# • Chapter 04 : Data Integrity and Normalization

## 4.1 Overview

Q : 04-01-01 : Explain Data Integrity ?

**Answer :**

**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 :

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

**Referential Integrity** : It 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.

Q : 04-01-02 : Explain Normalization ?

**Answer :**

**Normalization** : [It is the process of converting complex data structures into simple and stable data structures. It is based on the analysis of functional dependence]. [Normalization is a technique for reviewing the entity / attribute lists to ensure that attributes are stored "where they belong to". 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].

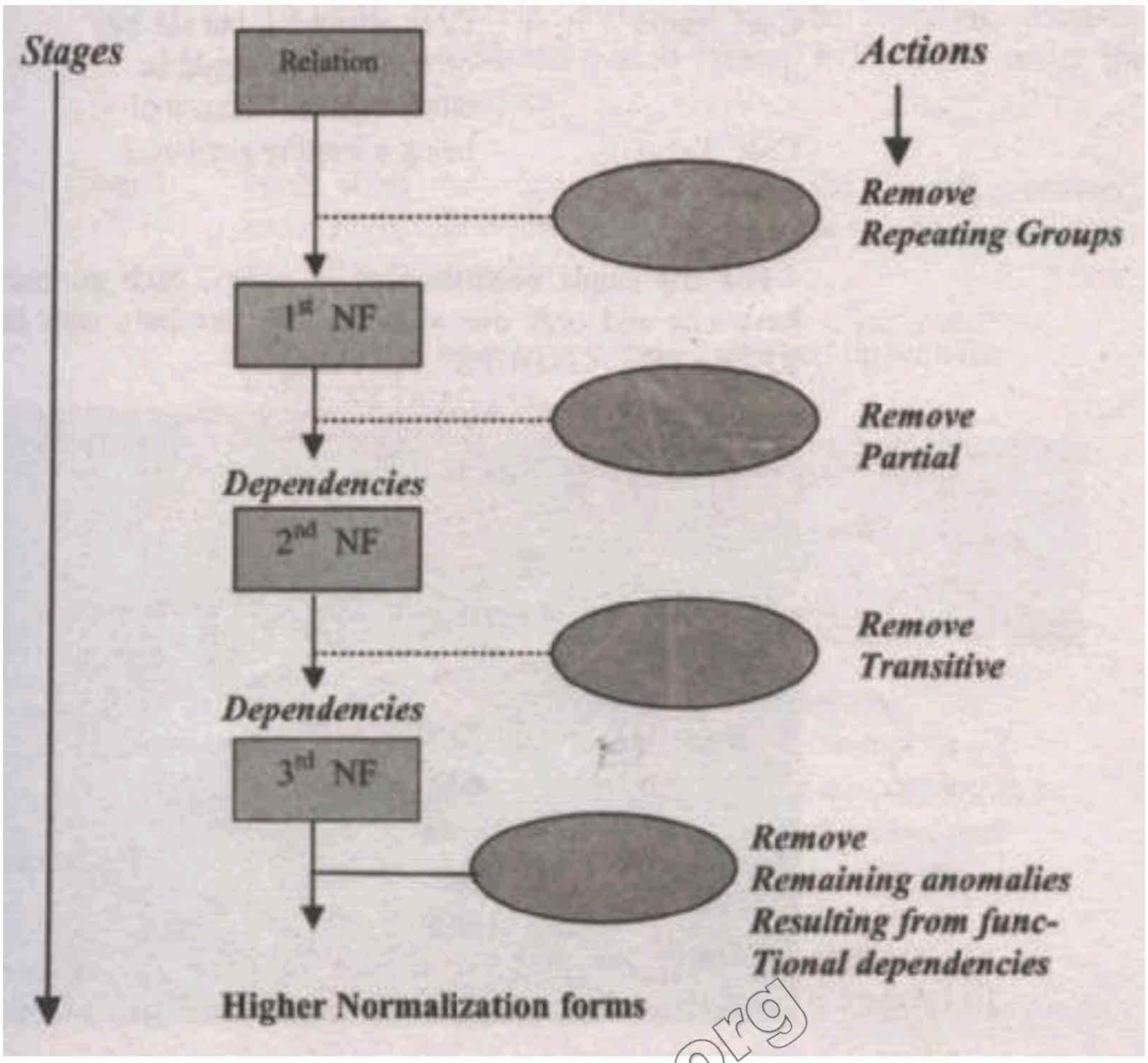
"A functional dependency is a particular relationship between two attributes. For any relation R, attribute B is functionally dependent on attribute A if, for every valid instance of A, that value of A uniquely determines the value of B" The functional dependence of B on A is represented by an arrow. An attribute may be functionally dependent on two or more attributes rather than a single attribute. There may be some hidden problems as :

