Unit IV

Chapter - 10 : Relational Database Design

Q.1 State and explain Armstrong's axioms and its property. SPPU - May 18), 4 Marks

Ans. :

FD Properties (Armstrong's Axioms/ Closures of FD)

FD Properties (Armstrong's Axioms/ Closures of FD)

- Given that Relation R(X, Y, Z, W); represents a table R with set of indivisible attributes X, Y, Z and W.
- It is possible to derive many properties of functional dependencies.
- Axioms are nothing but rules of inference which provides a simple technique for reasoning about functional dependencies.

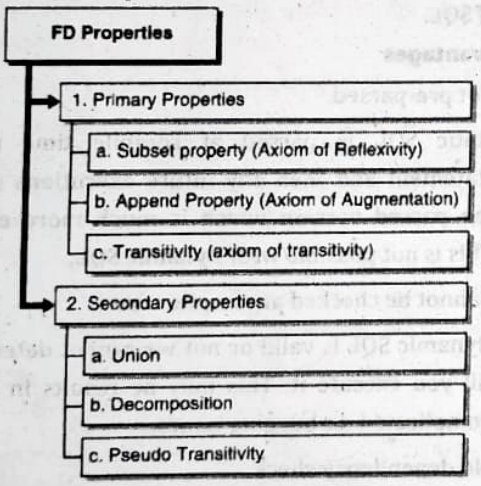
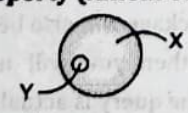


Fig. 10.1 : FD Properties

(1) Primary Properties

a. Subset property (Axiom of Reflexivity)



For given relation R(X, Y, Z, W),

If Y is a subset of X as shown in diagram,

Then $X \rightarrow Y$

(Which can be referred as X is functionally dependent on Y)

b. Append Property (Axiom of Augmentation)

For given relation R(X, Y, Z, W),

If $X \rightarrow Y$

Then $XZ \rightarrow YZ$

It is possible to append attribute Z to both sides of FD provided that it is part of same table.

c. Transitivity (axiom of transitivity)

For given relation R(X, Y, Z, W),

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

Then $X \rightarrow Z$

It is possible to use transitivity if attribute X, Y and Z are part of the same table.

(2) Secondary Properties

a. Union

For given relation R(X, Y, Z, W),

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

Then $X \rightarrow YZ$

b. Decomposition

For given relation R(X, Y, Z, W),

If $X \rightarrow YZ$

Then $X \rightarrow Y$ and $X \rightarrow Z$

c. Pseudo Transitivity

For given relation R(X, Y, Z, W),

If $X \rightarrow Y$ and $YZ \rightarrow W$

Q.2 What is lossless decomposition? SPPU - Dec. 17, 4 Marks

Ans. : Lossless-join Decomposition

- It is clear that decomposition must be lossless so that we do not lose any information from the relation that is decomposed.
- Lossless join decomposition ensures that we can never get the situation where spurious tuple are generated in relation, for every value on the join attributes there will be a unique tuple in one of the relations. For above join to become lossless we need to go for following steps.

Steps :

(i) Let R_1 and R_2 form decomposition of relation R as R_1 and R_2 are both sets of attributes from R.

(ii) Decompose the relation schema Department-Student into

Department-schema = (Dept_Id, Dname)

Student-schema = (Stud_id, Sname, Location)

(iii) The attributes in common must be a key for one of the relation for decomposition to be lossless. $R_1 \cap R_2 \neq \Phi$ There must not be null.

Note : You are joining a primary key and a foreign key of table.

