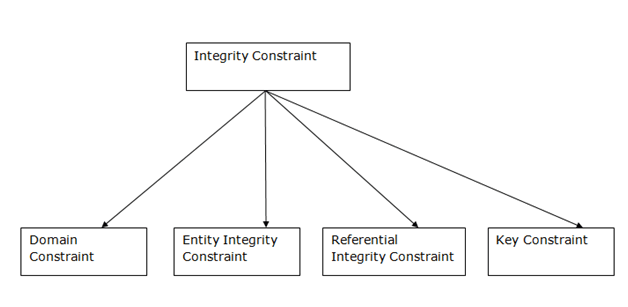
**Relational Integrity Constraints**

Every relation in a relational database model should abide by or follow a few constraints to be a valid relation, these constraints are called as **Relational Integrity Constraints**. Relational Integrity constraints in DBMS are referred to conditions which must be present for a valid relation.

Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected. Thus, integrity constraint is used to guard against accidental damage to the database.



The three main Integrity Constraints are:

1. Key Constraints
2. Domain Constraints
3. Referential integrity Constraints

* **Key Constraints**

We store data in tables, to later access it whenever required. In every table one or more than one attributes together are used to fetch data from tables. The Key Constraint specifies that there should be such an attribute(column) in a relation(table), which can be used to fetch data for any tuple(row). 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.

The Key attribute should never be **NULL** or same for two different rows of data.

For example, in the Employee table we can use the attribute ID to fetch data for each of the employee. No value of ID is null and it is unique for every row, hence it can be our key attribute.

**Example:**

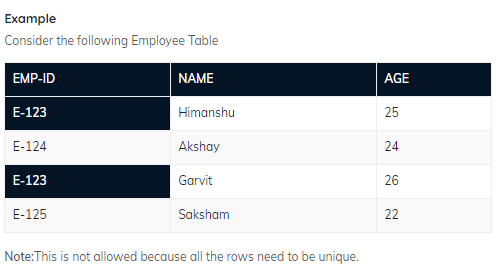
In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =" Google".

**CustomerID CustomerName Status**

1 Google Active

2 Amazon Active

3 Apple Inactive



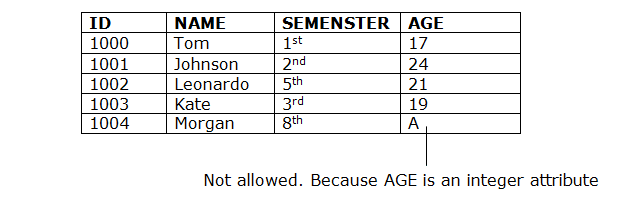
* **Domain Constraint**

Domain constraints refers to the rules defined for the values that can be stored for a certain attribute. 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.

Like we explained above, we cannot store Address of employee in the column for Name.

Similarly, a mobile number cannot exceed 10 digits.

**Example:**  
 

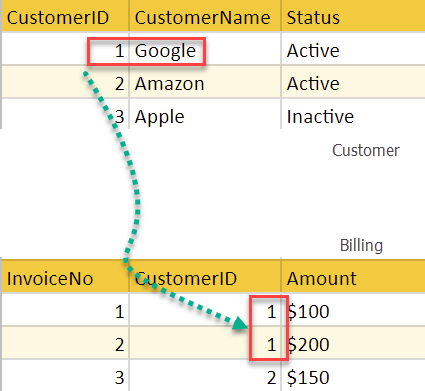
* **Referential Integrity Constraint**

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. If a table reference to some data from another table, then that table and that data should be present for referential integrity constraint to hold true.

The following two important results emerges out due to referential integrity constraint-

* We cannot insert a record into a referencing relation if the corresponding record does not exist in the referenced relation.
* We cannot delete or update a record of the referenced relation if the corresponding record exists in the referencing relation.

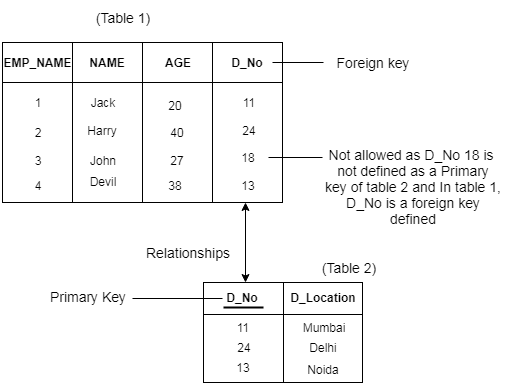
**Example:**



In the above example, we have 2 relations, Customer and Billing.

