Relational Model

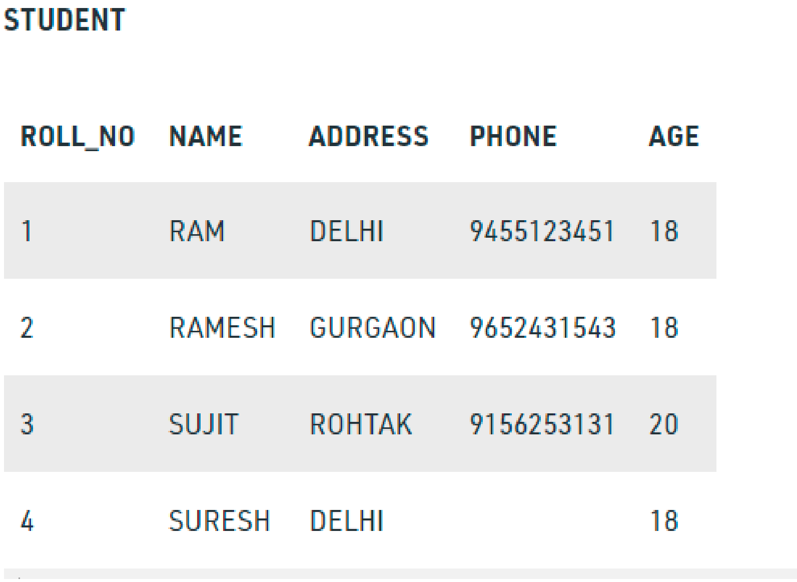
- By
- Jyoti Tryambake

Introduction

- Relational Model was proposed by E.F. Codd to model data in the form of relations or tables.
- After designing the conceptual model of Database using ER diagram,
  - convert the conceptual model in the relational model which can be implemented using any RDBMS languages like Oracle SQL, MySQL etc.

Introduction (cont.)

- Relational Model represents how data is stored in Relational Databases.
- A relational database stores data in the form of relations (tables).
- Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE and AGE shown in Table 1.



STUDENT

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

Terminologies

- **Attribute:** Attributes are the properties that define a relation. e.g.; **ROLL_NO, NAME**

- **Relation Schema:** A relation schema represents name of the relation with its attributes. e.g.; STUDENT (ROLL_NO, NAME, ADDRESS, PHONE and AGE) is relation schema for STUDENT. If a schema has more than 1 relation, it is called Relational Schema.