Example :

Above relation can be lossless decomposed as follows,

Schema 1 : Department schema contains (Dept_Id, Dname)

$\therefore R_1 \leftarrow \Pi_{Dept\_Id,Dname}$ (Department_student)

Dept_Id	Dname
10	Development
20	Teaching
30	HR

Schema 2 : Student schema contains (Stud_Id, Dept_Id, Sname, Location)

$\therefore R_2 \leftarrow \Pi_{Stud\_Id, Dept\_Id, Location,Sname}$ (Department_student)

Stud_Id	Dept_Id	Location	Sname
1	10	Mahim	Sushant
2	20	Vashi	Snehal
3	30	Worli	Pratiksha
4	20	Dadar	Supraja

This is lossless-join decomposition as $R_1 \cap R_2 \neq \Phi$ common column is Dept_Id

Q.3 What is Normalization ? SPPU - May 19. 2 Marks

Ans. : Normalization

Introduction

- Normalization is a step by step decomposition of complex records into simple records.
- Normalization is a process of organizing data in database in more efficient form. It results in tables that satisfy some constraints and are represented in a simple manner.
- This process is also called as canonical synthesis.
- Normalization is a step by step decomposition and database designers may not normalize relation to the highest possible normal form.
- The relations may be left in a lower normal form like 2NF, which may cause some penalties like data anomalies.

Definition

***Normalization** is a process of designing a consistent database by minimizing redundancy and ensuring data integrity through decomposition which is lossless.*

Q.4 State and explain 2NF. SPPU - May 19. 6 Marks

Ans. : Second Normal Form (2NF)

1. Introduction

- This normal form makes use of full functional dependency and tries to remove problem of redundant data that was introduced by 1NF decomposition.
- Therefore before applying 2NF to a relation, it needs to satisfy 1NF condition.

2. Definition

- A relation is in 2NF, if it is in 1NF and all non-key attributes in relation are fully functionally 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).
  - In short 2NF means,
  - It should be in 1NF.
- There should not be any partial dependency on primary key attributes.
- 2NF prohibits partial dependencies

3. Steps

(a) Find and remove attributes that are related to only a part of the key or not related to key.

(b) Group the removed attributes in another table.

(c) Assign the new table a key that consists of that part of the old composite key.

(d) If a relation is not in 2NF, it can be further normalized into a number of 2NF relations.

4. Example:

- Consider an employee table with columns as shown in diagram,
- The relational schema **not in 2 NF** is represented as,

Consider an Employee table with following FDs,

Employee_Id → Ename, Salary

Employee_Id, Project_Id → Hours, Allowance

As

{Employee_Id, Project_Id} → Ename, Salary, Hours, Allowance

Therefore,

Candidate key {Employee_Id, Project_Id} is selected as primary key.

- As attributes Hours, Allowance of employee table are full functionally dependent on primary key whereas attributes Ename and Salary are partially depends on primary key. (As Ename, Salary are depends **on part of primary key**)
- The state of Employee relational schema is,

Table 10.2 : Non-2NF Employee Table

Eid	Ename	Salary	Project_Id	Hours	Allowance
10	Mahesh	50000	E001	44	40000
12	Suresh	25000	B056	31	30000
15	Ganesh	26000	C671	23	20000
18	Mahesh	50000	E002	12	15000
15	Ganesh	26000	E001	24	20000
18	Mahesh	50000	B056	11	10000

- To normalize above schema to 2NF we can decompose tables as,
- Employee (Employee_Id, Ename, Salary)
- Employee_Id → Ename, Salary

Table 10.3 : 2NF Employee Table

Employee_Id	Ename	Salary
10	Mahesh	50000
12	Suresh	25000
15	Ganesh	26000
18	Mahesh	50000

- Project (Employee_Id, Project_Id, Hours, Allowance)
- Employee_Id, Project_Id → Hours, Allowance

Table 10.4 : 2NF Project Table

Employee_Id	Project_Id	Hours	Allowance
10	E001	44	40000
12	B056	31	30000
15	C671	23	20000
18	E002	12	15000
15	E001	24	20000
18	B056	11	10000

- Consider, Relation R(A, B, C, D, E, F) and the FDs as below,

A → BC, B → DC, D → EF

(i) The candidate Key is {AD} → {A, D, B, C, E, F} selected as primary key.

All attributes are partially dependent on primary key.

