IV C9 Q. Define Normalization

- Normalization is a process of designing a consistent db by minimizing redundancies and ensuring data integrity. through decomposition which is lossless.
- It is a process of organizing data in db in more efficient form. It results in tables that satisfy some constraints and are represented in a simpler manner.
- Normalization ensures: data integrity, prevents redundancy in data and prevents data anomaly.
- There exists 3 forms of Normalization
  (i) 1NF
  (ii) 2 NF
  (iii) 3 NF

(i). 1 NF (First Normal form.)

- simplest form of normalization, simplifies each attribute in relation.
- 1NF states that all attributes in relation must have atomic (indivisible) values and all attribute in a tuple must have a single value from domain of that attribute.
- a relation is in 1NF if every row contains exactly one value for each attribute.
- example: Consider an employee table with columns as shown in diagram:

Emp_ID	Ename	Salary	Ecity
10	Mayank	50 000	Mumbai, Pune
12	Suresh	25 000	Mumbai
15	Sachin	20 000	Pune
18	Jyoti	28 000	Mumbai, Delhi

To convert relational schema in 1NF, the Ecity attribute is divided in atomic domains, it may introduce some data redundancy.

Emp_ID	Ename	Salary	Ecity
10	Mayank	50 000	Mumbai
10	Mayank	50 000	Pune
12	Suresh	25 000	Mumbai
15	Sachin	20 000	Pune
18	Jyoti	28 000	Mumbai
18	Jyoti	28 000	Delhi

- 1NF will solve group redundancy occured in domain values as it allows only a single value from the domain of that attribute.
- It will solve all problems related to domain redundancy.
- Nested relations must be removed to convert relation in 1NF.

(ii) Second Normal Form (2NF)

- 2NF makes use of full functional dependancy and tries to remove problem of redundant data introduced by 1NF decomposition.
- A relation is in 2NF, if it is in 1NF and all non key attributes in relation are fully functional dependent on the primary key of the relation.
- Or a relation is in 2NF, if it is in 1NF and every non-key attribute is fully functionally dependent on the complete primary key of relation (and not depends on part of (partial) primary key).

- example:

Consider an employee table:

- The relational schema not in 2NF is represented as:

Consider an employee table with following FDs,

Emp_ID → Ename, Salary

Emp_ID, Project_ID → Hours, Allowance

As

{Emp_ID, Project_ID} → Ename, Salary, Hours, Allowance

Therefore;

Candidate Key {Emp_ID, Project_ID} is selected as primary key.

Hours & Allowance : fully functionally dpdt on primary key

Ename, Salary: partially dpdt on primary key.

Emp ID	Ename	Salary	P. ID	Hours	Allowance
10	Mahesh	50000	E001	44	40 000
12	Jyoti	25 000	B056	31	30 000
15	Suresh	26 000	C671	23	20 000
18	Mahesh	50 000	E002	12	15 000
15	Suresh	26 000	E001	24	20 000
18	Mahesh	50 000	B056	11	10 000

To normalize above schema to 2NF we can decompose table as,

Employees (Emp_ID, Ename, Salary)
Emp_ID → Ename, Salary.

Emp_ID	Ename	Salary
10	Mahesh	50 000
12	Jyoti	25 000
15	Suresh	26 000
18	Mayank	30 000

Project (Emp_ID, Project_ID, Hours, Allowance).
Emp_ID, Project_ID → Hours, Allowance.

Emp ID	P_ID	Hours	Allowance
10	E001	44	40 000
12	B056	31	30 000
15	C671	23	20 000
18	E002	12	15 000
15	E001	24	20 000
18	B056	11	10 000.