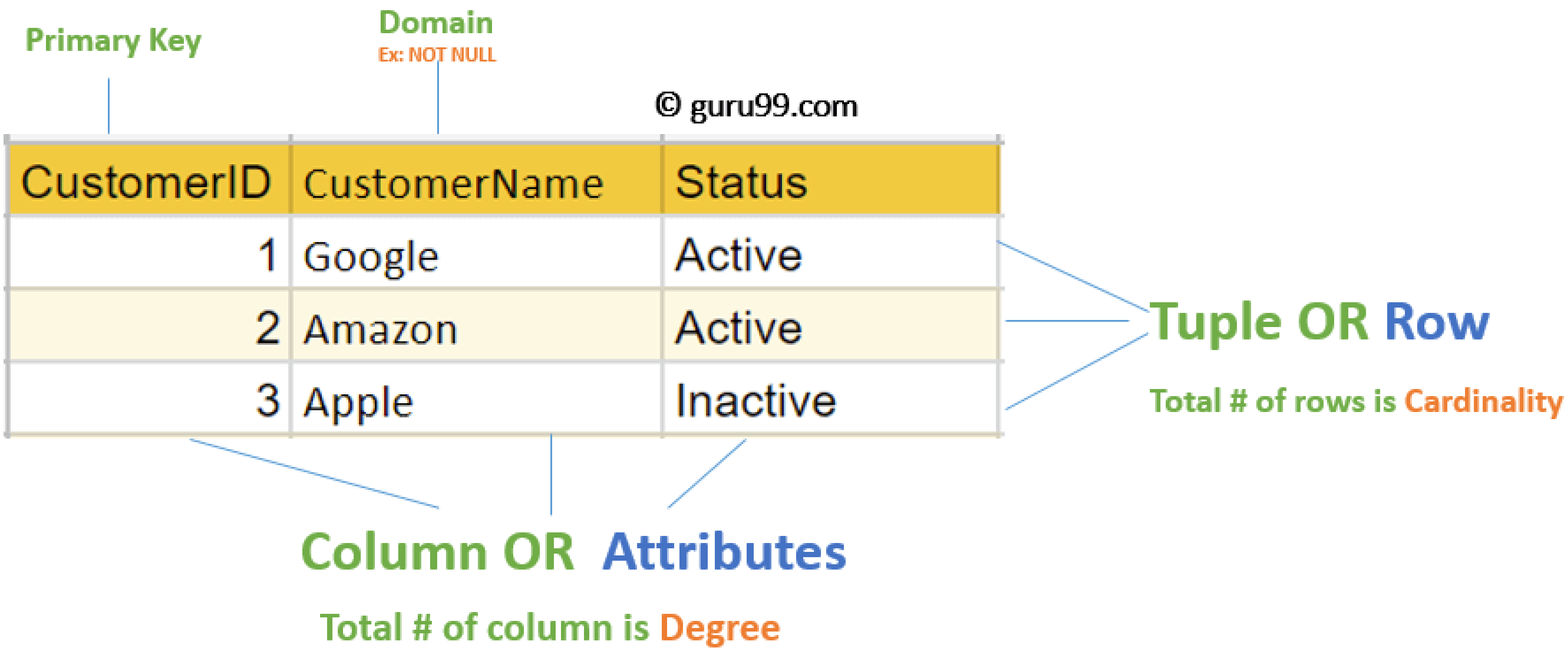
- **Tuple:** Each row in the relation is known as tuple. The above relation contains 4 tuples, one of which is shown as:

1	RAM	DELHI	9455123451	18

Terminologies

- **Relation Instance:** The set of tuples of a relation at a particular instance of time is called as relation instance. Table 1 (next slide) shows the relation instance of STUDENT at a particular time. It can change whenever there is insertion, deletion or updation in the database.
- **Degree:** The number of attributes in the relation is known as degree of the relation. The **STUDENT** relation defined previously (*previous slide- tuple example*) has degree 5.
- **Cardinality:** The number of tuples in a relation is known as cardinality. Let The **STUDENT** relation defined above has cardinality 4.
- **Column:** Column represents the set of values for a particular attribute. The column **ROLL_NO** is extracted from relation STUDENT.
- **NULL Values**: The value which is not known or unavailable is called NULL value. It is represented by blank space. e.g.; PHONE of STUDENT having ROLL_NO 4 is NULL.