Hence, Relation R is **not in 2NF.**

(ii) The 2NF Relation Schema is,

R1 (A, B, C, D) with FDs A → BC, B → DC

R2 (D, E, F) with FDs D → EF

5. Minimizing Tuple Redundancy

- The second normal form will avoid same tuples to be repeated in a table as it forces all non-key attributes must be full functionally depends on primary key of a relation.
- 2NF will create a new table for each partial key with all its dependent attributes.

Example:

- Let us consider the table we obtained after first normalization.

Sr. No.	Faculty code	Faculty name	Date of birth	Subject	Hours
1	100	Yogesh	17/07/64	DSA	16
2	100	Yogesh	17/07/64	SS	8
3	100	Yogesh	17/07/64	IS	12
4	101	Amit	24/12/72	MIS	16
5	101	Amit	24/12/72	PM	8
6	101	Amit	24/12/72	IS	12
7	102	Omprakash	03/02/80	PWRC	8
8	102	Omprakash	03/02/80	PCOM	8
9	102	Omprakash	03/02/80	IP	16
10	103	Nitin	28/11/66	DT	10
11	103	Nitin	28/11/66	PCOM	8
12	103	Nitin	28/11/66	SS	8
13	104	Mahesh	01/01/86	DT	10
14	104	Mahesh	01/01/86	ADBMS	8
15	104	Mahesh	01/01/86	PWRC	8

- While eliminating the repeating groups, we have introduced redundancy into table. Faculty code, Name and Date of birth are repeated since the same faculty is multi skilled.
- To eliminate this, let us split the table into 2 parts; one with the non-repeating groups and the other for repeating groups.

Faculty

Faculty code	Faculty Name	Date of birth
100	Yogesh	17/07/64
101	Amit	24/12/72
102	Omprakash	03/02/80
103	Nitin	28/11/66
104	Mahesh	01/01/86

Faculty_code → Faculty_name, Date_of_Birth

The other table is those with repeating groups.

Subject

Table 10.5

Sr. No	Faculty code	Subject	Hours
1	100	DSA	16
2	100	SS	8
3	100	IS	12
4	101	MIS	16
5	101	PM	8
6	101	IS	12
7	102	PWRC	8
8	102	PCOM	8
9	102	IP	16
10	103	DT	10
11	103	PCOM	8
12	103	SS	8
13	104	DT	10
14	104	ADBMS	8
15	104	PWRC	8

- Faculty code is the only key to identify the faculty name and the date of birth. Hence, Faculty code is the primary key in the first table and foreign key in the second table.
- Faculty code is repeated in the Subject table. Hence, we have to take into account the 'SNO' to form a composite key in Subject table. Now, SNO + Faculty code can unique identity each row in this Table 10.4.

Hence, the relation is now in Second Normal form.

6. Anomalies (Problems)

(a) Insertion

- New record needed to insert in both tables.
- Inserting the records of various Faculties teaching same subject would result the redundancy of hours information.

(b) Updation

- Updating some record in faculty table may exist in subject table.

- For a subject, the number of hours allotted to a subject is repeated several times. Hence, if the number of hours has to be changed, this change will have to be recorded in every instance of that subject. Any omissions will lead to inconsistencies.

(c) Deletion

- If a faculty leaves the organization, information regarding hours allotted to the subject is also needed to be deleted from subject table.
- Hence, This Subject table should therefore be further decomposed without any loss of information in 3rd normal form.

Q.5 State and explain 3 NF. SPPU - May 19

Ans. : Third Normal Form (3NF)

1. Introduction

- This normal form used to minimize the transitive redundancy.
- In order to remove the anomalies that arose in Second Normal Form and to remove transitive dependencies, if any, we have to perform third normalization.

2. Definition

- A relation is in 3NF, if it is in 2NF and no non-key attribute of the relation is transitively dependent on the primary key.
- 3NF prohibits transitive dependencies.
- In short 2NF means,
  1. It should be in 2 NF.
  2. There should not be any transitive partial dependency.

3. Example:

Now let us see how to normalize the second table obtained after 2NF.