**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.

**Homonyms** : A homonym is created when same name is used for Two different attributes. **Redundant Information** : Storing the same information in two different ways or forms if only one attribute can serve the purpose.

**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.

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




Q : 04-01-03 : Define Normalization and Normal Form ? Explain First Normal Form (1 NF) ?

**Answer :**

**Normalization** : [It is the process of converting complex data structures into simple and stable data structures. It is based on the analysis of functional dependence]. [Normalization is a technique for reviewing the entity / attribute lists to ensure that attributes are stored "where they belong to". It is the basis for a relational data base system].

**Normal Form** : 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.

**First Normal Form (1 NF)** : "A relation R is in First Normal Form if and only if all underlying domains contain atomic values only".

**Relation** : The Pre-Requisite is that "A relation has always a primary key associated with it".

**Unique Identification Key** : All entities must have a key, composed of a combination of one or more attributes which uniquely identify one occurrence of the entity.

**No Repeating Groups** : For any single occurrence of an entity, each attribute must have one and only one value or "An attribute must have no REPEATING GROUPS".

**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.

| DEPARTMENT | |
|---|---|
| Dept_No | |
| Dept_Name | |
| Emp_No | (error) |
| Emp_Name | (error) |

| DEPARTMENT | EMPLOYEE |
|---|---|
| Dept_No | Emp_No |
| Dept_Name | Emp_Name |

Q : 04-01-04 : Explain Second Normal Form (2 NF) ?

**Answer :**

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

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

**Necessary & Sufficient Conditions** : A table (relation) will be in 2NF if any of the following conditions apply :

The primary key consists of only one attribute.

No non-key attributes exist in the relation.

Every non-key attribute is functionally dependant on the full set of primary key attributes.

Consider table STUDENT in shorthand notation :

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

The functional dependencies in this relation are the as follows :

STUD-ID $\longrightarrow$ NAME, DEPT, MONFEE

STUD-ID,CRSNO $\longrightarrow$ 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.

**Partial Functional Dependency** : [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].

**Anomalies** : The partial functional dependency creates redundancy in the table, which results in certain anomalies when the table is updated :

**Insertion Anomaly** : To insert a row for the table, we must provide the values for both STUDENT-ID (Primary Key) and COURSE-NO (Not Primary Key).

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

**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).

**Important Note** : 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, by splitting the relation**,** we will get two relations. This step is done to get rid of the redundant data. Anomalies are removed at the end of 2 NF.

Q : 04-01-05 : Explain Third Normal Form (3 NF) and Transitive Dependency ?



**Answer :**

**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].

**Transitive Dependency** : [It is a functional dependency in a relation between two or more non-key attributes].

**Third Normal Form (3 NF) – Second Definition** : 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. The Anomalies, Insertion Anomaly, Deletion Anomaly and Modification Anomaly must be related with example data. These anomalies arise as a result of the transitive dependency. This problem (the transitive dependency) can be removed by de-composing the a relation into two relations.

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 $\longrightarrow$ NAME,SALESMAN

(b) SALESMAN $\longrightarrow$ Region (since each salesman is assigned a unique region).

Notice that SALES is in 2 NF, because the primary key 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.

**The Anomalies**

| 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

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

and

SMAN

| SALESMAN | REGION |
|---|---|
| Ahmad | South |
| Bashir | West |
| Khalid | East |
| Munir | North |

SALE1 (CUSTNO, NAME, SALMAN)
SMAN (SALESMAN, REGION)
Now, both the relations (SALE1 and SMAN) are in 3 NF, since no transitive dependency exists. We can verify that the anomalies that existed is SALES are not present in SALE1 and SMAN.
Note : 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.