Table also called Relation



Primary Key

Domain

Ex: NOT NULL

© guru99.com

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Tuple OR Row

Total # of rows is Cardinality

Column OR Attributes

Total # of column is Degree

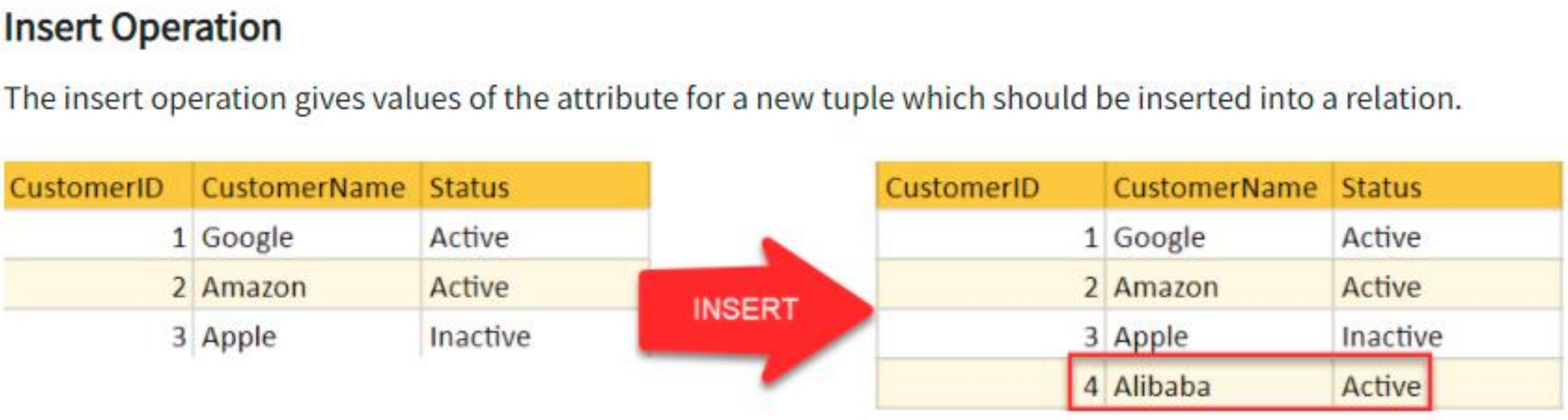
Data Manipulation

Four basic update operations performed on relational database model are

Insert, update, delete and select.

- Insert is used to insert data into the relation
- Delete is used to delete tuples from the table.
- Modify allows you to change the values of some attributes in existing tuples.
- Select allows you to choose a specific range of data.

Data Manipulation

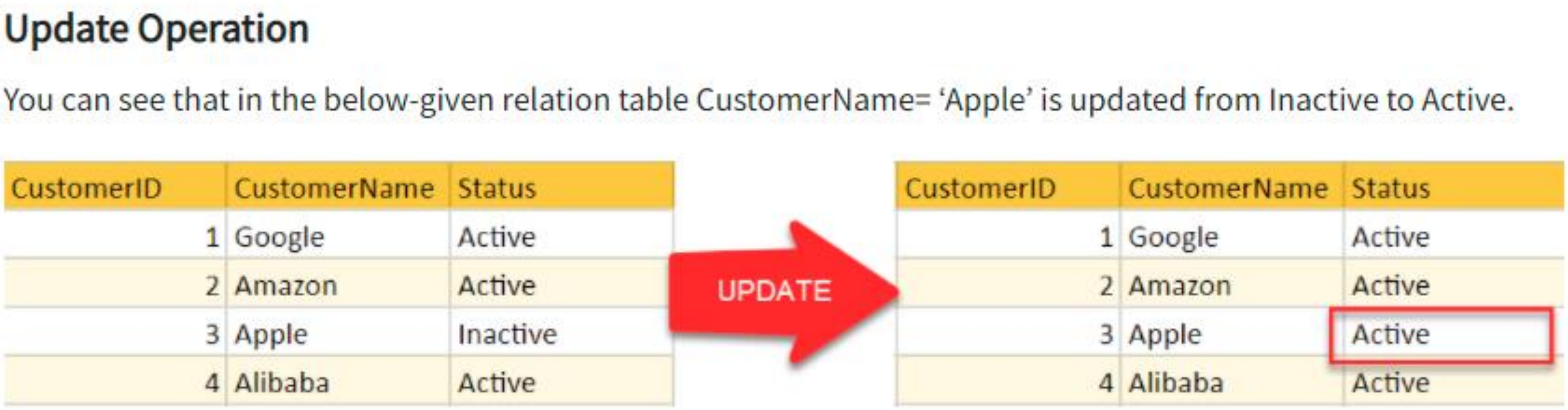


Insert Operation

The insert operation gives values of the attribute for a new tuple which should be inserted into a relation.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active



Update Operation

You can see that in the below-given relation table CustomerName= 'Apple' is updated from Inactive to Active.

