Database Development & Design

LOGICAL DESIGN (NORMALIZATION)

Outline

- Database Anomalies
- Database Normalization

Database Anomalies

Anomalies in the relational model refer to inconsistencies or errors that can arise when working with relational databases, specifically in the context of data insertion, deletion, and modification.

These anomalies can be categorized into three types:

- Insertion Anomalies
- Deletion Anomalies
- Update Anomalies

Database Anomalies

- **Insertion Anomalies:** These anomalies occur when it is not possible to insert data into a database because the required fields are missing or because the data is incomplete.

- **Deletion anomalies:** These anomalies occur when deleting a record from a database and can result in the unintentional loss of data.

- **Update anomalies:** These anomalies occur when modifying data in a database and can result in inconsistencies or errors.

Database Anomalies - Example

Student

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD-COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajasthan	India	18
4	SURESH		Punjab	India	21

Student_Course

STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

Insertion anomaly: If a tuple is inserted in referencing relation and referencing attribute value is not present in referenced attribute.

Example: If we try to insert a record in STUDENT_COURSE with STUD_NO = 7

Database Anomalies - Example

SID	Name	Subject	Mobile
1	Raj	English	65468154
2	Jyoti	Home science	87668545
3	Vikash	Maths	26865948
1	Raj	Maths	Null
3	Vikash	Science	Null

Update anomaly: If a record has multiple copies, and if we make updates in a few copies and leave the remaining copies with old values, then the search result for that record may be misled the information. This may create inconsistency.

Example: If we update the new mobile number for **Raj** at the second time then Raj has two mobile numbers. This may create confusion that which one is actually is the correct mobile number!

Database Anomalies - Example

SID	Name	Subject	Mobile
1	Raj	English	65468154
2	Jyoti	Home science	87668545
3	Vikash	Maths	26865948
1	Raj	Maths	Null
3	Vikash	Science	Null

Deletion anomaly: It occurs when deleting a record from a database and can result in the unintentional loss of data.

Example: If we delete the subject **Home Science** since it's no longer available then the record of student **Jyoti** is also deleted although Jyoti is still an active student in the college!

Database Normalization

Database Normalization

- DB Normalization is the process of organizing the attributes of the database to reduce or eliminate data redundancy (having the same data but at different places).
- Data redundancy unnecessarily increases the size of the database as the same data is repeated in many places.
- Inconsistency problems also arise during insert, delete and update operations (database anomalies).
- Normalization is an important process in database design that helps in improving the efficiency, consistency, and accuracy of the database. It makes it easier to manage and maintain the data and ensures that the database is adaptable to changing business needs.

Normalization Types

- Normal Forms Types are:
  - **First Normal Form (1st NF)**
  - **Second Normal Form (2nd NF)**
  - **Third Normal Form (3rd NF)**
  - **Boyce-Codd Normal Form (BCNF) (4th NF)**

- The database designers don't need normalize to the highest possible normal form (usually up to 3NF), and BCNF (4NF) rarely used in practice.

First Normal Form

A relation (table) is said to be in 1st NF if:

1. Each **cell** should contain a single value (table is **Atomic**).
2. Each **record** needs to be **unique**.
3. **Values** stored in a column should be of the **same domain**.
4. All the **columns** in a relation should have **unique names**.

To apply the 1st NF:

- If there is any multi-valued or complex attribute on relation, we should extract them to another relation (new table) using one-to-many relationship (Making everything **Atomic**).
- The primary key of the new relation is a combination of the primary key of the original relation plus an attribute from the newly created relation for unique identification; as we learned in mapping.
- Remove any composite attribute by taking the simple attributes only; as we learned in mapping.

First Normal Form - Examples

Employee

ID	Name	Mobile	city
1001	Ali	079525215	Amman
1001	Ali	079525215	Zarqa
1002	Mohammad	079995214	Irbid
1002	Mohammad	079995214	Amman
1003	Sami	079945852	Amman

Employee is a bad design as it is not in 1NF, as it have a multi-value attribute (city)

- To apply the 1st **NF**: we should remove any multi value attribute (each record needs to be unique).

