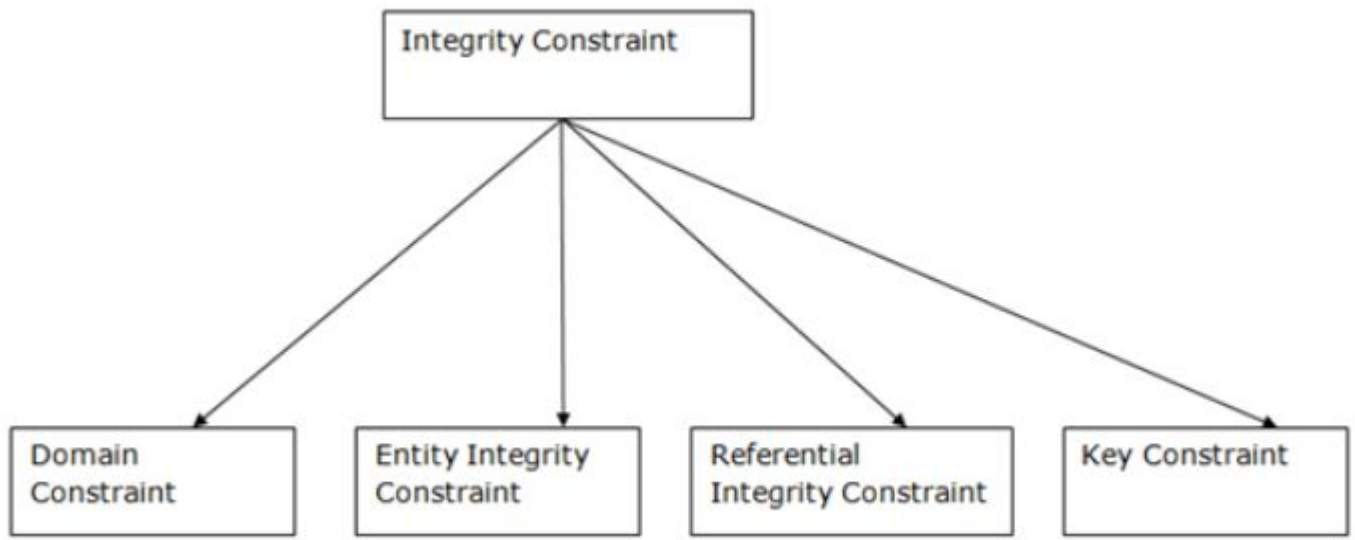
Data integrity

- Data Integrity is having correct and accurate data in your database. When we are storing data in the database we don't want repeating values, we don't want incorrect values or broken relationships between tables.

Integrity Constraints

- Integrity constraints are a set of rules. It is used to maintain the quality of information.
- 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.


Integrity Constraint
Domain Constraint
Entity Integrity Constraint
Referential Integrity Constraint
Key Constraint

Domain constraints

- Domain constraints can be defined as the definition of a valid set of values for an attribute.
- The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.

ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

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.