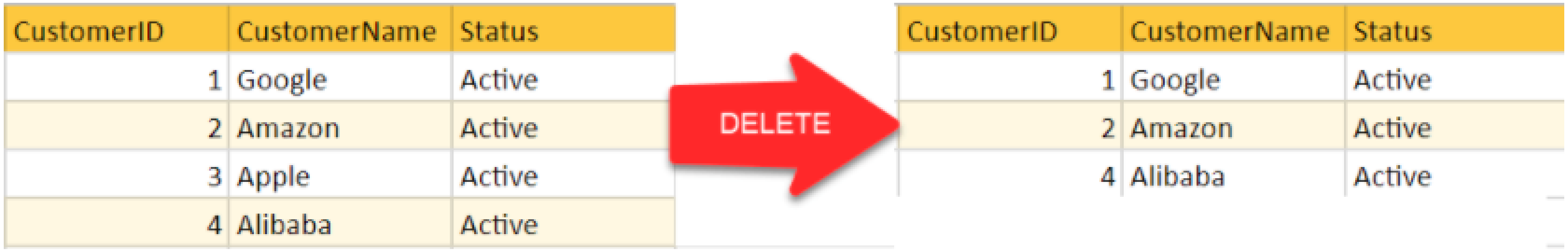
CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

Data Manipulation

Delete Operation

To specify deletion, a condition on the attributes of the relation selects the tuple to be deleted.



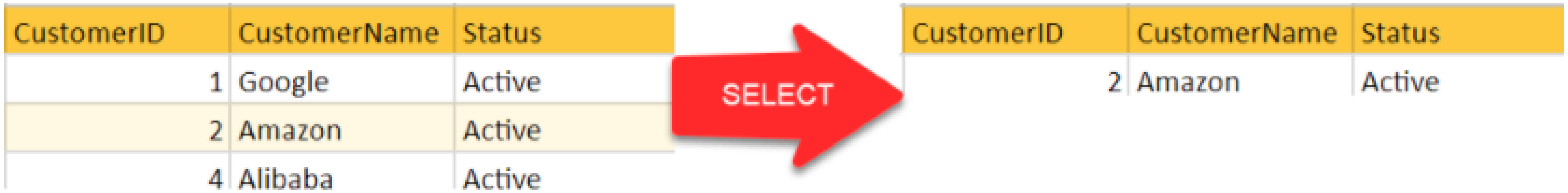
CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

In the above-given example, CustomerName= “Apple” is deleted from the table.

The Delete operation could violate referential integrity if the tuple which is deleted is referenced by foreign keys from other tuples in the same database.

Select Operation



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

CustomerID	CustomerName	Status
2	Amazon	Active

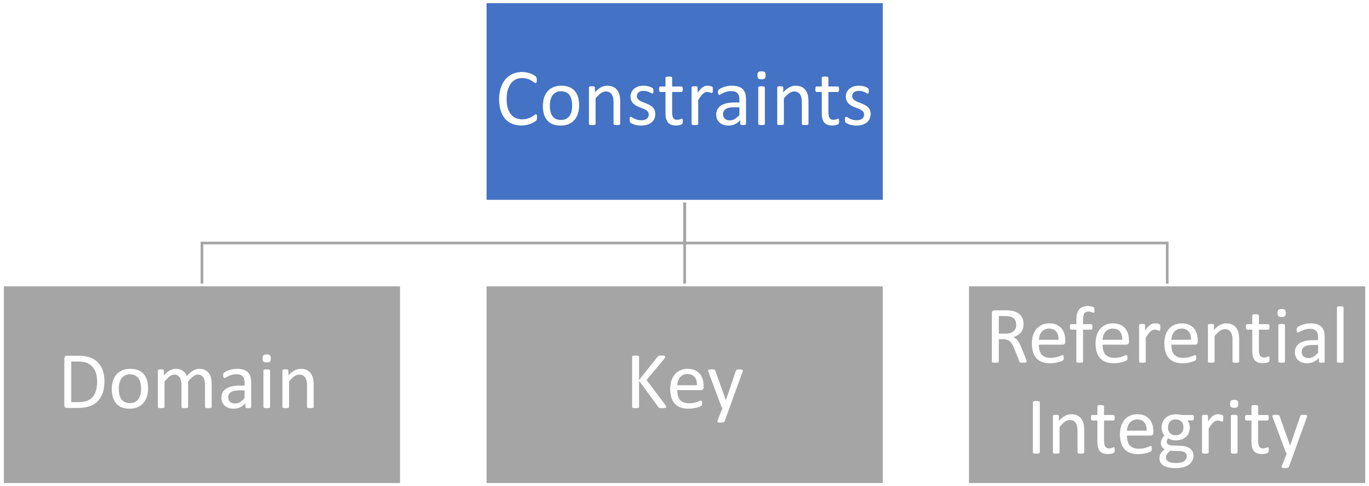
In the above-given example, CustomerName=”Amazon” is selected

