Course code : **CSE2007**
Course title : **Database Management System**
Module : **3**
Topic : **7.1**

Functional Dependency and Normalization

Objectives

This session will give the knowledge about

- Functional Dependency
- Normalization

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.

A column Y of a relational table is said to be functionally dependent upon column X when values of column Y are uniquely identified by values of column X.

$$X \rightarrow Y$$

The left side of FD is known as a determinant, the right side of the production is known as a dependent.

Example:

$$Emp\_Id \rightarrow Emp\_Name$$

Functional Dependency

If the information stored in a table can uniquely determine another information in the same table, then it is called Functional Dependency. Consider it as an association between two attributes of the same relation.

example: <Employee>

EmpID	**EmpName**	**EmpAge**
E01	Amit	28
E02	Rohit	31

In the above table, EmpName is functionally dependent on EmpID because EmpName can take only one value for the given value of EmpID:

EmpID -> EmpName

Fully Functional Dependency

An attribute is fully functional dependent on another attribute, if it is Functionally Dependent on that attribute and not on any of its proper subset.

For example, an attribute Q is fully functional dependent on another attribute P, if it is Functionally Dependent on P and not on any of the proper subset of P.

<EmployeeProject>

EmpID	ProjectID	Days
E099	001	320
E056	002	190

{EmpID, ProjectID} -> (Days)

Partial Dependency

Partial Dependency occurs when a nonprime attribute is functionally dependent on part of a candidate key. Let us see an example:

StudentID	ProjectNo	StudentName	ProjectName
S01	199	Katie	Geo Location
S02	120	Ollie	Cluster Exploration

- The prime key attributes are StudentID and ProjectNo.
- As stated, the non-prime attributes i.e. StudentName and ProjectName should be functionally dependent on part of a candidate key, to be Partial Dependent.
- The StudentName can be determined by StudentID that makes the relation Partial Dependent.
- The ProjectName can be determined by ProjectID, which that the relation Partial Dependent.

Types of Functional Dependency

Trivial functional dependency

- $A \rightarrow B$ has trivial functional dependency if B is a subset of A.
- The following dependencies are also trivial like: $A \rightarrow A$, $B \rightarrow B$

Example:

{Employee_id, Employee_Name} $\rightarrow$ Employee_Id is a trivial functional dependency as Employee_Id is a subset of {Employee_Id, Employee_Name}.

Non-trivial functional dependency

- $A \rightarrow B$ has a non-trivial functional dependency if B is not a subset of A.
- When A intersection B is NULL, then $A \rightarrow B$ is called as complete non-trivial.

Example:

ID $\rightarrow$ Name,

Name $\rightarrow$ DOB

Inference Rule (IR)

- The inference rule is a type of assertion. It can apply to a set of FD(functional dependency) to derive other FD.
- Using the inference rule, we can derive additional functional dependency from the initial set.

The Functional dependency has 6 types of inference rule:

Rule 1. Reflexive Rule (IR1)

In the reflexive rule, if Y is a subset of X, then X determines Y.

If $X \supseteq Y$ then $X \rightarrow Y$

Example:

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

$Y = \{a, b, c\}$

Inference Rule (IR)

Rule 2. Augmentation Rule (IR2)

- The augmentation is also called as a partial dependency. In augmentation, if X determines Y, then XZ determines YZ for any Z.
- If $X \rightarrow Y$ then $XZ \rightarrow YZ$

Example:

For R(A B C D), if $A \rightarrow B$ then $AC \rightarrow BC$

Rule 3. Transitive Rule (IR3)

- In the transitive rule, if X determines Y and Y determine Z, then X must also determine Z.

If $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

Inference Rule (IR)

Rule 4. Union Rule (IR4)

- Union rule says, if X determines Y and X determines Z, then X must also determine Y and Z.
- If X $\rightarrow$ Y and X $\rightarrow$ Z then X $\rightarrow$ YZ

Rule 5. Decomposition Rule (IR5)

- Decomposition rule is also known as project rule. It is the reverse of union rule.
- This Rule says, if X determines Y and Z, then X determines Y and X determines Z separately.
- If X $\rightarrow$ YZ then X $\rightarrow$ Y and X $\rightarrow$ Z

Inference Rule (IR)

Rule 6. Pseudo transitive Rule (IR6)

- In Pseudo transitive Rule, if X determines Y and YZ determines W, then XZ determines W.
- If $X \rightarrow Y$ and $YZ \rightarrow W$ then $XZ \rightarrow W$

Decomposition

A functional decomposition is the process of breaking down the functions of an organization into progressively greater (finer and finer) levels of detail.