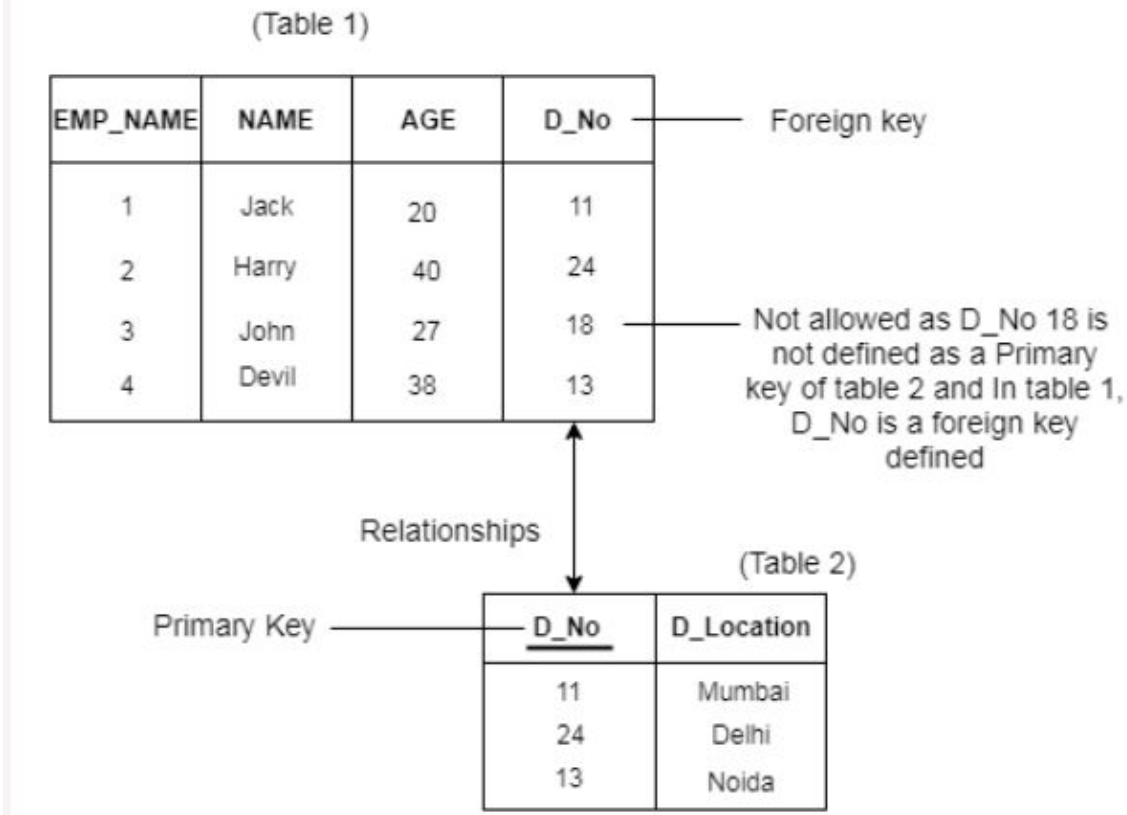
EMPLOYEE

EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value

Referential Integrity Constraints

- A referential integrity constraint is specified between two tables.
- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.


(Table 1)
EMP_NAME
NAME
AGE
D_No
Foreign key
1
Jack
20
11
2
Harry
40
24
3
John
27
18
4
Devil
38
13
Not allowed as D_No 18 is not defined as a Primary key of table 2 and In table 1, D_No is a foreign key defined
Relationships
(Table 2)
Primary Key
D_No
D_Location
11
Mumbai
24
Delhi
13
Noida

Key constraints

- Keys are the entity set that is used to identify an entity within its entity set uniquely.
- An entity set can have multiple keys, but out of which one key will be the primary key. A primary key can contain a unique and null value in the relational table.

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

Assertions

- Assertions are different from check constraints in the way that check constraints are rules that relate to one single row only. Assertions, on the other hand, can involve any number of other tables, or any number of other rows in the same table. Assertions also check a condition, which must return a Boolean value. We can take an illustrative example.

- Let us imagine that we have the following table, which contains employees in a company — we then also store an attribute containing their salary.

ID	Name	Age	DepNo	Salary
0	Hannah	18	10	1000$
1	Gavin	45	2	25.000$
2	Bobby	70	21	700$

- We then want to make an assertion that there is no employee in our database, which is paid more than 30.000$ or less than 500$. It would then look like :

```
Create Assertion SalaryAssertion check
        (Not exists
                (select ID
                From People p
                Where p.salary > 30.000 OR p.salary < 500
);
```

Employee(id,name,salary,age) salary >5000 age >18

Create Assertion e check (Not exists (Select * from Employee where salary < 5000 and age < 18)));

Relationship in database

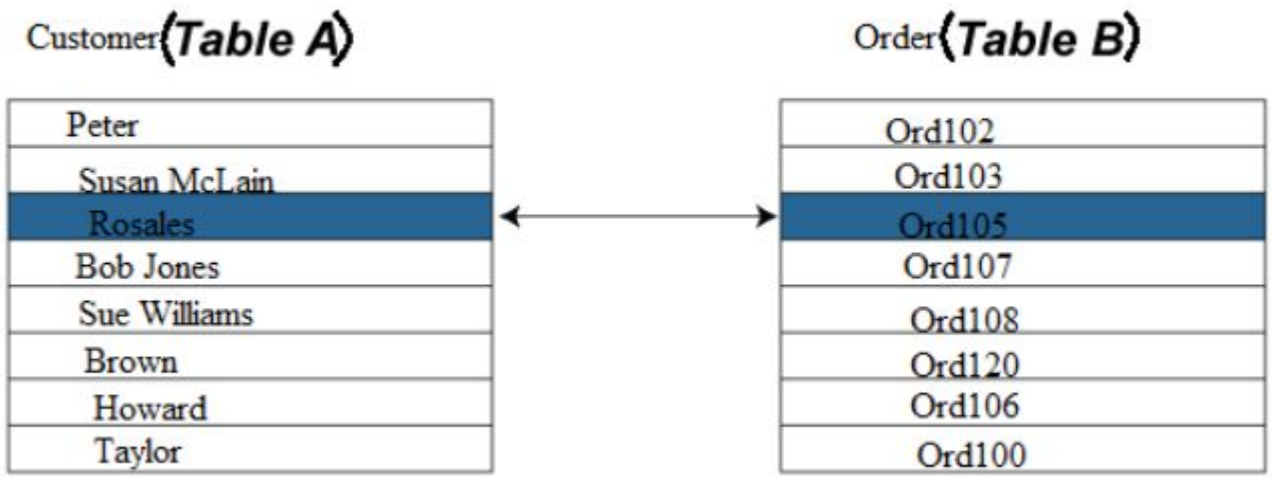
- In relational databases, a relationship exists between two tables when one of them has a foreign key that references the primary key o
- This single fact allows relational databases to split and store data in different tables, yet still link the disparate data items together.
- It is one of the features that makes relational databases such powerful and efficient stores of information.f the other table.