Best Practices for creating a Relational Model

- Data need to be represented as a collection of relations
- Each relation should be depicted clearly in the table
- Rows should contain data about instances of an entity
- Columns must contain data about attributes of the entity
- Cells of the table should hold a single value
- Each column should be given a unique name
- No two rows can be identical
- The values of an attribute should be from the same domain (*domain constraint*)

Relational Integrity Constraint

- Relational Integrity constraints in DBMS are referred to conditions which must be present for a valid relation.



Domain Constraints

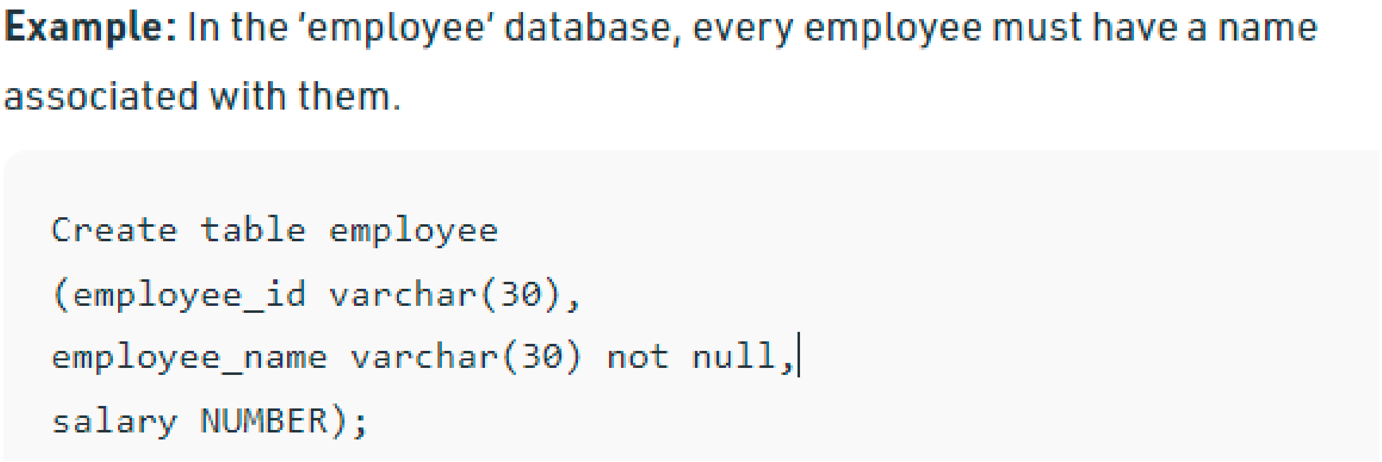
- Domain Constraints are user-defined columns that help the user to enter the value according to the data type.
- Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.
- Domain constraints specify that within each tuple, and the value of each attribute must be unique.
- This is specified as data types which include standard data types integers, real numbers, characters, Booleans, variable length strings, etc.

Domain Constraints (cont.)

- Domain Constraint = data type(integer / character/date / time / string / etc.) + Constraints(NOT NULL / UNIQUE / PRIMARY KEY / FOREIGN KEY / CHECK / DEFAULT)

Domain Constraints – Not Null

- Unassigned values/unknown or the missing attribute values and by default, a column can hold the null values.
- Not Null constraint restricts a column to not accept the null values which means it only restricts a field to always contain a value.

