## Integrity Constraints

Integrity constraints are rules that should  be applied on database columns to ensure the validity of data. By Guarding against accidental damage to the database.

Integrity constraints make sure that the data insertion, updating and other processes are performed in such a way that the integrity of data is not affected.

With all the authorised changes to the database happening, integrity constraints ensure that there is no loss of data consistency.

All the relations in a Relational Database Model need to follow some rules or constraints to maintain the integrity of data, these constraints are called Relational Integrity Constraints.

There are three type of integrity constraints –

1. Domain Constraints
2. Entity Constraints
3. Referential Constraints

## Domain Constraints

The domain integrity constraints restrict the value in the particular attributes of a relation.

Therefore, if in an attribute we want to take email as input, then for ensuring that email is valid, we can use domain constraints.

Before inserting email in the table, checks can be performed for validation.

Let’s take a different example:

We need to ensure that semester student is registering for is one of fall, winter, spring or summer:

Schema design for the Relation section:-

```
CREATE TABLE section (
course_id VARCHAR(8),
sec_id VARCHAR(8),
semester VARCHAR(6),
year NUMERIC(4,0),
building VARCHAR(15),
room_number VARCHAR(7),
time_slot_id VARCHAR(4),
primary key (course_id, sec_id, semester, year),
check (semester in (’Fall’, ’Winter’, ’Spring’, ’Summer’))
);
```

In purple we have marked the check(P). ‘P’ is the predicate on which checks are performed.

## Entity Constraints

- It puts constraints on Primary key i.e. primary should be unique and does not have NULL value.
- This constraint ensures that every row is uniquely identified in our table by ensuring the first constraint stated above.

## Referential Constraints

- It is specified between two relations and helps maintain consistency among the tuples of two relations.
- It ensures that if a value appears in one relation for a given set of attributes also should also appear for a certain set of attributes in another relation.
- Example: If “Prosthodontics” is a department name appearing in one of the tuples in the instructor relation, then there exists a tuple in the department relation for “Prosthodontics”.
- 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.

**Relation Project:**

| ProjectID | EmpID | ProjectName |
|---|---|---|
| 100 | 1 | pro_1 |
| 200 | 2 | pro_2 |
| 300 | 3 | pro_3 |
| 400 | 5 | pro_4 |

**Table Employee:**

| EmpID | EmpFname | EmpLname | Age | EmailID |
|---|---|---|---|---|
| 1 | Riya | Khanna | 21 | riya@abc.com |
| 2 | Sahil | Kumar | 32 | sahil@abc.com |
| 3 | Vishwas | Aanand | 24 | vishwas@abc.com |
| 4 | Harleen | Kaur | 27 | harleen@abc.com |

Now in the above example, EmpId in Relation Project is a Foreign key, and is a primary key in Relation Employee. (marked in blue)
EmpID = 5 (marked as red) in the foreign key of the relation Project is not allowed. Since, EmpID = 5 is not defined in the primary key of the relation Employee.
Hence, leading to Violation of Referential Constraints.

## Designing a database

Problem statement – Design order management system that has a database which captures all the orders of all the customers.

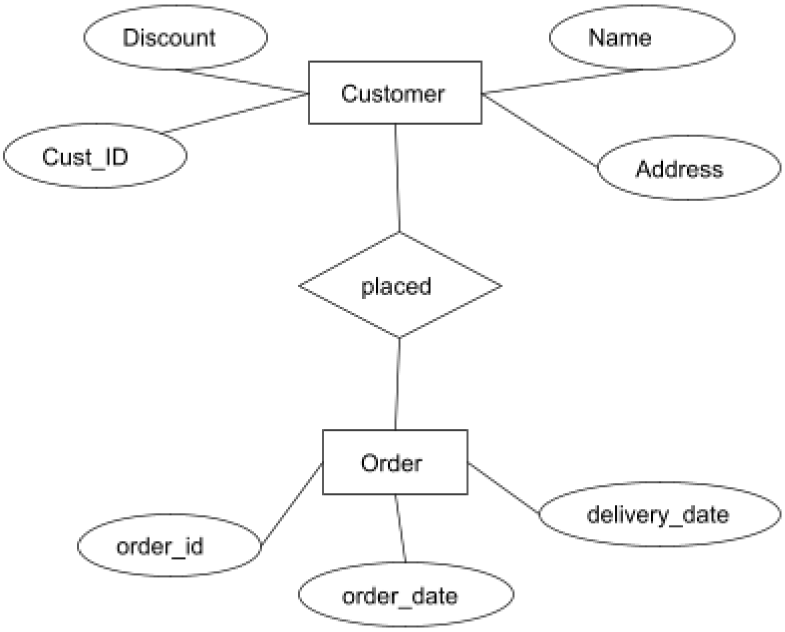
### Steps

1. Come up with the idea
2. Convert it to E/R diagram
3. Relational Model
4. Relational Database

## ER Diagram

- First we'll try to figure out all the entities.
- There will be two entities: customer and order. The relationship between the entity's customer and order is placed.
- Now we'll try to figure out attributes of both the entities.
- The attributes of the customer could be Customer Id, Name, Address, Discount provided to that customer.
- Attributes of order could be Order_ID, Delivery_Date, Order_Date.

Now after figuring out entities, attributes and relationship between entities, we're good to proceed for the ER diagram.



After that we can create a relational table according to ER Model.

**Relation Customer:**

| Cust_id | Name | Address | Discount | Order_id |
|---|---|---|---|---|
| 8495 | Uday | 97-B Model Town, Jalandhar, Punjab | 30% | 323134 |
| 4564 | Reet | 813 Aadarsh villa, Nokha, Rajasthan | 55% | 523623 |

**Relation Order:**

| Order_id | Order_date | Delivery_date | Cust_id |
|---|---|---|---|
| 323134 | 25/08/2021 | 27/08/2021 | 8495 |
| 523623 | 02/06/2021 | 05/06/2021 | 4564 |

Note: for the above example, In each relation a foreign key is marked with blue and primary is marked with red.