Functional Dependency

- The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

$$X \rightarrow Y$$

X is known as determinant.
Y is known as dependent

Employee

ID	Name
1	Ram
2	Sita
3	Megh
4	Garima

We can write,

$$ID \longrightarrow Name$$

We can say that, Name is functionally dependent on ID. OR ID can uniquely identifies the Name.

- Types of Functional dependency :-

i) Trivial FD
ii) Non-trivial FD

Trivial Functional dependency :-

$A \rightarrow B$ has trivial functional dependency if B is a subset of A.

ex— $A \rightarrow A$, $B \rightarrow B$

In table Employee,

$\{ID, Name\} \rightarrow ID$ | It is trivial FD because ID is subset of $\{ID, Name\}$.

Non-trivial Functional Dependency :-

- $A \rightarrow B$ is a non-trivial FD if B is not a subset of A.
- when $A \cap B$ is Null, then $A \rightarrow B$ is called as complete Non-trivial FD.

Ex- $ID \rightarrow Name$
$Name \rightarrow DOB$

Inference Rule (IR)

- The Armstrong's axioms are the basic inference rule. It is used to conclude functional dependencies on a relational database.
- The functional dependencies has six types of Inference rule.

1. Reflexive Rule :-
In the reflexive rule, if Y is a subset of X then X determines Y.

if $\boxed{X \supseteq Y \text{ then } X \rightarrow Y}$

Ex- $X = \{a, b, c, d, e\}$
$Y = \{a, b, c\}$

2. Augmentation Rule :-
- It is called as a partial dependency. In augmentation, if X determines Y, then XZ determines YZ for any Z.

$\boxed{\text{if } X \rightarrow Y \text{ then } XZ \rightarrow YZ}$

Example -
For R (ABCD), if $A \rightarrow B$
then $AC \rightarrow BC$

3. Transitive Rule :-
In this rule, if X determines Y and Y determine Z, then X must also determine Z.

$\boxed{\begin{array}{l} \text{if } X \rightarrow Y \text{ and } Y \rightarrow Z \\ \text{then } X \rightarrow Z \end{array}}$

4. Union Rule :-

It says, if X determines Y and X determines Z, then X must also determine Y and Z.

$$\boxed{\text{if } X \rightarrow Y \text{ and } X \rightarrow Z \text{ then } X \rightarrow YZ}$$

5. Decomposition Rule :-

It is known as Project rule. It is the reverse of Union rule. It says, if X determines Y and Z, then X determines Y and X determines Z separately.

$$\boxed{\text{if } X \rightarrow YZ \text{ then } X \rightarrow Y \text{ and } X \rightarrow Z}$$

6. Pseudo transitive Rule :-

In this, if X determines Y and YZ determines W, then XZ determines W.

$$\boxed{\text{if } X \rightarrow Y \text{ and } YZ \rightarrow W \text{ then } XZ \rightarrow W}$$

Total and Partial Functional Dependency

Partial Functional dependency :-

Partial dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key. An attribute that does not occur in any candidate key is called non-prime attribute.

R (A, B, C, D, E, F)

ABC → DF // It is partial FD's.
B → DF

because
ABC → DF
↓
subset of this
B → DF

Total Functional dependency :-

In any relation, there exists Full FD's between any two attributes X and Y, when X is functionally dependent on Y and is not functionally dependent on any proper subset of Y.

R (A, B, C, D, E, F)

- ABC → DF
  B → DF // total FD's because there is no subset of B.

- ABC → DF // Total FD's
  CE → DF // Partial FD's
  E → DF

NORMALIZATION

- It 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 characteristics like insertion, update and deletion anomalies.
- It divides the larger table into the smaller table and links them using relationship.

Anomalies in DBMS :-

There are three types of anomalies that occur when the database is not normalized.

Employee

emp_id	emp_name	emp_address	emp_dept
101	Rick	Delhi	D001
101	Rick	Delhi	D002
123	Maggie	Agra	D890
166	Glenn	Chennai	D900
166	Glenn	Chennai	D004

Update anomaly :- In the above table, we have two rows of 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.

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 does not 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.

- Here, are the most commonly used normal forms :—
  - i) First normal form (1 NF)
  - ii) Second normal form (2 NF)
  - iii) Third normal form (3 NF)
  - iv) Boyce & codd normal form (BCNF)

First Normal Form :—

- A relation will be 1 NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
- It disallows the multi-valued attribute, composite attribute & their combinations.

Employee

emp-id	emp-name	emp-Phone	emp-State
14	John	7272826385, 9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389 8589830302	Punjab

∴ It is not in 1 NF because of multi-valued attribute emp-Phone.