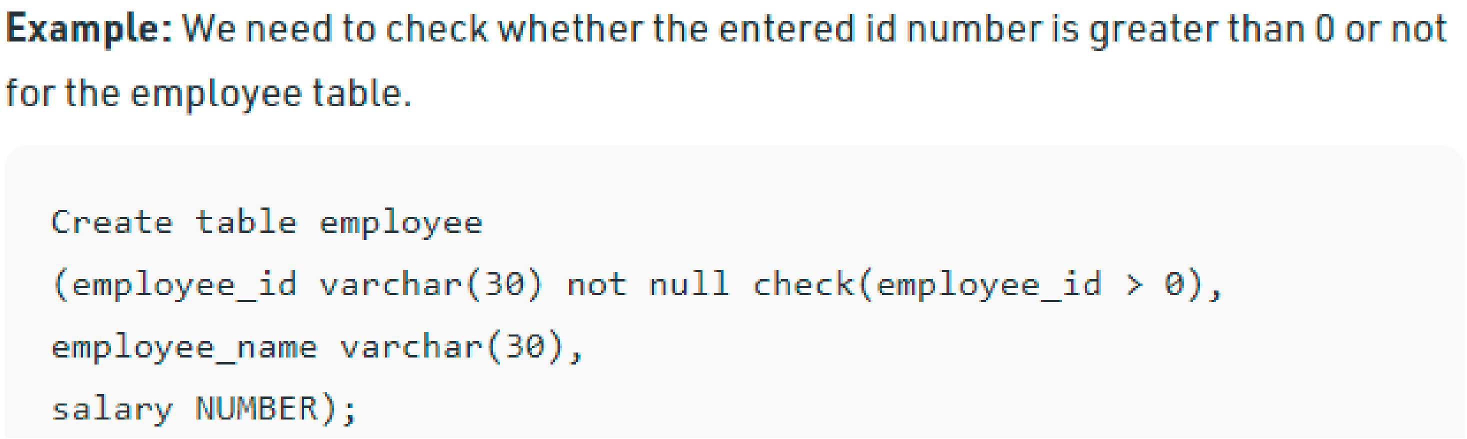


Example: In the 'employee' database, every employee must have a name associated with them.

```
Create table employee
(employee_id varchar(30),
employee_name varchar(30) not null,
salary NUMBER);
```

Domain Constraints – Check

- It defines a condition that each row must satisfy which means it restricts the value of a column between ranges.
- It is just like a condition or filter checking before saving data into a column.
- It ensures that when a tuple is inserted inside the relation must satisfy the predicate given in the check clause.

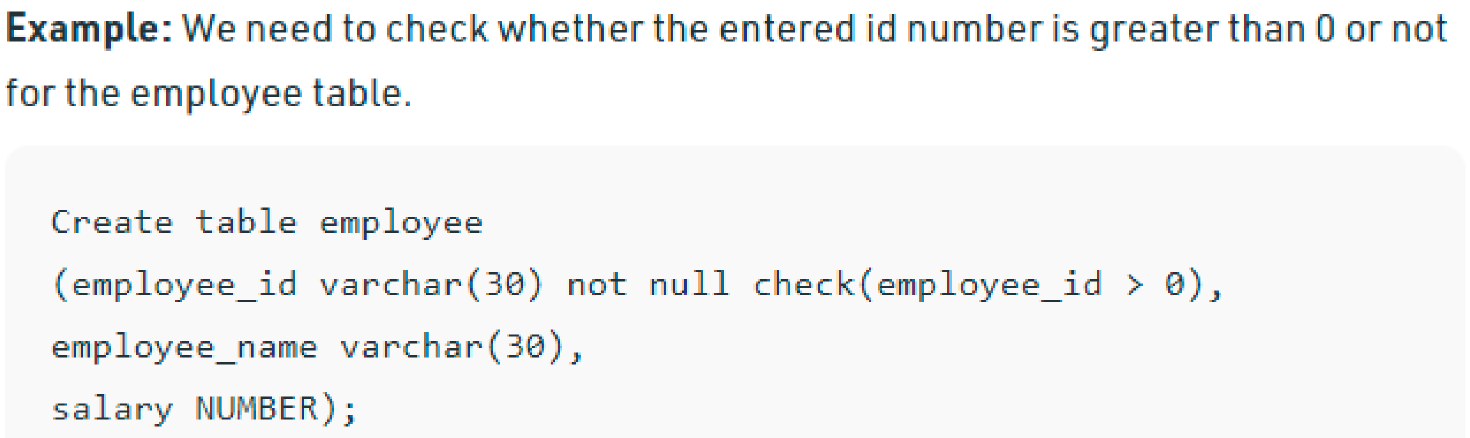


Example: We need to check whether the entered id number is greater than 0 or not for the employee table.

```
Create table employee
(employee_id varchar(30) not null check(employee_id > 0),
employee_name varchar(30),
salary NUMBER);
```

