Chapter 4 DATA INTEGRITY AND NORMALIZATION

4.1 OVERVIEW

4.1.1 Data Integrity

Database integrity refers to the correctness and consistency of data. It is another form of database protection. While it is related to security and precision, it has some broader implications as well. Security involves protecting the data from unauthorized operations, while integrity is concerned with the quality of data itself. Integrity is usually expressed in terms of certain constraints which are the consistency rules that the database is not permitted to violate. Following two are the most important constraints in relational databases:

(i) **Entity Integrity:** is a constraint on primary values that states that no attribute of a primary key should contain nulls.

(ii) **Referential integrity:** is a constraint on foreign key values that states that if a foreign key exists in a relation, then either the foreign key value must match the primary key value of some tuple in its home relation or the foreign key value must be completely null.

4.1.2 Normalization

Normalization is the process of converting complex data structures into simple and stable data structures. It is based on the analysis of *functional dependence*.

In other words, Normalization is a technique for reviewing the entity/attribute lists to ensure that attributes are stored "where they belong". It is the basis for a relational data base system. In practice, it is simply an applied common sense. More formally stated, it is the process of analyzing the dependencies of attributes within entities. Attributes for each entity are checked consecutively against three sets of rules, making adjustments when necessary to put the entity in First, Second and Third normal form.

First, we discuss what is functional dependence.

"A functional dependency is a particular relationship between two attributes. For any relation R, attribute B is functionally dependent on attribute A if, for

every valid instance of A, that value of A uniquely determines the value of B". The functional dependence of B on A is represented by an arrow, as

A B) An attribute may be functionally dependent on two or more attributes rather than a single attribute. For example, consider the relation:

COURSE (STUDID , CRSNO , CRSDATE)

The functional dependency in the relation is represented as follows:

STUDID , CRSNO ⟶ CRSDATE

The attribute on the left hand side of arrow is called determinant.

Before NORMALIZATION process, the initial entity/attribute list(s) must be checked for errors or oversights. There may be some hidden problems as:

(i) **Synonyms:** A synonym is created when two different names are Used for the same information (attribute). If an attribute resides in more than one entity, make sure that all entities use the same attribute name. For example, consider the following two entities:

ITEM	SUPPLIER	
Stock_no	Supplier_Id	(error)
Item_colour	Supplier_Name	
Supplier_Code		

We should use Supplier_Code instead of Supplier_Id in SUPPLIER.

(ii) **Homonyms:** A homonym is created when same name is used for Two different attributes. Consider the following example:

CUSTOMER	SUPPLIER	
Company_Name	Company_Name	(error)

We should use Supplier_Name instead of Company_Name in SUPPLIER.

(iii) **Redundant Information:** Storing the same information in two different ways or forms. Consider the example:

Employee	
Employee_Age	(error)
D_O_Birth	

Only one attribute can serve the purpose (The programmer can manipulate the Age by using D_O_Birth as the basis).

(iv) **Mutually Exclusive Data:** Mutually exclusive data exists when attributes occur whose values can be expressed as "yes/no" indicators,

can not all be true for any single entity. As an example, consider the proposed attributes of "MARRIED" and "SINGLE" in an Employee entity:

Employee
Married (a flag set if the employee is married) error
Single (a flag set if the employee is single) error

