Module-4

Features of good Relational database design

1. Minimum redundancy
2. Fewer NULL values in tuples

Minimum Redundancy

Redundancy: Storage of same data in more than one location.

- A good Relational database design must have Minimum Redundancy

Disadvantage of Redundancy: wastage of storage space.

ISBN	Book_title	P_id	pname	phone
1	C++	P01	Hills publication	1234
2	SQL	P01	Hills publication	1234
3	DBMS	P02	Sunshine Publications	789
4	CA	P02	Sunshine Publications	789
5	OOPS	P03	Bright Publications	567
6	UNIX	P03	Bright Publications	567

Table: Book_publisher relation

Here details of publisher are repeated. If any publisher publishes 100 books, then 100 times details of that publisher is repeated unnecessarily. In addition certain anomalies can appear.

Insertion anomaly: it leads to a situation in which certain information cannot be inserted in to relation unless some other information is stored. For example. Details of new publisher cannot be inserted unless he has not published any book. Similarly information about new book cannot be inserted unless information regarding publisher is not known.

Deletion anomaly:- it leads to a situation in which deletion of certain information resulting in losing values of some other information associated with it.For example tuple with ISBN 5 and 6 is deleted then information about publisher with p_id P03 is lost.

Modification or Update anomaly:- it leads to a situation in which repeated data changed at one place result in inconsistency unless same data is also changed at other place.

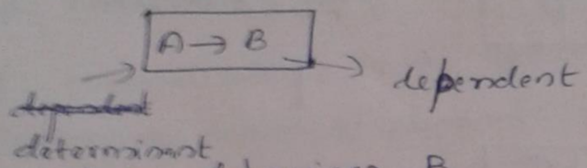
The same information can be expressed on multiple rows; therefore updates to the relation may result in logical inconsistencies. For example updation of p_id of some publisher must applied to multiple records such that wherever that publisher records present.

Fewer NULL values in tuples

- We can use NULL values
- But cannot use NULL values for primary key
- Wastage of storage space
- Leads to problems while taking count() avg () etc.

Functional Dependancy (FD)

Def: A functional dependency A→B in a relation holds if two tuples having same value of attribute A also have same value for attribute B.



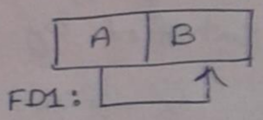
→ A determines B

→ B is functionally dependent on A

Each value of A is associated with precisely one B value.

A	B
1	1
2	4
3	9
4	16
2	4
7	9

This example shows A→B. Each value of A there is associated one and only one value of B.



FD1 : A→B

The following table illustrate that A does not functionally determine B.

A	B
1	1
2	4
3	9
3	10

Since for A=3, there is associated more than one value of B.

A	B	C
1	4	2
3	5	6
3	4	6
7	3	8
9	1	0

$A \rightarrow B$ is false

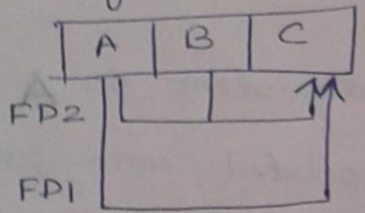
$A \rightarrow C$ is true

$AB \rightarrow C$ is true

FD1 : $A \rightarrow C$

FD2 : $AB \rightarrow C$.

FD diagram



Trivial and nontrivial functional dependency

A FD $A \rightarrow B$ is trivial if $B \subseteq A$

Or

A dependency is said to be trivial, if it is satisfied by all relations.

$X \rightarrow X$

$XY \rightarrow X$

eg:- $A \rightarrow A$, $AB \rightarrow A$ for above example are trivial FD's.

The dependency that are not trivial are called nontrivial functional dependency.

Normalization

Normalization is a database design technique which organizes tables in a manner that reduces redondancy and dependency of data. It divides larger tables to smaller tables and links them using relationships.

Def: Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like insertion, update and deletion anomalies.

First Normal Form (1NF)

- Disallow multivalued, composite and their combinations. Also disallow nested relations.

Def:- A relational schema R is said to be in first Normal form iff the domain of all attributes of R contain atomic (or indivisible) values only.

The following table is an unnormalized relation, bcoz multivalued attribute, content.