The decomposition of the Employee table into 1 NF has been shown below :-

emp_id	emp_name	emp_phone	emp_state
14	John	7272826385	UP
14	John	9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389	Punjab
12	Sam	8589830302	Punjab

Second Normal Form (2 NF) :—

- In the 2 NF, relational must be in 1 NF.
- In the second normal form, all non-key attributes are fully functional dependent on the primary key.

Ex – let's assume, a school can store the data of teachers & the subjects they teach. In a school, a teacher can teach more than one subject.

Teacher

Id	Subject	Age
25	chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	computer	38

In the given table, non-prime attribute Age is dependent on Id which is proper subset of a candidate key. That's why it violates the rule for 2NF. To convert the given table into 2NF, we decompose it into two tables.

Teacher_Detail

Id	Age
25	30
47	35
83	38

Teacher_Subject

Id	subject
25	Chemistry
25	Biology
47	English
83	Math
83	Computer

Third Normal Form (3NF) :—

- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in 3NF.
- A relation is in 3NF, if it holds atleast one of the following conditions for every non-trivial functional dependency $X \rightarrow Y$.
  i) X is a super key.
  ii) Y is a prime attribute, i.e, each element of Y is part of some candidate key.

Example - Employee_Detail

ID	Name	zip	state	city
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston
444	Lan	60007	US	chicago
555	Katharine	06389	UK	Norwich
666	John	462007	MP	Bhophal

Super key in above table

{ ID }, { ID, Name }, { ID, Name, zip }, ... so on.

candidate key - { ID }

Boyce Codd Normal Form (BCNF)

- It is advance version of 3NF. It is stricter than 3NF
- A table is in BCNF if every FD X → Y, X is the super key of the table.
- For BCNF, the table should be in 3NF and for every FD, LHS is super key.

Example - let's assume there is a company where employees work in more than one department

Employee table

Emp_id	emp_country	emp_dept	Dept_type	Emp_Dept_no.
264	India	Designing	D394	283
264	India	Testing	D394	300
364	UK	Stores	D283	232
364	UK	Developing	D283	549

In the above table, FD are as follows:

Emp_id → emp_country

emp_dept → { dept_type, Emp_Dept_No

candidate key : { Emp_id, emp_dept

The table is not in BCNF because neither Emp_Dept nor Emp_id alone are keys.

To convert the given table into BCNF, we decompose it into three tables.

i) Emp_country table

ii) Emp_Dept table

iii) Emp_Dept_Mapping table

Emp_country table

Emp_id	Emp_country
264	India
264	India UK

Emp_Dept table

Emp_Dept	Dept_Type	Emp_Dept_No
Designing	D394	283
Testing	D394	300
stores	D283	232
Developing	D283	549

Emp_Dept_Mapping table

Emp_id	Emp_Dept
D394	283
D394	300
D283	232
D283	549

FD's are

Emp_id → Emp_country

Emp_Dept → {Dept_type, Emp_Dept_No}

Candidate key :-

For 1st table → Emp_id

For 2nd table → Emp_Dept

For 3rd table → {Emp_id, Emp_Dept}

Now, this is in BCNF because left side part of both the FD's are key.

Fourth Normal Form (4-NF)

- A relation will be in 4 NF, if it is in BCNF and has no multi-valued dependency.
- For a dependency A → B, if for a single value of A, multiple values of B exists, then the relation will be multi-valued dependency.

Example - Student

Stu-id	Course	Hobby
21	Computer	Dancing
21	Math	Singing
34	chemistry	Dancing
74	Biology	cricket
59	physics	Hockey

The given student table is in 3NF, but the course and Hobby are two independent entity. Hence, there is no relationship between course and Hobby.

In the student table, a student with stu-id, 21 contains two courses, computer and Math and two hobbies, Dancing & singing. So there is a multi-valued dependency on stu-id, which leads to unnecessary repetition of data.

So, to make the above table into 4 NF, we can decompose it into two tables.

Student-Course

Stu-id	Course
21	computer
21	Math
34	chemistry
74	Biology
59	Physics

Student_Hobby

Stu-id	Hobby
21	Dancing
21	Singing
34	Dancing
74	Cricket
59	Hockey

Inclusion Dependency

- Multivalued dependency and Join dependency can be used to guide database design although they both are less common than functional dependencies.
- Inclusion dependencies are quite common. They typically show little influence on designing of the database.
- The inclusion dependency is a statement in which some columns of a relation are contained in other columns.
- The example of Inclusion dependency is a Foreign key. In one relation, the referring relation is contained in the primary key column of the referenced relation.
- Suppose we have two relations R and S which was obtained by translating two entity sets such that every R entity is also an S entity.
- Inclusion dependency would be happen if projecting R on its key attributes yields a relation that is contained in the relation obtained by projecting S on its key attributes.
- In inclusion dependency, we should not split groups of attributes that participate in an inclusion dependency.
- In Practice, most inclusion dependency are key-based that is involved only keys.