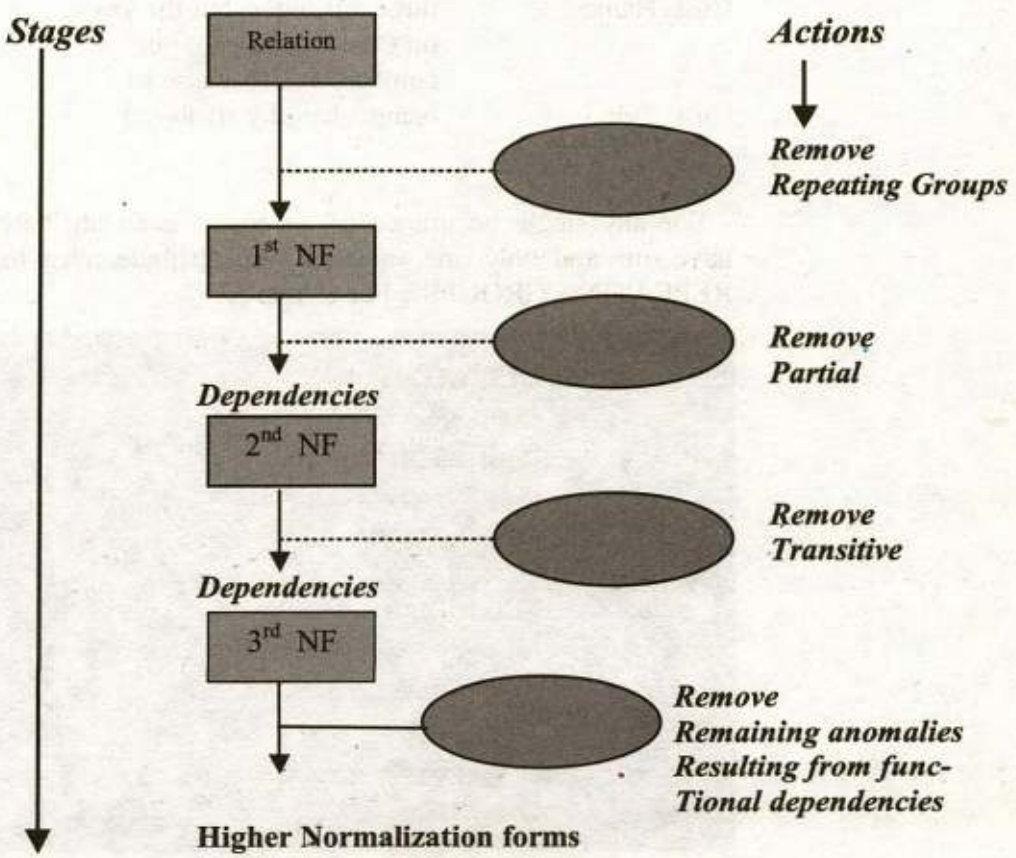
Quite often, errors of this type represent values of a larger category. Whenever possible, resolve the error by creating the larger categorical attribute. In this case, these two elements should be combined into a single attribute of "MARITAL_STATUS" which would have a value of either M (married) or S (single).

Employee
MARITAL_STATUS (An indicator of the employees Marital status)

Normalization Steps:

Normalization is often accomplished in steps, each of which corresponds to a normal form. It can be graphically expressed as follows:

(Please see the next page)



Note: Our scope of study in this semester is only upto 3rd NF.

A Normal Form is a state of a relation that can be determined by applying simple rules, regarding dependencies (or relationship between attributes), to that relation. Following is a brief discussion on different stages:

(i) First Normal Form (1 NF)

"A relation R is in First Normal Form if and only if all underlying domains contain atomic values only".

The pre-requisite is that, A relation has always a primary key associated with it. Thus, we can define it as follows also:

- All entities must have a key, composed of a combination of one or more attributes which uniquely identify one occurance of the entity. For example:

CUSTOMER

Cust_Id	(We can create key on these
Cust_Name	three attributes, but the key
..	on Cust_Name could be
..	cumbersome (because of
Cust_Telno	being a lengthy attribute)
..	
..	

- For any single occurance of an entity, each attribute must have one and only one value or "An attribute must have no REPEATING GROUPS". For example:

Case-1:	DEPARTMENT	
	Dept_No	
	Dept_Name	
	Emp_No	(error)
	Emp_Name	(error)
Case-2:	MAGAZINE	
	Mag_Code	
	Mag_Name	
	Mag_Addr	
	Mag_Telno	
	Issue_Date	(error)

Case-3:

ITEM
Item_Code
Item_Desc
Stock_No
Item_Type
Weight
Colour
Qty_Ordered
Unit_Price
Order_No (error)
Cust_Name

The error exists in the above examples because some Attribute(s) are being repeated for a single occurance of Each record. We should try to avoid this repitition. Following are the steps to achieve this:

Step-1: Whenever repeating groups occur, the repeating Attribute must be removed and placed "where it Belongs", under the entity that it describes.

