**Practical - 1**

**Case study of Normalisation**

Normalisation :- Normalization is a method to remove all the anomalies and bring the database to a consistent state .

It transforms data from a problem into relations while ensuring data integrity and eliminating data redundancy.

Consider an example to understand normalisation

**ONLINE BUS BOOKING SYSTEM**

Booking\_date: userid:

Username: email:

Ticket No: Seatno:

Departuredate: Departuretime:

PassengerId: PassengerName:

PassengerAddress: DOB:

GENDER: Phoneno:

Busno: Busname:

Capacity: type:

Routeno: Route-name:

Source: Destination:

Distance: Fare:

Modeof payment:

There are three types of anomalies that occur when the database is not normalized. These are – Insertion, update and deletion anomaly

The above table is not normalized. We will see the problems that we face when a table is not normalized

**Update anomaly**: In the above table we have two rows for employee Rick as he belongs to two departments of the company. If we want to update the address of Rick then we have to update the same in two rows or the data will become inconsistent. If somehow, the correct address gets updated in one department but not in other then as per the database, Rick would be having two different addresses, which is not correct and would lead to inconsistent data.

**Insert anomaly**: Suppose a new employee joins the company, who is under training and currently not assigned to any department then we would not be able to insert the data into the table if emp\_dept field doesn’t allow nulls.

**Delete anomaly**: Suppose, if at a point of time the company closes the department D890 then deleting the rows that are having emp\_dept as D890 would also delete the information of employee Maggie since she is assigned only to this department.

To overcome these anomalies we need to normalize the data. In the next section we will discuss about normalization.

The various Normal forms are described below :

**1st Normal Form (1NF)**

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

The new tables are as follows:

1)**USER**(***userid,***username,u\_email)

2)**PASSENGER**(***Pid,phno***,pname,paddress, DOB ,gender)

3)**BUS\_ROUTE**(***routeno***,***busno***,routename,source,destination,distance,fare,dept\_time,bname,capacity,type,dept\_date)

3)**RESERVATION**(***seatno,busno,*** status,bookingdate,ticketno, mode\_of\_payment)

**2nd Normal Form (2NF)**

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

The new tables are as follows:

**1)USER**(***userid***->username,u\_email)

***2)PASSENGER(pid->*** pname,paddress, DOB, gender,userid)

3)**CONTACTS**(***pid,phid***->phno)

**4)BUS\_ROUTE**(***routeno,busno***>bname,capacity,type,source,destination,distance,fare,Dept\_time,dept\_date)

**5)RESERVATION**(***seatno,busno->***status,bookingdate,ticketno, mode\_of\_payment)

**3rd Normal Form (3NF)**

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

The new tables are as follows:

**1)USER**

(***userid->***username,u\_email)

2)**PASSENGER**

(***pid***->pname,paddress,DOB, gender,userid)

**3)CONTACTS**

(***pid,phno***->phno)

***(As, busno->routeno and routeno->distance)***

Break **BUS\_ROUTE relation** into 2 tables**a)**

**BUS**

(***busno***->bname,capacity,type,routeno)

**b)ROUTE**

**Boycc Code Normal Form(BCNF)**

* A table is in BCNF if it is in 3NF and if every determinant is a candidate key.
* BCNF is a stronger form of 3NF
* BCNF => 3NF

1)BOOKING

(seatno-> pid,busno,status)

* Here,all the attributes other than seatno acts as a candidate key.
* Eg,pid can act as a primary key alone.
* Busno can also act as a primary key.
* Status is not unique(i.e.either booked or available),so we use(seatno and status) as candidate key.

**4TH Normal Form**

* A table is in 4NF if it is in BCNF and if it has no multi-valued dependencies.

**5TH Normal Form**

* A table is in 5NF, also called "ProjectionJoin Normal Form" (PJNF), if it is in 4NF and if every join dependency in the table is a consequence of the candidate keys of the table.