Decomposition helps in eliminating some of the problems of bad design such as redundancy, inconsistencies and anomalies.

There are two types of decomposition :

- Lossy Decomposition
- Lossless Join Decomposition

Lossy Decomposition

"The decompositio of relation R into R1 and R2 is lossy when the join of R1 and R2 does not yield the same relation as in R."

Roll_no	Sname	Dept
111	parimal	COMPUTER
222	parimal	ELECTRICAL

Roll_no	Sname
111	parimal
222	parimal

Sname	Dept
parimal	COMPUTER
parimal	ELECTRICAL

Roll_no	Sname	Dept
111	parimal	COMPUTER
111	parimal	ELECTRICAL
222	parimal	COMPUTER
222	parimal	ELECTRICAL

Lossless Join Decomposition

"The decompositio of relation R into R1 and R2 is lossless when the join of R1 and R2 yield the same relation as in R."

Roll_no	Sname	Dept
111	parimal	COMPUTER
222	parimal	ELECTRICAL

Roll_no	Sname
111	parimal
222	parimal

Roll_no	Dept
111	COMPUTER
222	ELECTRICAL

Roll_no	Sname	Dept
111	parimal	COMPUTER
222	parimal	ELECTRICAL

Normalization

- Normalization is the process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations

- 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 characteristics like Insertion, Update and Deletion Anomalies.

- Normalization divides the larger table into the smaller table and links them using relationship.

- The normal form is used to reduce redundancy from the database table.

First Normal Form (1NF)

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

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

First Normal Form (1NF)

Before Normalization

roll_no	name	subject
101	Akon	OS, CN
103	Ckon	Java
102	Bkon	C, C++

After 1NF

roll_no	name	subject
101	Akon	OS
101	Akon	CN
103	Ckon	Java
102	Bkon	C
102	Bkon	C++

First Normal Form (1NF)

Before Normalization

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385, 9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389, 8589830302	Punjab

After 1NF

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 (2NF)

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

TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

Second Normal Form (2NF)

TEACHER_DETAIL

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

TEACHER_SUBJECT

TEACHER_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 third normal form.

A relation is in third normal form if it holds atleast one of the following conditions for every non-trivial function dependency $X \rightarrow Y$.

- X is a super key.
- Y is a prime attribute, i.e., each element of Y is part of some candidate key

Third Normal Form (3NF)

Resultant table of 2NF

emp_id	emp_name	emp_zip	emp_state	emp_city	emp_district
1001	John	282005	UP	Agra	Dayal Bagh
1002	Ajeet	222008	TN	Chennai	M-City
1006	Lora	282007	TN	Chennai	Urrapakkam
1101	Lilly	292008	UK	Pauri	Bhagwan
1201	Steve	222999	MP	Gwalior	Ratan

Here, emp_state, emp_city & emp_district dependent on emp_zip. And, emp_zip is dependent on emp_id that makes non-prime attributes (emp_state, emp_city & emp_district) transitively dependent on super key (emp_id). This violates the rule of 3NF.

Third Normal Form (3NF)

Resultant table of 3NF

emp_id	emp_name	emp_zip
1001	John	282005
1002	Ajeet	222008
1006	Lora	282007
1101	Lilly	292008
1201	Steve	222999

emp_zip	emp_state	emp_city	emp_district
282005	UP	Agra	Dayal Bagh
222008	TN	Chennai	M-City
282007	TN	Chennai	Urrapakkam
292008	UK	Pauri	Bhagwan
222999	MP	Gwalior	Ratan

Third Normal Form (3NF)

Resultant table of 2NF

s#	city	status
s1	London	20
s2	Paris	10
s3	Tokyo	30
s4	Paris	10

Result of 3NF

s#	city
s1	London
s2	Paris
s3	Tokyo
s4	Paris

city	status
London	20
Paris	10
Tokyo	30
Rome	50

The table supplier is in 2NF but not in 3NF because it contains a ***transitive dependency***

- SUPPLIER.s# —> SUPPLIER.city
- SUPPLIER.city —> SUPPLIER.status
- SUPPLIER.s# —> SUPPLIER.status

Boyce Codd normal form (BCNF)

- BCNF is the advance version of 3NF. It is stricter than 3NF.
- A table is in BCNF if every functional dependency 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.