Step-2: Next, study the relationship of where the Repeating attribute came from, and where the Attribute went to . Determine if the From-To Relationship is 1:M or M:N.

Let us apply these steps in case-1. We end up with the Following two relations.

DEPARTMENT	EMPLOYEE
Dept_No	Emp_No
Dept_Name	Emp_Name

Now ask for the relationship in this case:
"For one department, are there one or more mployees?" Then repeat the question in reverse. "For one employee, Are there one or many departments?" In this case, one Department has many employees, but one employee has Only one department. Therefore, the relationship is 1:M. When the relationship is 1:M, this is acceptable. No Further adjustment is necessary to make it 1:M or M:1.

But if we apply the above steps in case-3, we end up with The following two relations:

ITEM	ORDER
Item_Code	Order_No
Item_Desc	Cust_Name
Stock_No	Qty_Ordered
Item_Type	Unit_Price
Weight	
Colour	

Now ask the similar question in this case also, we find out that for each item, there are many orders, and for each Order, there are many items. Thus it is M:N relation. The Main problem here is, "where to store the intersection Data i.e., Qty-Ordered and Unit-price". To solve this, Create another entity "ITEM_ORDERED" as:

ITEM-ORDERED
Order_No
Item_No
Qty_Ordered
Unit_Price

And define the key as "the combination of Order_No and Item_No on this entity. This provides us two more tables Having 1:M relationship, which is acceptable.

To have a better understanding about the whole concept, let us consider the following table of data (having repeating groups):

STUD-ID	NAME	DEPT	MONFEE	CRSNO	CDTE
100	Ahmad	Marketing	1000	SPSS	190696
				SURVEYS	100796
140	Nazir	Accounting	1200	TAXACCT	120897
110	Hamid	Inf.Systems	1100	SPSS	140796
				COBOL	220796
190	Rashid	Finance	1200	INVESTMT	200697
150	Hussain	Marketing	1000	SPSS	190697
				SYSANAL	200797

To bring it to first Normal Form, we eliminate the repeating groups from the Table and fill in the missing information (in the cells having no information). Name the table as "STUDENT".

STUDENT

STUD-ID	NAME	DEPT	MONFEE	CRSNO	CDTE
100	Ahmad	Marketing	1000	SPSS	190696
100	Ahmad	Marketing	1000	SURVEYS	100796
140	Nazir	Accounting	1200	TAXACCT	120897
110	Hamid	Inf.Systems	1100	SPSS	140796
110	Hamid	Inf.Systems	1100	COBOL	220796
190	Rashid	Finance	1200	INVESTMT	200697
150	Hussain	Marketing	1000	SPSS	190697
150	Hussain	Marketing	1000	SYSANAL	200797

(ii) **Second Normal Form (2 NF):** A relation is in second normal form (2 NF) if it is in 1 NF and every non-key attribute is fully functionally dependent on the primary key. More precisely:

"To be in 2 NF, every non-key attribute must depend on the key and all parts of the key".

A table (relation) will be in 2NF if any of the following conditions apply.

(a) The primary key consists of only one attribute.
(b) No non-key attributes exist in the relation.
(c) Every non-key attribute is functionally dependant on the full set of primary key attributes.

Now consider the table STUDENT (which is in 1 NF).There are a lot of redundancies in this table, so it is not an acceptable stage. In shorthand notation, it is expressed as:

STUDENT(STUD-ID,NAME,DEPT,MONFEE,CRSNO,CDTE)

The functional dependencies in this relation are the as follows:

STUD-ID ⟶ NAME,DEPT,MONFEE
STUD-ID,CRSNO ⟶ CDTE

The primary key in ii above is the composite key:

STUD-ID + CRSNO.