Subject

Table 10.5

Sr. No	Faculty code	Subject	Hours
1	100	DSA	16
2	100	SS	8
3	100	IS	12
4	101	MIS	16
5	101	PM	8
6	101	IS	12
7	102	PWRC	8
8	102	PCOM	8
9	102	IP	16
10	103	DT	10
11	103	PCOM	8
12	103	SS	8
13	104	DT	10
14	104	ADBMS	8
15	104	PWRC	8

- In this Table 10.5, hours depend on the subject and subject depends on the Faculty code and Sr. No.
- But, hours is neither dependent on the faculty code nor the SNO. Hence, there exists a transitive dependency between Sr. No., Subject and Hours.
- If a faculty code is deleted, due to transitive dependency, information regarding the subject and hours allotted to it will be lost.
- For a table to be in 3rd Normal form, transitive dependencies must be eliminated.
- So, we need to decompose the table further to normalize it.

Fac_Sub

Sr. No	Faculty code	Subject
1	100	DSA
2	100	SS
3	100	IS
4	101	MIS
5	101	PM
6	101	IS
7	102	PWRC
8	102	PCOM
9	102	IP
10	103	DT
11	103	PCOM
12	103	SS
13	104	DT
14	104	ADBMS
15	104	PWRC

Sub_Hrs

Subject	Hours
DSA	16
SS	8
IS	12
MIS	16
PM	8
PWRC	8
PCOM	8
IP	16
DT	10
ADBMS	8

- After decomposing the 'Subject' table we now have 'Fac_Sub' and 'Sub_Hrs' table respectively.

4. Advantages

(I) Insertion

No redundancy of data for subject and hours while inserting the records.

(II) Updation

- Subject and hours are stored in the separate table.
- So updation becomes much easier as there is no repetitiveness of data.

(III) Deletion

Even if the faculty leaves the organization, the hours allotted to a particular subject can be still retrieved from the Sub_Hrs table.

Q.6 Specify codd's Norms to be satisfied by RDBMS? **SPPU - May 19. 4 Marks**

Ans. : Boyce Codd Normal Form (BCNF)

1. Introduction

- This normal form is governed by Raymond F. Boyce and E.F. Codd (1974)
- BCNF is more rigorous form of 3NF.
- The intention of Boyce-Codd Normal Form (BCNF) is that 3NF does not satisfactorily handle the case of a relation processing two or more composite or overlapping candidate keys.
- **Candidate key** is a column in a table which has the ability to become a primary key.
- **A determinant** is any attribute (simple or composite) on which some other attribute is fully functionally dependent.

  $a \rightarrow b$

  Then, attribute 'a' is determinant.

2. Definition

A relation R is said to be in BCNF, if and only if every determinant is a candidate key.

3. Sample relations

ID	Ename	Qualification	Grade	Did	DName
1	Satish	B.E.	C	20	EX
2	Savita	M.E.	B	30	CE
3	Mahesh	Ph.D.	A	10	IT

- Suppose Grade of faculty depends on his qualification. This relation has 3 determinants as ID, Qualification and Did. But if only (ID, Did) is candidates key then relation may not be in BCNF. For relation to be in BCNF every determinant must be candidates key.
- For table,

ID, Qualification → Grade

ID→ Enami, Did

Did→ DName

ID	Ename	Did
1	Satish	20
2	Savita	30
3	Mahesh	10

- This relation has only one determinant as ID and it is also a candidate's key then relation is in BCNF.

ID	Qualification	Grade
1	B.E.	C
2	M.E.	B
3	Ph.D.	A

- This relation has determinants as ID and Qualification they are in candidates key then relation is in BCNF.

Example :

- Soldiers are part of one or many units, and each unit is under the control of an officer.

SOLDIERID	OFFICERID	UNITID
1	A	1
2	A	1
3	B	2

- Firstly, we'll identify the dependencies.
- There is a dependency between (SOLDERID + OFFICERID) and UNITID, a soldier and an officer implies their respective unit, but there is also a dependency between UNITID and OFFICERID.