Tuple for CustomerID =1 is referenced twice in the relation Billing. So, we know CustomerName=Google has billing amount $300

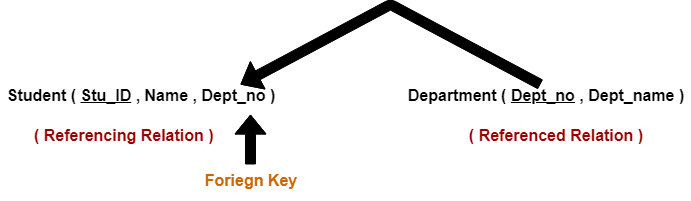
**Example:**



**Example-**

Consider the following two relations- ‘Student’ and ‘Department’.

Here, relation ‘Student’ references the relation ‘Department’.



**Student**

**STU\_ID Name Dept\_no**

S001 Akshay D10

S002 Abhishek D10

S003 Shashank D11

S004 Rahul D14

**Department**

**Dept\_no Dept\_name**

D10 ASET

D11 ALS

D12 ASFL

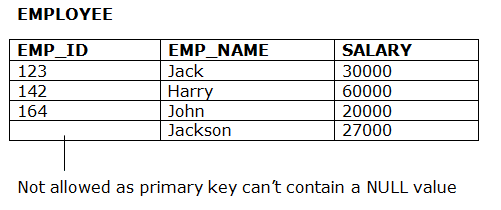
D13 ASHS

Here,

* The relation ‘Student’ does not satisfy the referential integrity constraint.
* This is because in relation ‘Department’, no value of primary key specifies department no. 14.
* Thus, referential integrity constraint is violated.
* **Entity Integrity Constraints**

The entity integrity constraint states that primary key value can't be null. This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows. A table can contain a null value other than the primary key field.

**Example:**



* **Tuple Uniqueness Constraint**

Tuple Uniqueness constraint specifies that all the tuples must be necessarily unique in any relation.

**Example-01:**

Consider the following Student table-

**STU\_ID Name Age**

S001 Akshay 20

S002 Abhishek 21

S003 Shashank 20

S004 Rahul 20

This relation satisfies the tuple uniqueness constraint since here all the tuples are unique.

**Example-02:**

Consider the following Student table-

**STU\_ID Name Age**

S001 Akshay 20

S001 Akshay 20

S003 Shashank 20

S004 Rahul 20

This relation does not satisfy the tuple uniqueness constraint since here all the tuples are not unique.

**Operations in Relational Model**

Four basic update operations performed on relational database model are

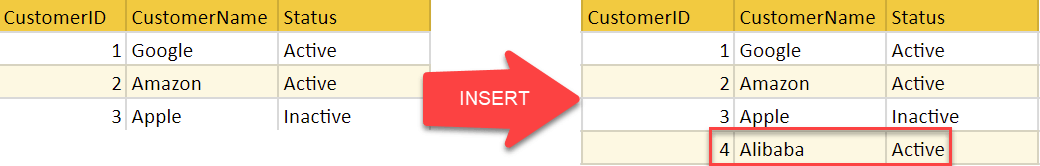
**Insert, update, delete and select.**

* 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.

Whenever one of these operations are applied, integrity constraints specified on the relational database schema must never be violated.

* **Insert Operation**

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



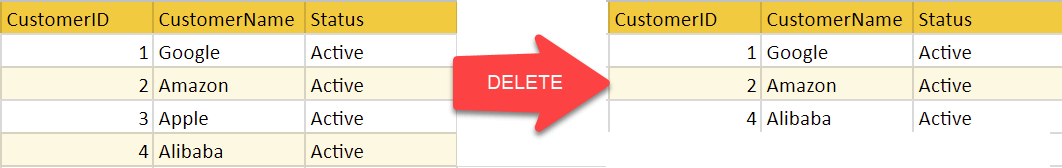
* **Update Operation**

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



* **Delete Operation**

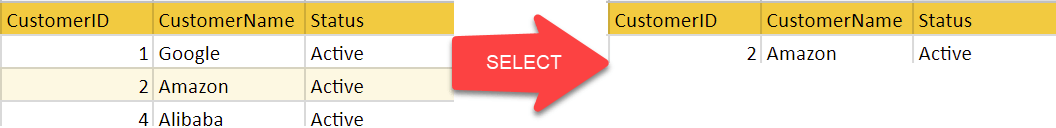
To specify deletion, a condition on the attributes of the relation selects the tuple to be deleted.



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**



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