Therefore, the non-key attributes NAME,DEPT and MONFEE are functionally dependent on part of the primary key (STUD-ID) but not on CRSNO.

A partial functional dependency exists when one or more non-key attributes (such as NAME) are functionally dependant on part (but not all) of the primary key.

The partial functional dependency in the above table creates redundancy in that table, which results in certain anomalies when the table is updated. i.e.,

(i) Insertion Anomaly: To insert a row for the table, we must provide the values for both STUD-ID and CRSNO.

(ii) Deletion Anomaly: If we delete a row for one student, we lose the information that the student completed a course on a particular date.

(iii) Modification Anomaly: If a students monthly fee changes, we must record the change in multiple rows (for students, who have completed more than one course).

To convert a relation to 2 NF, we decompose the relation (having redundant data) into two relations that satisfy one of the conditions described above.

Now, splitting the relation (STUDENT) , we will get the following two relations, namely STUDENT1 and COURSE. This step is done to get rid of the redundant data) The two tables are shown below:

TABLE 1: STUDENT1(STUD-ID,NAME, DEPT, MONFEE)

STUDENT1

STUD-ID	NAME	DEPT	MONFEE
100	Ahmad	Marketing	1000
140	Nazir	Accounting	1200
110	Hamid	Inf.Systems	1100
190	Rashid	Finance	1200
150	Hussain	Marketing	1000

This relation (table) satisfies condition 1 stated above), thus it is in 2 NF.

TABLE 2: COURSE(STUD-ID,CRSNO,CDTE)

COURSE

STUD-ID	CRSNO	CDTE
100	SPSS	190696
100	SURVEYS	100796
140	TAXACCT	120897
110	SPSS	140796
110	COBOL	220796
190	INVESTMT	200697
150	SPSS	190697
150	SYSANAL	200797

Note: The attributes STUD-ID and CRSNO have been concatenated to make a unique key for the relation.

This relation (table 2) satisfies condition 3 above), thus it is in 2 NF.

These two relations are free of anomalies now.

(iii) **Third Normal Form (3 NF)**

A relation is in third normal form (3 NF) if it is in 2 NF and no transitive dependencies exist:

What is a Transitive Dependency? It is a functional dependency in a relation between two (or more) non-key attributes.

A more precise definition for 3 NF is:"A non-key attribute must not depend on any other non-key attribute" or if a non-key attribute's value can be obtained simply by knowing the value of another non=key attribute, the relation is not in 3 NF.

Consider a relation as follows:

SALES(CUSTNO,NAME,SALESMAN,REGION)

Where CUSTNO is the primary key.

The following functional dependencies exist in the relation.

(a) *CUSTNO -------> NAME,SALESMAN*

(b) *SALESMAN -------> Region (since each salesman is assigned a unique region)*

Notice that SALES is in 2 NF, because the primary kay consists of a single attribute (CUSTNO). However, there is a transitive dependency, because REGION is functionally dependent on SALESMAN which in turn is functionally dependent on CUSTNO. As a result, there are update anomalies in relation SALES.

CUSTNO	NAME	SALESMAN	REGION
8023	AAAA	Ahmad	South
9167	BBBB	Bashir	West
7924	CCCC	Ahmad	South
6837	DDDD	Khalid	East
9596	EEEE	Bashir	West
7018	FFFF	Munir	North

Figure : A relation with Transitive dependency

The Anomalies:

(i) Insertion Anomaly: A new salesman (Abid), assigned to the North region can not be entered until a customer has been assigned to that salesman (since a value of CUSTNO must be provided to insert a row in the table(relation).

(ii) Deletion Anomaly: If customer number 6837 is deleted from the relation, we lose the information that salesman Khalid is assigned to the east region.