  SOLDERID → UNITID

  UNIT ID → OFFICEID

  SOLDIERID, OFFICEID → UNITID
- This last dependency however is not partial (dependence on part of a prime attribute), nor transitive (dependence of a nonprime attribute on another nonprime attribute, and OFFICERID is a prime attribute).
- What we have is a table where a determinate in the table is not a candidate key (UNITID).
- Candidate key are SOLDERID and OFFICEID.
- We can convert the above to BCNF by realizing that a better composite key is one of SOLDERIERID and UNITID, which creates a dependency between UNITID and OFFICERID, which is a partial dependency.
- This is then resolved by dividing the table, the solution being as follow :

Candidate key (SOLDERID, SOLDERID) → UNITID

SOLDIERID	UNITID
1	1
2	1
3	2

Candidate key (UNIT ID) AND UNIT ID → OFFICEID

UNITID	OFFICERID
1	A
1	A
2	B

The above table is now in BCNF.

Fourth Normal Form (4NF)

1. Introduction

This normal form is given by Ronald Fagin (1977).

- Fourth normal form tries to remove multi valued dependency among attributes.

2. Definition

A relation is said to be in **fourth normal form** if each table contains no more than one multivalued dependency per key attribute.

3. Example :

Seminar	Faculty	Topic
DBP-1	Brown	Database Principles
DAT-2	Brown	Database Advanced Techniques
DBP-1	Brown	Data Modeling Techniques
DBP-1	Robert	Database Principles
DBP-1	Robert	Data Modeling Techniques
DAT-2	Maria	Database Advanced Techniques

- In the above example, same topic is being taught in a seminar by more than 1 faculty and each Faculty takes up different topics in the same seminar.
- Hence, Topic names are being repeated several times.
- This is an example of multivalued dependency. (No multivalued dependency)

Seminar	Topic
DBP-1	Database Principles
DAT-2	Database Advanced Techniques
DBP-1	Data Modeling Techniques

- To eliminate multivalued dependency, split the table such that there is no multivalued dependency. (One multivalued dependency).

Seminar	Faculty
DBP-1	Brown
DAT-2	Brown
DBP-1	Robert
DAT-2	Maria

Chapter - 11 : Query Processing

Q.1 Define query processing. SPPU - May 18. 4 Marks

Ans. : Introduction to Query Processing

(1) Relational query processing refers to the range of activities which includes in extracting data from a database using a database query.

(2) A query processing involves below steps,

1. Scan 2. Parse 3. Validate

1. **Scan :** The scanner reads the language tokens such as SQL keywords, relation names in the text of the query.
2. **Parse :** The Parser check the query syntax to verify it is as per the syntax rules of the query language.
3. **Validate :** The query must also be validated by checking that all attributes and relation names are valid in the schema of the particular database being queried by query.

(3) Query Execution Plan

- Query execution plan will give idea about how query will be executing in stepwise manner.
- An internal representation of the query can be created as a tree data structure which is called **a query tree**.
- The DBMS must find all alternative execution strategies for retrieving the result of the query from the database.

Example : Department information can be accessed in following ways,

1. Department table
2. Employee_Department_Join table

- A query can have many possible execution strategies.
- Process of selecting a suitable strategy for processing a query is known as query optimization.
- Query optimization is achievable for simple queries but it becomes very complex for difficult queries.

(4) Each DBMS has a number of general database access algorithms that such as SELECTION, PROJECTION or JOIN or combination of these operations.

Q.2 What are the measures of Query cost ? (SPPU - Dec. 18. 4 Marks)

Ans. : Measures of a Query Cost

(1) Query Cost

- Query cost do mean by the predicted execution time required for query execution.
- The cost is the time spent on query execution, plus the CPU time required, plus all other types of time required for query execution.

