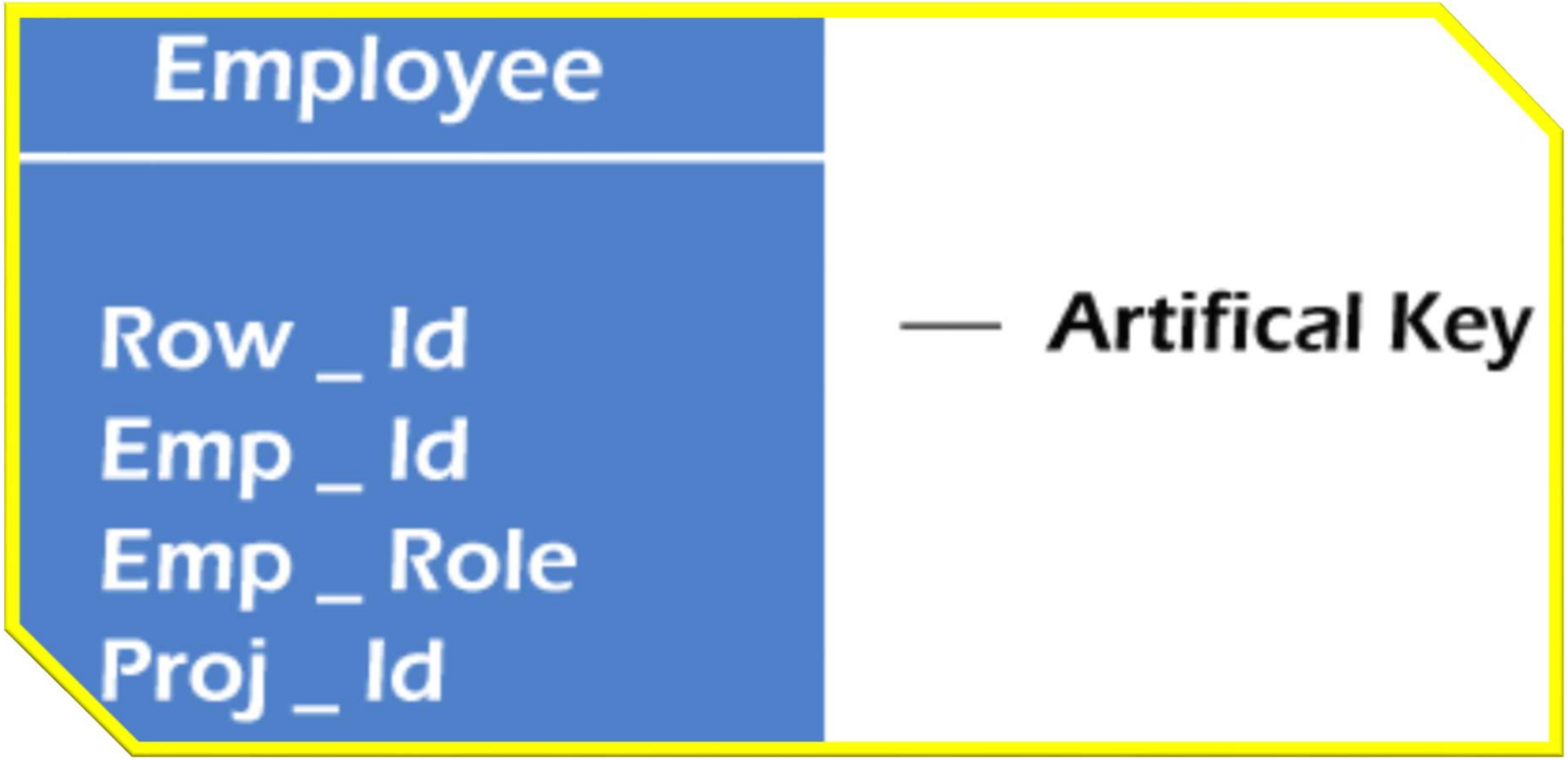
Composite Key (contd..)



Artificial Key

- ❑ The key created using arbitrarily assigned data are known as artificial keys. These keys are created when a primary key is large and complex and has no relationship with many other relations. The data values of the artificial keys are usually numbered in a serial order.
- ❑ Consider an example, where the primary key is composed of **Emp_ID**, **Emp_role**, and **Proj_ID**, is large in employee relations. *Consequently, it may be better to add a new virtual attribute to identify each tuple in the relation uniquely*.

Artifical Key (contd..)



Employee

Row _ Id
Emp _ Id
Emp _ Role
Proj _ Id

— Artifical Key

Any Questions?

Floor is Open

for

Discussion ….

References & Acknowledgements

- ❑ https://www.tutorialspoint.com/
- ❑https://www.mysqltutorial.org/mysql-administration/
- ❑https://www.javatpoint.com
- ❑https://education.oracle.com/learning-explorer