Domain Constraints – Check

- It defines a condition that each row must satisfy which means it restricts the value of a column between ranges.
- It is just like a condition or filter checking before saving data into a column.
- It ensures that when a tuple is inserted inside the relation must satisfy the predicate given in the check clause.



Example: We need to check whether the entered id number is greater than 0 or not for the employee table.

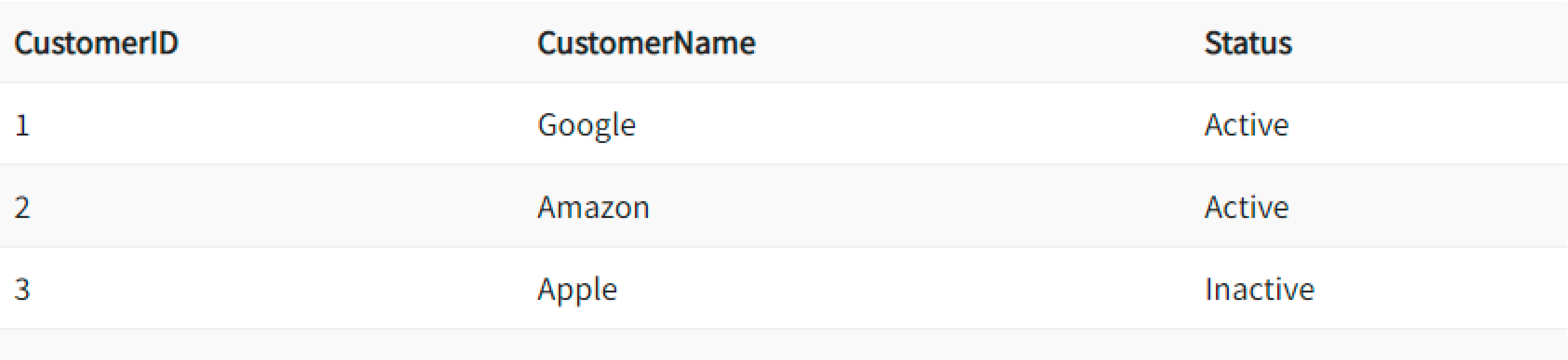
```
Create table employee
(employee_id varchar(30) not null check(employee_id > 0),
employee_name varchar(30),
salary NUMBER);
```

Key Constraints

- An attribute that can uniquely identify a tuple in a relation is called the key of the table. The value of the attribute for different tuples in the relation has to be unique.