Lecture No-28

Relational Decomposition

- when a relation in the relational model is not in appropriate normal form then the decomposition of a relation is required
- If the relation has no proper decomposition, then it may lead to problems like loss of information.
- Decomposition is used to eliminate some of the problems of bad design like anomalies, inconsistencies and redundancy.
- Types of decomposition —
  - i) lossless decomposition
  - ii) dependency Preserving

Lossless decomposition :—

- If the information is not lost from the relation that is decomposed, then the decomposition will be lossless.
- The lossless decomposition guarantees that the join of relations will result in the same relation as it was decomposed.
- The relation is said to be lossless decomposition if natural joins of all the decomposition give the original relation.
- Example - Employee_Department

emp-id	e_Name	e_Age	e_city	Dept_id	Dept_Name
22	Denim	28	Mumbai	827	Sales
33	Alina	25	Delhi	438	Marketing
46	Stephan	30	Banglore	869	Finance
52	Katherine	36	Mumbai	575	Production
60	Jack	40	Noida	678	Testing

This table is decomposed into two relations - Employee & Department

Employee table

Emp-id	e_name	e_Age	e_city
22	Denim	28	Mumbai
33	Alina	25	Delhi
46	Stephan	30	Banglore
52	Katherine	36	Mumbai
60	Jack	40	Noida

Department table

Dept_id	emp-id	Dept_Name
827	22	Sales
438	33	Marketing
869	46	Finance
575	52	Production
678	60	Testing

Now, when these two relations are joined on the common column 'Emp_id' then the resultant relation look like Employee_Department table.
Hence, the decomposition is lossless join decomposition.

Dependency Preserving :-

- It is an important constraint of the database
- In the dependency preservation, at least one decomposed table must satisfy every dependency.
- If a relation R is decomposed into relation R_1 and R_2, then the dependencies of R either must be a part of R_1 or R_2 or must be derivable from the combination of FD's of R_1 and R_2.

- Ex, Suppose there is a relation R (A, B, C, D) with

  FD set $A \rightarrow BC$

  The relational R is decomposed into R_1 (ABC) and R_2 (AD) which is dependency preserving because FD $A \rightarrow BC$ is a part of relation R_1(ABC).

Lecture No - 34

Lecture No - 20

Multivalued Dependency

- Multivalued dependency occurs when two attributes in a table are independent of each other but, both depend on a third attribute.
- It consists of atleast two attributes that are dependent on a third attribute, that's why it always requires at least three attributes.

Example - Suppose there is a bike manufacturer company which produces two colors (white and black) of each model every year.

Bike_Model	Manuf_year	Color
M2011	2008	White
M2001	2008	Black
M3001	2013	White
M3001	2013	Black
M4006	2017	White
M4006	2017	Black

Here, Columns Color and Manuf_year are dependent on Bike_Model and independent of each other.

In this case, these two columns can be called as multivalued dependent on Bike_Model. The representation of these dependencies is show below -

Bike_Model ⟶ Manuf_Year

Bike_Model ⟶ Color

This can be read as "Bike_Model" multidetermined "Manuf_Year" and "Bike_Model" multidetermined "Color".

Lecture No - 37

Lecture No - 31

JOIN Dependency

- If a table can be recreated by joining multiple tables and each of this table have a subset of the attributes of the table, then the table is in join dependency.
- It is a generalization of Multivalued dependency.
- Join dependency can be related to 5NF, where, a relation is in 5NF, only if it is already in 4NF and it cannot be decomposed further.

Example -

Employee

E_Name	E_Skills	E_Job
Tom	Networking	EJ001
Harry	web development	EJ002
Katie	Programming	EJ002

This table can be decomposed into the following three tables; therefore it is not in 5NF.

Employee Skills

E_Name	E_Skills
Tom	Networking
Harry	web development
Katie	Programming

EmployeeJob

E_Name	E_Job
Tom	EJ001
Harry	EJ002
Katie	EJ002

JobSkills

E-Skills	E_Job
Networking	EJ001
web dev.	EJ002
Programming	EJ002

Our Join dependency :-

{(E_Name, E_Skills), (E_Name, E_Job), (E_Skills, E_Job)}

The above relations have Join dependency. So they are not in 5NF. That would mean that a join relation of the above three relations is equal to our original relation <Employee>.