Explain Normalization techniques (along with an example).

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable anomalies like Insertion, Update and Deletion Anomalies.

Normalization Techniques :

1. **First normal form (1NF)**

   As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

   Example: Suppose a company wants to store the names and department details of its employees. It creates a table that looks like this:

   Table: Employee

Id	Name	Address	Department_Id
101	John	Delhi	D001, D002
123	Alice	Agra	D345
166	Glenn	Chennai	D466, D463

   This table is not in 1NF as the rule says “each attribute of a table must have atomic (single) values”, the Department_Id values for employees John & Glenn violates that rule.

   To make the table complies with 1NF we should have the data like this:

   Table: Employee

Id	Name	Address	Department_Id
101	John	Delhi	D001
101	John	Delhi	D002
123	Alice	Agra	D345

166	Glenn	Chennai	D466
166	Glenn	Chennai	D463

2. **Second normal form (2NF)**

A table is said to be in 2NF if both the following conditions hold:

- Table is in 1NF (First normal form)
- No non-prime attribute is dependent on the proper subset of any candidate key of the table.
- An attribute that is not part of any candidate key is known as a non-prime attribute.

Example: In table Employee, Since each employee can be part of multiple departments, the table can have multiple rows for each employee.

Candidate Keys: {Id, Name}

Non prime attribute: Department_Id

Table: Employee

Id	Name	Address
101	John	Delhi
123	Alice	Agra
166	Glenn	Chennai

Table: Department

Id	Department_Id
101	D001
101	D002
166	D466
166	D463

3. **Third Normal form (3NF)**
   - A table design is said to be in 3NF if both the following conditions hold:
   - Table must be in 2NF
   - Transitive functional dependency of non-prime attributes on any super key should be removed.
   - An attribute that is not part of any candidate key is known as a non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:

- X is a super key of table
- Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as a prime attribute.

Table: Employee

Id	Name	Zip	State	City	District
1001	John	28005	UP	Agra	Dayal Bagh
1002	Ajit	222008	TN	Chennai	M-city
1006	Lora	282007	TN	Chennai	Urapakkam
1101	Lilly	292008	UK	Pauri	Bhagwan
1201	Steve	222999	MP	Gwalior	Ratan

Super keys: {Id}, {Id, Name}, {Id,Name, Zip}…so on
Candidate Keys: {Id}
Non-prime attributes: all attributes except Id are non-prime as they are not part of any candidate keys.

Here State, City & District depend onZip. And, Zip is dependent on Id that makes non-prime attributes (State, City & District) transitively dependent on super key (Id). This violates the rule of 3NF.

Table1: Employee

Id	Name	Zip
1001	John	28005
1002	Ajit	222008
1006	Lora	282007
1101	Lilly	292008
1201	Steve	222999

Table2: Address

Zip	State	City	District
28005	UP	Agra	Dayal Bagh
222008	TN	Chennai	M-city
282007	TN	Chennai	Urapakkam
292008	UK	Pauri	Bhagwan
222999	MP	Gwalior	Ratan

4. **Boyce Codd normal form (BCNF)**

- It is an advanced version of 3NF that's why it is also referred to as 3.5NF.
- BCNF is stricter than 3NF.
- A table complies with BCNF if it is in 3NF and for every functional dependency X->Y, X should be the super key of the table.

Example: Suppose there is a table of authors with their books, genre and some info about each book:

Author	Nationality	Book title	Genre	Number of pages
William Shakespeare	English	The Comedy of Errors	Comedy	100
Markus Winand	Austrian	SQL Performance Explained	Textbook	200
Jeffrey Ullman	American	A First Course in Database Systems	Textbook	500
Jennifer Widom	American	A First Course in Database Systems	Textbook	500

The functional dependency:

book title → genre, number of pages

is one FD violating the BCNF rules.

We split our relation into two relations:

- the ones in the functional dependency (book title, genre, number of pages)
- the rest: (book title, author, nationality). Note that the left hand-side of the FD (book title) stays in the relation!

The example data looks like this. We select the values of columns from the original relation and we eliminate the duplicate rows.

Table: Author

Author	**Nationality**
William Shakespeare	English
Markus Winand	Austrian
Jeffrey Ullman	American
Jennifer Widom	American

The (book title, genre, number of pages) table is in BCNF.
But (book title, author, nationality) isn't.
We have the dependency:

author → nationality

Together with the trivial dependency

book title → book title,

the pair (book title, author) is the key of the relationship.

We have to decompose the table one more time. This time we decompose into:

- columns forming the functional dependency: (author, nationality)
- the remaining columns: (author, book title)

This time every table is in BCNF.

Table: Book

Book title	**Genre**	**Number of pages**
The Comedy of Errors	Comedy	100
SQL Performance Explained	Textbook	200
A First Course in Database Systems	Textbook	500

Table: Author_Title_Mapping

Author	**Book title**
William Shakespeare	The Comedy of Errors
Markus Winand	SQL Performance Explained
Jeffrey Ullman	A First Course in Database Systems
Jennifer Widom	A First Course in Database Systems