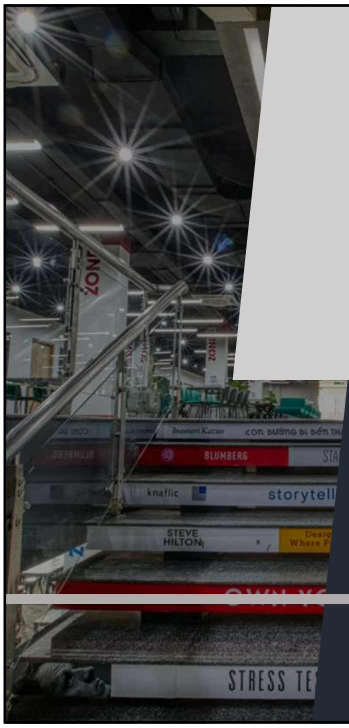

CSE301 – DATABASE

Normalization

1

Normalization

What is Normalization?

- Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.
- It is a process of organizing data in the database
  - ✓ Minimize redundancy in relation.
  - ✓ Divides larger tables into smaller tables and links them using relationships.

Introduction: 1-2

2

Who is invented Normalization?

- The inventor of the relational model Edgar Codd proposed the theory of normalization of data with the introduction of the First Normal Form, and he continued to extend theory with Second and Third Normal Form. Later he joined Raymond F. Boyce to develop the theory of Boyce-Codd Normal Form.
- The Theory of Data Normalization in MySQL server is still being developed further. For example, there are discussions even on 6^{th} Normal Form. However, in most practical applications, normalization achieves its best in 3rd Normal Form.

What is redundancy?

- When some data is stored multiple times unnecessarily in a database.

Student_id	Name	Age	Dept_Code	Dept_Name	Hod_Name
1	A	18	101	SE	XYZ
2	B	19	101	SE	XYZ
3	C	18	101	SE	XYZ
4	D	20	102	CNDC	PQR
5	E	21	102	CNDC	PQR
6	F	19	103	ECE	KLM

Why we need Normalization?

- To remove insertion, update and deletion Anomalies.
- Avoid data inconsistency
- Decrease the database size and increase database access time

Insertion Anomalies

- Insertion Anomalies: When certain data cannot be inserted into database, without the presence of data.
- Example:
  - ✓ Suppose for a new admission, until and unless a student opts for a branch, data of the student cannot be inserted, or else we will have to set the branch information as NULL.
  - ✓ Also, if we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students.

Insertion Anomalies

Student_id	Name	Age	Dept_Code	Dept_Name	Hod_Name
1	A	18	101	SE	XYZ
2	B	19	101	SE	XYZ
3	C	18	101	SE	XYZ
4	D	20	102	CNDC	PQR
5	E	21	102	CNDC	PQR
6	F	19	103	ECE	KLM
7	G	18			
			104	ME	ABC

Repetition of data

Not allowed

Deletion Anomalies

- Deletion Anomalies: If we delete some data (which is unwanted), it causes deletion of other data (which is wanted).
- Example:
  - ✓ In our Student table, two different information's are kept together, Student information and Branch information.
  - ✓ Hence, at the end of the academic year, if student records are deleted, we will also lose the branch information.
  - ✓ This is Deletion anomaly.

Deletion Anomalies

Student_id	Name	Age	Dept_Code	Dept_Name	Hod_Name
1	A	18	101	SE	XYZ
2	B	19	101	SE	XYZ
3	C	18	101	SE	XYZ
4	D	20	102	CNDC	PQR
5	E	21	102	CNDC	PQR
6	F	19	103	ECE	KLM

if this student record is deleted, we will also delete the branch information of ECE.

Update Anomalies

- Update Anomalies: When we want to update single piece of data, but it must be done at all of its copies.
- Example:
  - ✓ What if Mr. XYZ leaves the college? or is no longer the HOD of computer science department?
  - ✓ In that case all the student records will have to be updated, and if by mistake we miss any record, it will lead to data inconsistency.
  - ✓ This is Update anomaly.