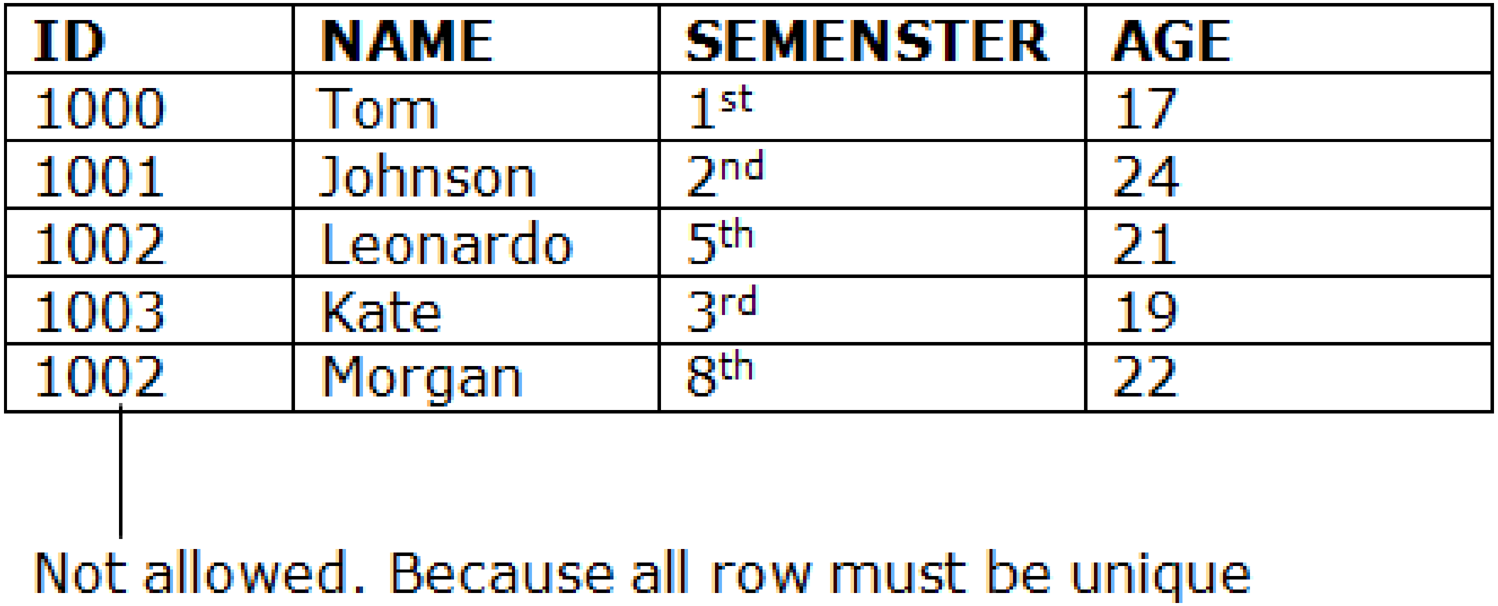
Example:

- In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =” Google”.



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Key Constraints (cont.)



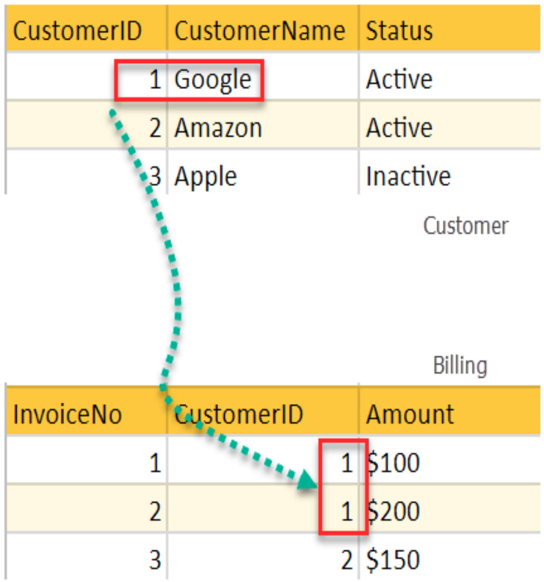
ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1002	Morgan	8th	22

Not allowed. Because all row must be unique

Referential Integrity Constraints

- Referential Integrity constraints in DBMS are based on the concept of Foreign Keys.
- A foreign key is an important attribute of a relation which should be referred to in other relationships.
- Referential integrity constraint state happens where relation refers to a key attribute of a different or same relation.
- However, that key element must exist in the table.

Example:



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Customer

Billing

InvoiceNo	CustomerID	Amount
1	1	$100
2	1	$200
3	2	$150

Advantages of Relational Database Model

- **Simplicity**: A Relational data model in DBMS is simpler than the hierarchical and network model.
- **Structural Independence**: The relational database is only concerned with data and not with a structure. This can improve the performance of the model.
- **Easy to use**: The Relational model in DBMS is easy as tables consisting of rows and columns are quite natural and simple to understand
- **Query capability**: It makes possible for a high-level query language like SQL to avoid complex database navigation.
- **Data independence**: The Structure of Relational database can be changed without having to change any application.
- **Scalable**: Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.