Types of Relationship

1. One to One relationship
2. One to many
3. many to one relationship
4. Many to many relationships

One to One Relationship (1:1):

- It is used to create a relationship between two tables in which a single row of the first table can only be related to one and only one records of a second table. Similarly, the row of a second table can also be related to anyone row of the first table.

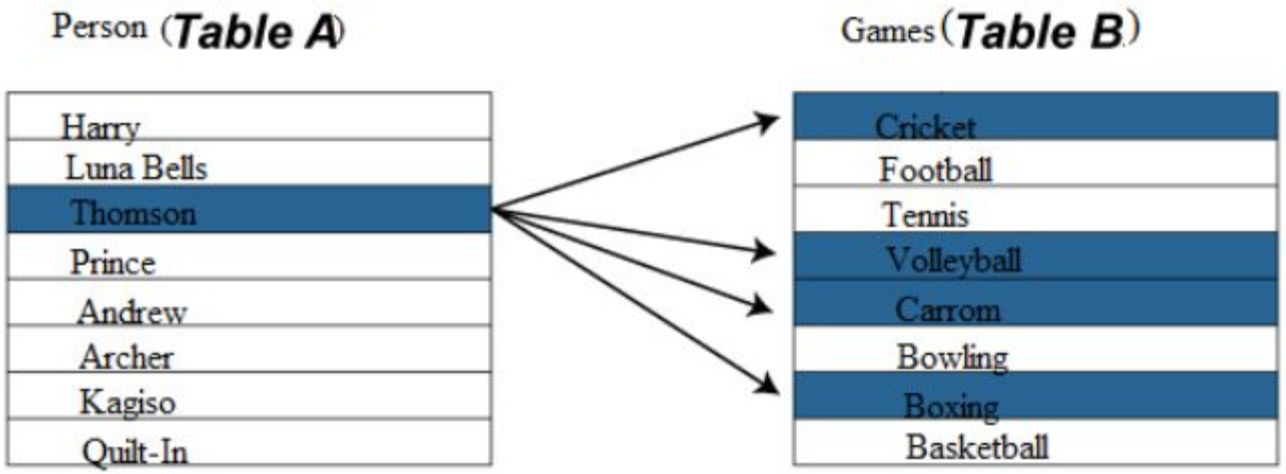

Customer(Table A)
Order(Table B)
Peter
Susan McLain
Rosales
Bob Jones
Sue Williams
Brown
Howard
Taylor
Ord102
Ord103
Ord105
Ord107
Ord108
Ord120
Ord106
Ord100

One to Many Relationship:

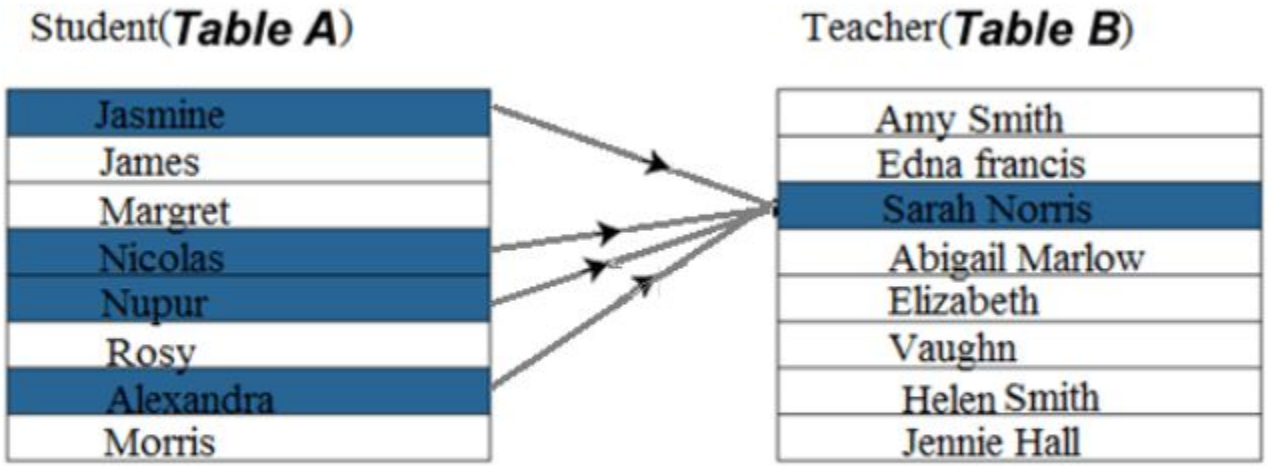
- It is used to create a relationship between two tables. Any single rows of the first table can be related to one or more rows of the second tables, but the rows of second tables can only relate to the only row in the first table. It is also known as a **many to one** relationship.

One to many relation ship

Representation of **One to Many** relational databases:

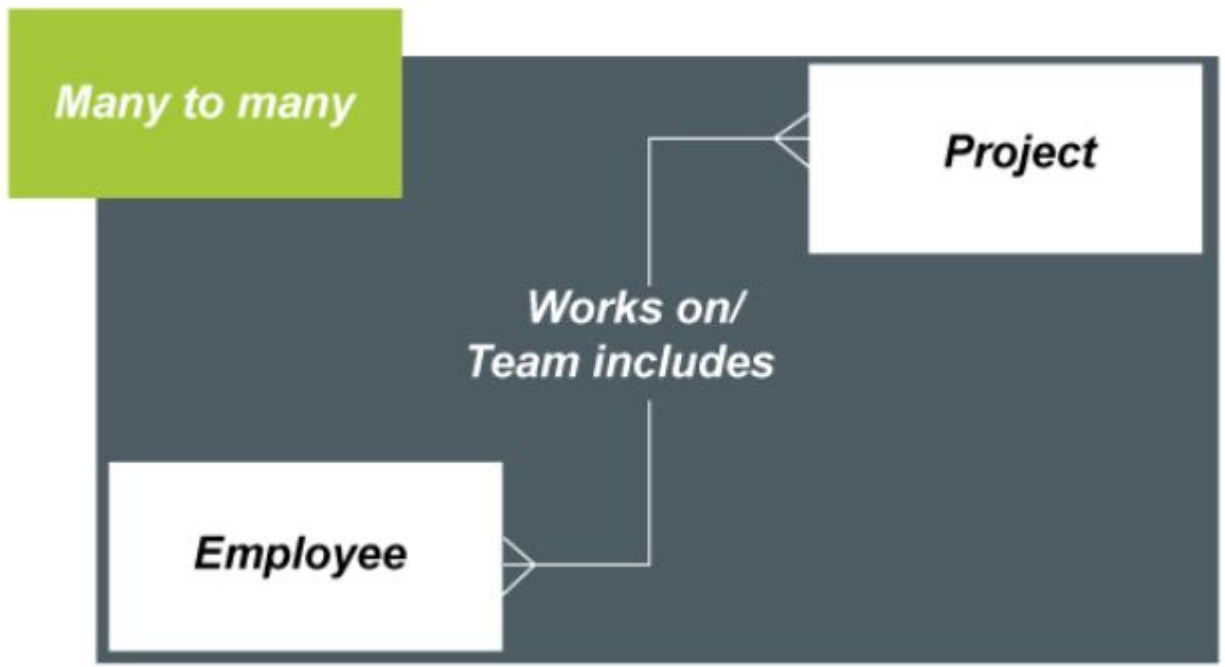


Many to one


Student(Table A)
Jasmine
James
Margret
Nicolas
Nupur
Rosy
Alexandra
Morris
Teacher(Table B)
Amy Smith
Edna francis
Sarah Norris
Abigail Marlow
Elizabeth
Vaughn
Helen Smith
Jennie Hall

Many to Many Relationship:

- It is **many to many** relationships that create a relationship between two tables. Each record of the first table can relate to any records (or no records) in the second table. Similarly, each record of the second table can also relate to more than one record of the first table. It is also represented an **N:N** relationship.


Many to many
Project
Works on/
Team includes
Employee