Update Anomalies

Student_id	Name	Age	Dept_Code	Dept_Name	Hod_Name
1	A	18	101	SE	MNO
2	B	19	101	SE	MNO
3	C	18	101	SE	MNO
4	D	20	102	CNDC	PQR
5	E	21	102	CNDC	PQR
6	F	19	103	ECE	KLM

3 rows updated with same data

Insertion, Deletion and Update Anomalies

- Avoid data inconsistency: when the different data exists in multiple places in the database for the same condition due to update anomaly.
- Decrease the database size and increase database access time:
  - ✓ Unwanted attributes increase database size
  - ✓ Small tables increase the database access speed.

Data Inconsistency

Update Student set Hod_Name = ‘RST’ where Name = ‘C’

Student_id	Name	Age	Dept_Code	Dept_Name	Hod_Name
1	A	18	101	SE	XYZ
2	B	19	101	SE	XYZ
3	C	18	101	SE	RST
4	D	20	102	CNDC	PQR
5	E	21	102	CNDC	PQR
6	F	19	103	ECE	KLM

Hod is only updated in row 3 but not updated in row 1 & 2

Normal Forms

Normal Forms in DBMS

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (2NF)
- Boyce-Codd Normal Form (BCNF)

❑ There exists several other normal forms even after BCNF but generally we normalize till BCNF only.

First Normal Form (1NF)

- A given relation is called in First Normal Form (1NF) :
  - if each cell of the table contains only an atomic value.

    Or

    if the attribute of every tuple is either single valued or a null value.
  - Order of rows & order of columns are irrelevant.
  - In every column all value must be in same domain.
  - Every column should have a unique name
- Note:
  - ✓ By default, every relation is in 1NF.

First Normal Form (1NF)

- Example:
  - ✓ The following relation is not in 1NF:

Student_id	Name	Subjects
100	Dat	Computer Networks, Data Structure
101	Duc	Database, Java, Software Engineering
102	Phuc	Database

First Normal Form (1NF)

- ✓ This relation can be brought into 1NF.
- ✓ This can be done by rewriting the relation such that each cell of the table contains only one value.

Student_id	Name	Subjects
100	Dat	Computer Networks
100	Dat	Data Structure
101	Duc	Database
101	Duc	Java
101	Duc	Software Engineering
102	Phuc	Database

Second Normal Form (2NF)

- ❑ A given relation is called in Second Normal Form (2NF) if and only if
  - ✓ Relation already exists in 1NF.
  - ✓ No partial dependency exists in the relation ([prime → non-prime] should not be present)

Partial Dependency

- A partial dependency is a dependency where few attributes of the candidate key (prime attributes) determines non-prime attribute(s).

OR

- A partial dependency is a dependency where a portion of the candidate key or incomplete candidate key (prime attributes) determines non-prime attribute(s).
- In other words, $A \rightarrow B$ is called a partial dependency if and only if
  - ✓ A is a proper subset of some candidate key (prime attribute)
  - ✓ B is a non-prime attribute.
- If any one condition fails, then it will not be a partial dependency.

Second Normal Form (2NF)

- Example: The following relation is not in 2NF:
- R(ABCD), FD: $\{AB \rightarrow D, B \rightarrow C\}$
- Find candidate key =?
  - $\{AB\}^+ = AB = ABCD = R$ So, Candidate key is = AB
- Prime attributes: A, B (part of candidate key)
- Non-prime attributes: C, D
- Now check all FD's. Is there any (Prime → Non-prime) exists or not?
- $B \rightarrow C$ is Prime → Non-prime. So this relation is not in 2NF.