(2) Measures Used for Query Cost

The query cost can be measured with help of following factors :

a. Access cost to secondary storage

This is the cost for operations searching, reading, and writingdata blocks that reside in secondary storage or disk. Factorssuch as whether the contiguous allocation used to store block on the same disk or it is scattered on the disk which affect the access cost.

b. Storage cost

This is the cost for storing intermediate files which is generated byan execution strategy for the query.

c. Computation cost

This is the cost of memory operations during query execution or it is CPU time to execute query.Such operation includes searching and sortingrecords, merging records for a join or performing computations on various field values.

d. Memory usage cost

This is the cost which shows the number of memory buffersrequired during query execution.

e. Communication cost

This is the cost of sending the query results from the one database site to the other site. (In case of a distributed or parallel database).

f. **Number of different resources** used like printer, disk accesses etc.

g. **Data transfer rate**

h. **Cost of scanning disk segment** containing tuples.

i. **Cost models for different index** access methods (tree structures-hashing).

(3) Other measures used for Query Cost

- The response time for a query evaluation plan can be act like a good measure of the cost of the query evaluation plan.
- In large DBMS number of blocks transferred from disk in unit time is usually the most important cost.
- Block transfer from **disk** is slow as compared to in **memoryoperations**. Hence, disk access cost is important measure of the cost of a query-evaluation plan.
- Estimating the CPU time is difficult as compared to estimating the disk access cost.
- We need to consider differences between following factors also;
  - Variance between rotational latency and seek time
  - Rotational latency is waiting time for the desired data to roll under the read write head and seek time is time that it takes to move the head on to the desired data.
  - Distinguish between Sequential I/O and random I/O.
  - **Sequential I/O** is when the blocks that we want to read are contiguous on disk memory while **random I/O** is when the blocks are noncontiguous or random.
  - Distinguish between Read and write operations
  - Time required to write a block of data to disk is more than to time required to read a block of data from disk.
- So more accurate measure to estimate will be,
  1. The number of seek operations
  2. The number of blocks Read
  3. The number of blocks written
- Above cost estimates generally ignores the cost of writing the final result of an operation back on to the disk.
- The costs of all the algorithms used depend on the size of the buffer present in main memory.

  Best case
  - Entire data can be read into the buffers and once it is loaded no need to access disk again.
  - Worst case
  - Buffer can hold only one or few blocks per relation.
- Generally, for estimating cost we assume the worst case.

(4) Example :

Measuring Query cost in SQL Server 2000

```
SELECT  A.AU_LNAME [AUTHOR],T.TITLE [TITLE]
FROM    AUTHORS A,TITLEAUTHOR L,  TITLES T
WHERE   L.AU_ID    = A.AU_ID
AND     T.TITLE_ID = L.TITLE_ID;
```

The above query will we join three different tables and extract data from those tables. So, with help of SQL server we can look at various cost of each table access operation (Read Operation) individually. (Shown in below diagram)

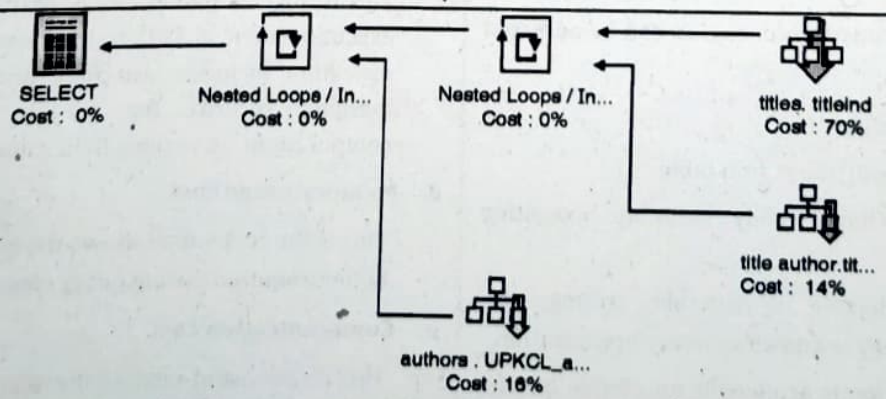


Fig. 11.1 : Sample Query Cost in SQL Server 2000