Course	Content
Programming	Java, C++
web	HTML, PHP, ASP

Rearrange the relation to convert it to 1NF

course	content
Programming	Java
Programming	C++
Web	HTML
Web	~~ASP~~ PHP
Web	ASP

~~Example/~~

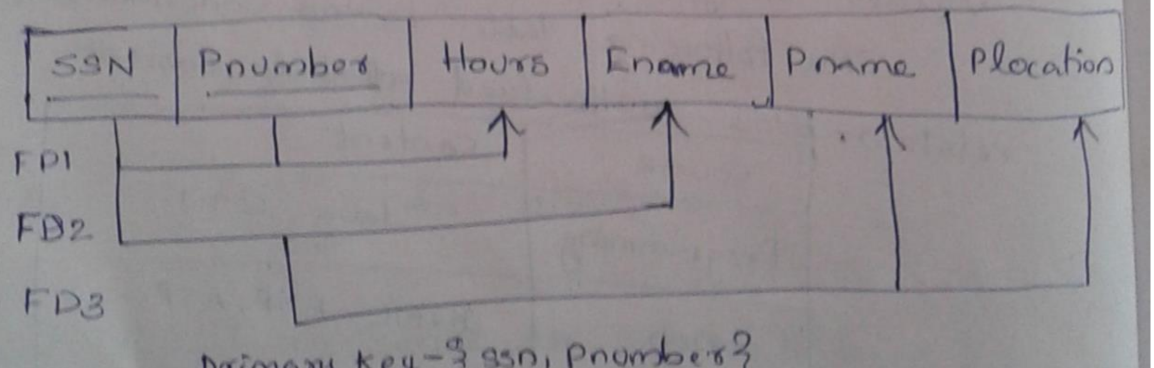
~~EMP_PROJ(SSN, Ename {PROJS(Pnumber, Hours)})~~

Second normal form (2NF)

- Based on the concept of full functional dependency.

A functional dependency X → Y is a full FD, if removal of any attribute A from X means that dependency does not hold any more.

A functional dependency X → Y is partial dependency, if some attribute A∈X, can be removed from X and the dependency still holds.



primary key - {ssn, Pnumber}

FD2 and FD3 violates 2NF condition, bcoz Ename is partialy dependent on key {ssn, Pnumber}. FD3 also violates bcoz {Pnumber} → {Pname, Plocation} is a partial FD.

2NF normalization

table EP1

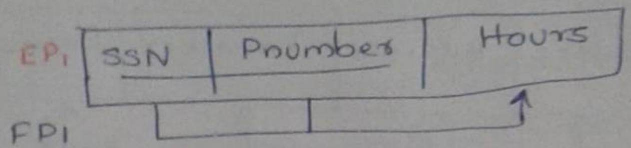


Table EP2

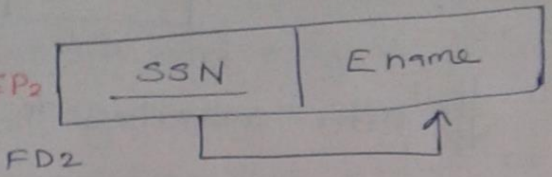
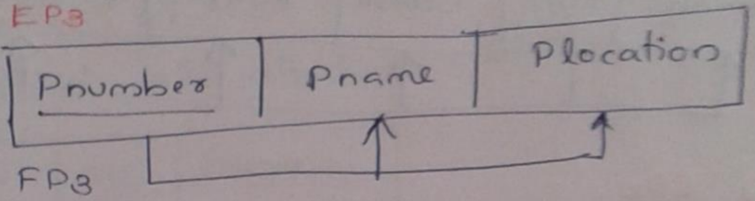


Table EP3



Def: A relational schema, R is said to be in 2NF, it must be in 1NF and every nonprime attribute A in R is fully fonctionally dependent on the primary key of R.

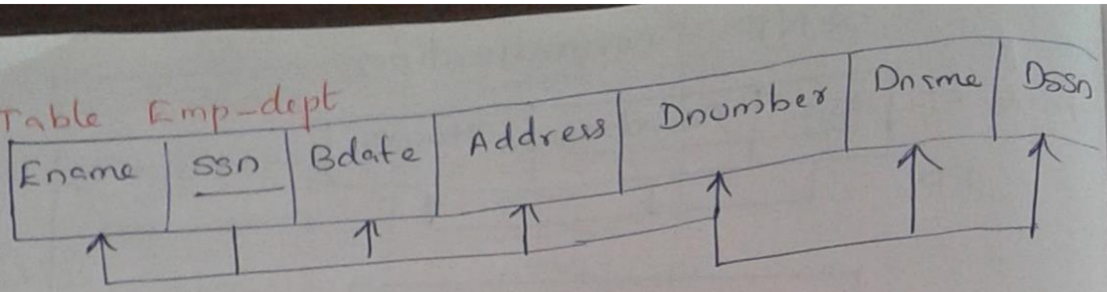
Non-prime attribute means if it is not a part of primary key.