Second Normal Form (2NF)

- Example: How to decompose this relation into 2NF?
- If a partial dependency exists,
- we remove the partially dependent attribute(s) from the relation by placing them in a new relation along with a copy of their determinant (Create a new table for each partial dependency).
- $R(ABCD)$, FD: $\{AB \rightarrow D, B \rightarrow C\}$ will be,
  - $R_1(\underline{AB}D)$, FD: $\{AB \rightarrow D\}$ here AB is key
  - $R_2(\underline{B}C)$, FD: $\{B \rightarrow C\}$ here B is key

Second Normal Form (2NF)

- $R(ABC)$, FD: $\{B \rightarrow C\}$ will be,
  - $R_1(\underline{ABC})$, FD: $\{AB \rightarrow C\}$ here AB is key
  - $R_2(\underline{B}C)$, FD: $\{B \rightarrow C\}$ here B is key

A	B	C
A	1	X
B	2	Y
C	3	Z
C	3	Z
D	3	Z
C	3	Z

A	B	C
A	1	X
B	2	Y
C	2	Z
C	3	W
D	3	W
C	3	W

B	C
1	X
2	Y
3	Z

Second Normal Form (2NF)

- Example: The following relation is in 2NF:
- R (V , W , X , Y , Z) FD: {VW → XY, Y → V, WX → YZ}
- The possible candidate keys for this relation are: VW , WX , WY
- From here, Prime attributes = { V , W , X , Y }, Non-prime attributes = { Z }
- Now, if we observe the given dependencies, there is no partial dependency.
- This is because there exists no (prime → non-prime) dependency where incomplete candidate key determines any non-prime attribute.
- Thus, we conclude that the given relation is in 2NF.

Third Normal Form (3NF)

- A given relation is called in Third Normal Form (3NF) if and only if
  - ✓ Relation already exists in 2NF.
  - ✓ No transitive dependency exists for non-prime attributes.

OR

- A relation is called in Third Normal Form (3NF) if and only if : Any one condition holds for each non-trivial functional dependency A → B
  - ✓ A is a super key
  - ✓ B is a prime attribute.

Transitive Dependency

- A → B is called a transitive dependency if and only if
  - ✓ A is not a super key.
  - ✓ B is a non-prime attribute.
  - ✓ If any one condition fails, then it is not a transitive dependency.
  - ✓ Or simply we say if non-prime → non-prime dependency exists then, it is transitive dependency.

Third Normal Form (3NF)

- A given relation is called in Third Normal Form (3NF) if and only if
  - ✓ Relation already exists in 2NF.
  - ✓ No transitive dependency exists for non-prime attributes.

OR

- A relation is called in Third Normal Form (3NF) if and only if : Any one condition holds for each non-trivial functional dependency A → B
  - ✓ A is a super key
  - ✓ B is a prime attribute.

Third Normal Form (3NF)

- Example: The following relation is not in 3NF:
- R(ABC), FD: {A → B, B → C}
- Find candidate key =?
  - = $\{A\}^+$ = AB = ABC = R So, Candidate key is = A
- Prime attributes: A (part of candidate key)
- Non-prime attributes: B, C
- Now check all FD's. Is there any (Non-prime → Non-prime) exists or not?
- B → C is Non-prime → Non-prime. So this relation is not in 3NF.

Third Normal Form (3NF)

- Example: How to decompose this relation into 3NF?
- If a partial dependency exists, first we remove the partially dependent attribute(s) from the relation by creating a new table for each partial dependency.
- If a transitive dependency exists, we remove the transitively dependent attribute(s) from the relation by placing the attribute(s) in a new relation along with a copy of the determinant. (Create a new table for each transitive dependency).
- R(ABC), FD: {A → B, B → C} will be,
  - $R_1(\underline{A}B)$, FD: {A → B} here A is key
  - $R_2(\underline{B}C)$, FD: {B → C} here B is key

Third Normal Form (3NF)

- Example: The following relation is in 3NF:
- R (A , B , C , D , E)   FD: {A → BC, CD → E, B → D, E → A}
- Find candidate key =?
  - So, Candidate keys are A , E , CD , BC.
- Prime attributes: A , B, C, D, E (part of candidate key)
- Non-prime attributes: There are no non-prime attributes
- Now check all FD's. Is there any (Non-prime → Non-prime) exists or not?
- There is no Non-prime → Non-prime. So this relation is in 3NF.