(iii) Modification Anomaly: If salesman Ahmad is re-assigned to the east region, several rows must be changed to reflect the fact (two rows in this case).

These anomalies arise as a result of the transitive dependency. This problem (the transitive dependency) can be removed by de-composing the relation SALES into two relations as shown below:

SALE 1

CUSNO	NAME	SALESMAN
8023	AAAA	Ahmad
9167	BBBB	Bashir
7924	CCCC	Ahmad
6837	DDDD	Khalid
8596	EEEE	Bashir
7018	FFFF	Munir

SMAN

SALESMAN	REGION
Ahmad	South
Bashir	West
Khalid	East
Munir	North

SALE1(CUSTNO,NAME,SALESMAN)

SMAN(SALESMAN,REGION)

Now, both the relations (SALE1 & SMAN) are in 3 NF, since no transitive dependency exist. We can verify that the anomalies that exist is SALES are not present in SALE1 and SMAN.

Note: that SALESMAN which is the determinant in the transitive dependency in SALES, became the primary key in SMAN. SALESMAN is also a foreign key in SALE1

Exercise 4c

1. Fill in the blanks:
   (i) Entity integrity constraint states that the _____ can not be null.
   (ii) _____ key must refer to the primary key in another table or it must be null.
   (iii) Normalization is the process of converting _____ structures into simple and stable structures.
   (iv) A(n)_____ is a partial relationship between attributes of an entity.
   (v) During the first normal form _____ groups are removed.
   (vi) To be in 2NF, a relation must be in _____.
   (vii) In 3NF, no _____ dependency exists.
   (viii) A(n) _____ exists when one or more non-key attributes are functionally dependant on part of the primary key.
   (ix) When a new record is added in a relation, it may cause _____ anomaly.
   (x) Referential integrity is a constraint on _____ key value.

2. Select the correct option:
   (i) In 3NF, which form of dependency is removed?
   a) functional b) non-functional
   c) associative d) transitive

   (ii) In relational database, a table is also called a:
   a) tuple b) relation
   c) file d) schema

   (iii) In 3NF, a non-key attribute must not depend on a(n):
   a) non-key attribute b) key attribute
   c) composite key d) sort key

   (iv) Different attributes in two different tables having same name are referred to as:
   a) synonym b) homonym
   c) acronym d) mutually exclusive

   (v) Every relation must have a:
   a) primary key b) candidate key
   c) secondary key d) composite key

3. Mark as True or False
   (i) Normalization is the process of converting complex data structures into simple data structures.
   (ii) A relation is decomposed to convert it from 1NF to 2NF.
   (iii) The primary key can not be a composite key.
   (iv) In 2NF, every non-key attribute must depend on the key attribute.
   (v) A relationship involving three relations is known as a ternary relationship.
   (vi) A database anomaly leads the database to an inconsistent state.
   (vii) Partial dependencies are removed in 3NF.
   (viii) A relation may have multiple primary keys.
   (ix) In relational database, no relation can exist in isolation.
   (x) The database is normalized to avoid certain database anomalies.

4. What is meant by data integrity? What are the two types?

5. What do we do to attain entity integrity?

6. Define referential integrity. How can it be achieved?

7. Explain the following terms:
   (i) Synonym
   (ii) Homonym
   (iii) Redundancy
   (iv) Mutual Exclusiveness of data

8. What is normalization? How it can be used to bring the database in a consistent state?

9. When is a relation in first normal form? Explain with example.

10. What are the conditions for a relation to be in second normal form? Give example.

11. Define transitive dependency. How it can be removed? Explain with the context of normalization.

12. What are the database anomalies? Briefly discuss insertion, deletion and modification anomalies.

13. What anomalies arises due to transitive dependency? Discuss briefly.

14. Define functional dependency? How partial dependencies effect a relation?

15. Convert the ER diagram you have designed in the previous exercise for the admission system of your college to relational database. Also normalize the relations up to third-normal form.