Employee

ID	Name	Mobile
1001	Ali	079525215
1002	Mohammad	079995214
1003	Sami	079945852

Employee_city

ID	city
1001	Amman
1001	Zarqa
1002	Irbid
1002	Irbid
1003	Amman

First Normal Form - Examples

Employee

ID	Name	Mobile	Address
1001	Ali	079525215	Amman
1001	Ali	078521466	Amman
1002	Mohammad	079995214	Irbid
1002	Mohammad	078922541	Irbid
1003	Sami	079945852	Amman

Employee is a bad design as it is not in 1NF, as it have a multi-value attribute (Mobile)

- To apply the 1st **NF**: we should remove any multi value attribute (each record needs to be unique).

Employee

ID	Name	Address
1001	Ali	Amman
1002	Mohammad	Irbid
1003	Sami	Amman

Employee_Mobile

Employee ID	Mobile
1001	079525215
1001	078521466
1002	079995214
1002	078922541
1003	079945852

First Normal Form - Examples

Employee

ID	Name	Mobile	City
1001	Ali	079525215	Amman
1001	Ali	078521466	Zarqa
1002	Mohammad	079995214	Irbid
1002	Mohammad	078922541	Irbid
1003	Sami	079945852	Amman

Employee is a bad design as it is not in 1NF, as it have multi-value attributes (Mobile and city)

- To apply the 1st **NF**: we should remove any multi value attribute (each record needs to be unique).

Employee

ID	Name
1001	Ali
1002	Mohammad
1003	Sami

Employee_Mobile

Employee ID	Mobile
1001	079525215
1001	078521466
1002	079995214
1002	078922541
1003	079945852

Employee_City

ID	city
1001	Amman
1001	Zarqa
1002	Irbid
1003	Amman

First Normal Form - Examples

Employee

ID	Name	Mobile	Mobile	Address
1001	Ali	079525215	078521466	Amman
1002	Mohammad	079995214	078922541	Irbid
1003	Sami	079945852		Amman

Employee is a bad design as it is not in 1NF, as it have multi-value attributes (Mobile)

- To apply the 1st **NF**: we should make every cell **Atomic**

Employee

ID	Name	Address
1001	Ali	Amman
1002	Mohammad	Irbid
1003	Sami	Amman

Employee_Mobile

Employee ID	Mobile
1001	079525215
1001	078521466
1002	079995214
1002	078922541
1003	079945852

First Normal Form - Examples

Employee

ID	Name	Mobile	Address
1001	Ali	079525215, 078521466	Amman
1002	Mohammad	079995214, 078922541	Irbid
1003	Sami	079945852	Amman

Employee is a bad design as it is not in 1NF, as it have multi-value attributes (Mobile)

- To apply the 1st **NF**: we should make every cell **Atomic**

Employee

ID	Name	Address
1001	Ali	Amman
1002	Mohammad	Irbid
1003	Sami	Amman

Employee_Mobile

Employee ID	Mobile
1001	079525215
1001	078521466
1002	079995214
1002	078922541
1003	079945852

First Normal Form - Examples

Employee

ID	Name	Mobile	Address
1001	Ali	079525215, 078521466	Amman
emp1002	Mohammad	079995214, 078922541	Irbid
1003	Sami	079945852	Amman

Employee is a bad design as it is not in 1NF, as it have multi-value attributes (Mobile) and ID contains different data domains.

- To apply the 1st **NF**: Values stored in a column should be of the same domain and every cell in the relation should be **Atomic**.

Employee

ID	Name	Address
1001	Ali	Amman
1002	Mohammad	Irbid
1003	Sami	Amman

Employee_Mobile

Employee ID	Mobile
1001	079525215
1001	078521466
1002	079995214
1002	078922541
1003	079945852

First Normal Form - Examples

Employee

ID	Name	Mobile	Address
1001	Ali Rashed	079525215	Amman
1002	Mohammad Ahmad	079995214	Irbid
1003	Sami Sameer	079945852	Amman

Employee is a bad design as it is not in 1NF, as it have a composite attribute (Name)

- To apply the 1st **NF**: we should make every cell **Atomic**

Employee

ID	fname	lname	Mobile	Address
1001	Ali	Rashed	079525215	Amman
1002	Mohammad	Ahmad	079995214	Irbid
1003	Sami	Sameer	079945852	Amman

Second Normal Form

A relation (table) is said to be in 2nd NF if:

1. Relation (table) must be in 1st NF.
2. All non-key attributes are **fully** functional dependent on the primary key. (There should be **NO Partial Dependency**).