Third normal form (3NF)

- The relation must be in 2NF.
- 3NF based on the concept of transitive dependency.

Def:- A relational schema R is in 3NF if and only if it satisfies 2NF and every non-key attribute is nontransitively dependent on the primary key.

Table Emp-dept



⇓ after converting to 3NF

Table ED1

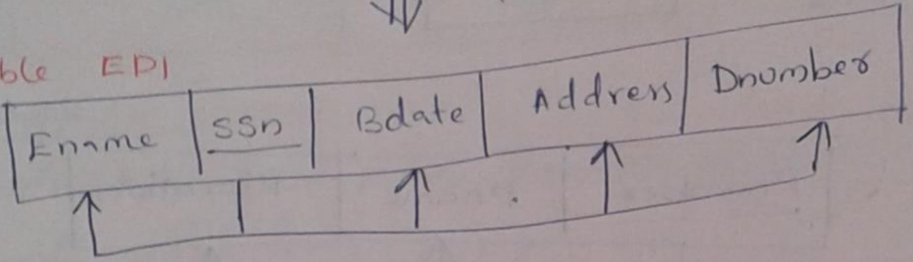
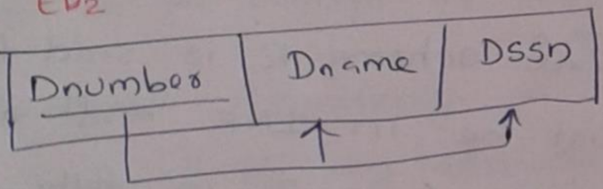


Table ED2



A relational schema R is in 3NF if it is in 2NF and for every FD, X → A either of the following is true.

(1) X is a superkey of R

(2) A is a prime attribute of R.

Boyce Codd Normal Form (BCNF)

A relational schema 'R' is said to be in BCNF, if it satisfies 3NF and an additional constraint that for every FD X → A, X must be a superkey of R.

Multivalued dependency (MVD) ($\twoheadrightarrow$)

In a relational schema 'R' an attribute 'y' is said to be multidependent on attribute x, ($x \twoheadrightarrow y$) if and only if for a particular value of x, the set of value of y is completely determined by the value of x alone and is independent on the value of z, where x, y, z are subset of attribute of R.

ISBN	AID	Phone
123	A1 A2	1234 1256
567	A3	1246 1267 4576 3274

ISBN $\twoheadrightarrow$ AID

AID $\twoheadrightarrow$ Phone

If the MVD $x \twoheadrightarrow y$ is satisfied by all relations on schema R, then $x \twoheadrightarrow y$ is trivial dependency; ie $x \twoheadrightarrow y$ is trivial if $y \subseteq x$ or $y \cup x = R$.

Fourth Normal Form (4NF)

— It is based on the concept of multivalued Functional dependency.

— It must be in 3NF.

~~Defi~~ — Join dependency

Defi — Let 'R' be a relational schema and $R_1, R_2, \ldots R_n$ be the decomposition of R. R is said to be satisfied the join dependency $*(R_1, R_2, \cdots R_n)$ if and only iff

$$\pi_{R_1}(R_1) \bowtie \pi_{R_2}(R_2) \cdots \bowtie \pi_{R_n}(R_n) = R$$

Fifth Normal Form (5NF)

It is also called project join normal form (PJNF). It is based on the concept of join dependency. It must be in 4NF.

Denormalization

The process of normalization is applied to a relation to reduce redundancy. However it results in more relations which increase CPU overhead.

Inorder to speed up the database access, the relations are required to be taken from higher normal form to lower normal form. The process of taking a schema from higher normal form to lower normal form is known as denormalization.