Consider the following relationship : **R (A,B,C,D)**

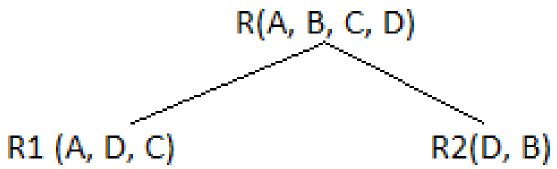
and following dependencies :

A -> BCD
BC -> AD
D -> B

Above relationship is already in 3rd NF. Keys are **A** and **BC.**

Hence, in the functional dependency, **A -> BCD**, A is the super key.
in second relation, **BC -> AD**, BC is also a key.
but in, **D -> B**, D is not a key.

Hence we can break our relationship R into two relationships **R1** and **R2.**



Breaking, table into two tables, one with A, D and C while the other with D and B.

Boyce Codd normal form (BCNF)

Resultant table of 3NF

emp_id	emp_nationality	emp_dept	dept_type	dept_no_of_emp
1001	Austrian	Production and planning	D001	200
1001	Austrian	stores	D001	250
1002	American	Technical support	D134	100
1002	American	Purchasing department	D134	600

- Functional dependencies in the table above:
  emp_id -> emp_nationality
  emp_dept -> {dept_type, dept_no_of_emp}
- Candidate key: {emp_id, emp_dept}
- The table is not in BCNF as neither emp_id nor emp_dept alone are keys.

Boyce Codd normal form (BCNF)

Resultant table of BCNF

emp_id	emp_nationality
1001	Austrian
1002	American

emp_dept	dept_type	dept_no_of_emp
Production and planning	D001	200
stores	D001	250
Technical support	D134	100
Purchasing department	D134	600

emp_id	emp_dept
1001	Production and planning
1001	stores
1002	Technical support
1002	Purchasing department

Multivalued Dependency

- Multivalued dependency occurs when two attributes in a table are independent of each other but, both depend on a third attribute.
- A multivalued dependency consists of at least two attributes that are dependent on a third attribute that's why it always requires at least three attributes.
- Here columns COLOR and MANUF_YEAR are dependent on BIKE_MODEL and independent of each other.

BIKE_MODEL	MANUF_YEAR	COLOR
M2011	2008	White
M2001	2008	Black
M3001	2013	White
M3001	2013	Black
M4006	2017	White
M4006	2017	Black

Fourth normal form (4NF)

- A relation will be in 4NF if it is in Boyce Codd normal form 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 a multi-valued dependency.

STU_ID	COURSE	HOBBY
21	Computer	Dancing
21	Math	Singing
34	Chemistry	Dancing
74	Biology	Cricket
59	Physics	Hockey

Fourth normal form (4NF)

In the STUDENT relation, a student with STU_ID, 21 contains two courses, Computer and Math and two hobbies, Dancing and Singing. So there is a Multi-valued dependency on STU_ID, which leads to unnecessary repetition of data.

Resultant 4NF

STU_ID	COURSE
21	Computer
21	Math
34	Chemistry
74	Biology
59	Physics

STU_ID	HOBBY
21	Dancing
21	Singing
34	Dancing
74	Cricket
59	Hockey

Fifth normal form (5NF)

- A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.
- 5NF is satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.
- 5NF is also known as Project-join normal form (PJ/NF).

SUBJECT	LECTURER	SEMESTER
Computer	Anshika	Semester 1
Computer	John	Semester 1
Math	John	Semester 1
Math	Akash	Semester 2
Chemistry	Praveen	Semester 1

Fifth normal form (5NF)

In the above table, John takes both Computer and Math class for Semester 1 but he doesn't take Math class for Semester 2. In this case, combination of all these fields required to identify a valid data.

Suppose we add a new Semester as Semester 3 but do not know about the subject and who will be taking that subject so we leave Lecturer and Subject as NULL. But all three columns together acts as a primary key, so we can't leave other two columns blank.

So to make the above table into 5NF, we can decompose it into three relations P1, P2 & P3:

Fifth normal form (5NF)

Resultant 5NF

SUBJECT	LECTURER
Computer	Anshika
Computer	John
Math	John
Math	Akash
Chemistry	Praveen

SUBJECT	SEMESTER
Computer	Semester 1
Computer	Semester 1
Math	Semester 1
Math	Semester 2
Chemistry	Semester 1

LECTURER	SEMESTER
Anshika	Semester 1
John	Semester 1
John	Semester 1
Akash	Semester 2
Praveen	Semester 1

Summary

This session will give the knowledge about

- Functional Dependency
- Normalization