- **To apply the 2nd NF:**
  - To remove all partial dependencies by break the relation to two relations. Remove the attribute which is causing partial dependency and move it to some other relation where it fits in well.
  - The relation is automatically in 2nd NF**, if and only if the primary key consist of single attribute** (every non-prime attribute of the relation is dependent on the whole of every candidate key) and it's already in 1st NF.
  - Remember: An attribute that is not part of any candidate key is known as non-prime attribute.

Second Normal Form

CustomerID	AccountID	CustomerName	CustomerCity	AccountStatus	AccountBalance
1	1111012	Dana	Tokyo	Activated	50000
2	1112524	Dana	London	Activated	38000
3	1111968	Andrew	Tokyo	blocked	25000
4	1111968	Ali	Amman	blocked	25000
4	2222222	Ali	Amman	Activated	30000

CustomerName and AccountStatus are non-prime attributes, and each depends on a part of the candidate key (CustomerID, AccountID) i.e., it has Partial Dependency, so it is a bad design, and violates 2ND normal form.

Second Normal Form

To remove all partial dependencies by break the relation to two relations. Remove the attribute which is causing partial dependency and move it to some other relation where it fits in well.

Second Normal Form

CustomerID	AccountID	CustomerName	CustomerCity	AccountStatus	AccountBalance

CustomerID	CustomerName	CustomerCity

AccountID	AccountStatus	AccountBalance

CustomerID	AccountID

2nd Normal Form

Second Normal Form

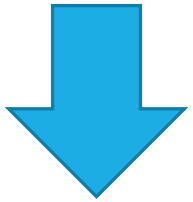
emp SSN	proj number	hours	emp name	proj name	proj location
123456789	1	32.5	John	ProductX	Bellaire
123456789	2	7.5	John	ProductY	Sugarland
333445555	2	10	Franklin	ProductY	Sugarland
333445555	3	10	Franklin	ProductZ	Houston
333445555	10	10	Franklin	Computerization	Stafford
333445555	20	10	Franklin	Reorganization	Houston
453453453	1	20	Joyce	ProductX	Bellaire
453453453	2	20	Joyce	ProductY	Sugarland
666884444	3	40	Ramesh	ProductZ	Houston

- Emp SSN → {emp name}
- Proj number → {proj name, proj location}

It has Partial Dependency, so it is a bad design, and violates 2^{ND} normal form.

Second Normal Form

emp SSN	proj number	hours	emp name	proj name	proj location



2nd Normal Form

emp SSN	emp name

proj number	proj name	proj location

emp SSN	proj number	hours

Third Normal Form

A relation (table) is said to be in 3rd NF if:

1. Relation (table) must be in 2nd NF.
2. Has **No Transitive functional dependencies**. All the attributes in a relation are identified only by the candidate keys of that relation and not by any non-prime attributes

- **To apply the 3rd NF:**
  - We should remove any transitive dependencies; a non-key attribute may not be functionally dependent on another non-key attribute.

Third Normal Form

Book	Author name	Author Age
Book A	Author1	66
Book B	Author2	40
Book C	Author1	66

Book	Author name

Author name	Author Age

- **{Book} ->{Author name}**
- **{Author name} ->{Author Age}**
- **So:**
  - **{Book} -> {Author Age}**

It has Transitive Dependency, so it is a bad design, and violates 3rd normal form.

3rd Normal Form

Third Normal Form

empID	empName	zip	state	city

empID -> Zip && Zip -> {state, city} && empID -> {state, city}

It has Transitive Dependency, so it is a bad design, and violates 3rd normal form.

empID	empName	zip

zip	state	city

3rd Normal Form

References

- Elmasri, R., & Navathe, S. (2017). Fundamentals of database systems (Vol. 7). Pearson
- Eng. Lina’s slides
- Dr. Raneem’s slides.