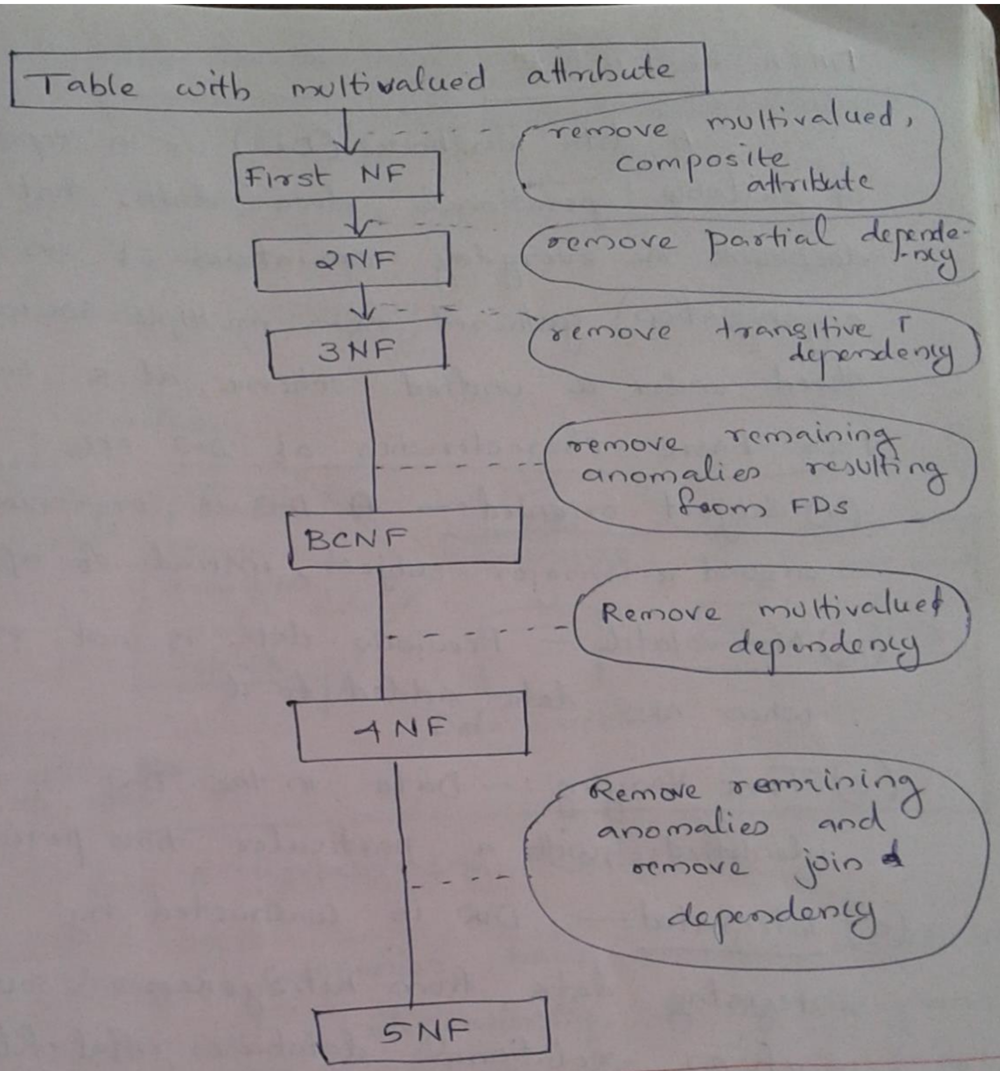
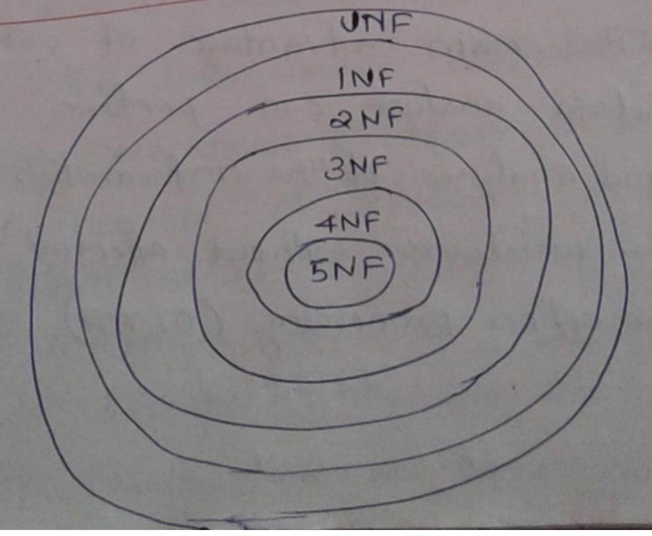


Table with multivalued attribute
remove multivalued, composite attribute
First NF
remove partial dependency
2NF
remove transitive dependency
3NF
remove remaining anomalies resulting from FDs
BCNF
Remove multivalued dependency
4NF
Remove remrining anomalies and remove join dependency
5NF



UNF — Unnormalized relation