

Module - III (Part-2)

NORMALIZATION

NORMALIZATION

2

NORMAL FORMS

- 1NF, 2NF, 3NF, BCNF
- Properties of decompositions, normalization, schema refinement in database design
- Other kinds of dependencies: 4NF, 5NF, DKNF

Database Normalization

3

- The main goal of **Database Normalization** is to restructure the logical data model of a database to:
 - Eliminate redundancy
 - Organize data efficiently
 - Reduce the potential for data anomalies.

Database Normalization definitions

4

- How to take a raw collection of data and break it up into more logical units or tables, in order to reduce the occurrence of redundant data in the database. This process of reducing data redundancy is referred to as *Normalization*.
- *Normalization* is a body of rules addressing analysis and conversion of data structures into relations that exhibit more desirable properties of internal consistency, minimal redundancy and maximum stability.

Database Normalization definitions

5

- **Normalization** is the process by which attributes are grouped together to form a well-structured relation.
 - We focused on the characteristics of a good relation:
 - Analyzing sample relations
 - Identifying design flaws
 - And learning how to eliminate them
 - This is called **Normalizing** a relation
- **Normalization** is a process of decomposing relations to produce smaller, well-structured relations.
- **Normalization** is a tool to validate and improve a logical design, so that it satisfies certain constraints that avoid unnecessary duplication of data.

Data Anomalies

6

- **Data anomalies** are inconsistencies in the data stored in a database as a result of an operation such as update, insertion, and/or deletion.
- Such inconsistencies may arise when have a particular record stored in multiple locations and not all of the copies are updated.
- We can prevent such anomalies by implementing 7 different level of normalization called **Normal Forms (NF)**

Brief History/Overview

7

- **Database Normalization** was first proposed by **Edgar F. Codd**.
- **Codd** defined the **first three Normal Forms**, which we'll look into, of the **7** known **Normal Forms**.
- In order to do normalization we must know what the requirements are for each of the three Normal Forms that we'll go over.
- One of the key requirements to remember is that **Normal Forms** are **progressive**. That is, in order to have **3rd NF** we must have **2nd NF** and in order to have **2nd NF** we must have **1st NF**.

NORMAL FORMS

- **Normal Form** is a state of a relation that result by decomposing that relation for a good design to **avoid redundancy**.
- The **Normal Forms** defined in **Relational database theory** represent guidelines for record design.
- The design guidelines are meaningful even if one is not using a **relational database system**.
- We present the guidelines without referring to the concepts of the **relational model** in order to emphasize their generality, and also to make them easier to understand.

Steps in Normalization

- **Normalization** can be accomplished and understood in steps, and each step results to a Normal Form.
- The **Normal Forms** are ways of measuring the levels, or depth to which a database has been normalized.
- **Normal Forms** must be implemented in sequential order.

NORMAL FORMS

10

- **First Normal Form (1NF)**
 - ▣ Included in the definition of a relation
 - ▣ Table format; no repeating groups and Primary Key (PK) identified
 - **Second Normal Form (2NF)**
 - ▣ 1NF and no partial dependencies
 - **Third Normal Form (3NF)**
 - ▣ 2NF and no transitive dependencies
 - **Boyce-Codd Normal Form (BCNF)**
 - ▣ Every determinant is a candidate key (special case of 3NF)
- Defined in terms of functional dependencies

NORMAL FORMS

11

- **Fourth Normal Form (4NF)**
 - ▣ Defined using multivalued dependencies
 - ▣ 3NF and no independent multivalued dependencies
- **Fifth Normal Form (5NF) or Project Join Normal Form (PJNF)**
 - ▣ Defined using join dependencies

NORMAL FORMS

12

- The normal forms based on FDs are *rst normal form (1NF)*, *second normal form (2NF)*, *third normal form (3NF)*, and *Boyce-Codd normal form (BCNF)*.
- These forms have increasingly restrictive requirements: Every relation in BCNF is also in 3NF,
- every relation in 3NF is also in 2NF, and every relation in 2NF is in 1NF.
- A relation is in **first normal form** if every field contains only atomic values, that is, not lists or sets.
- This requirement is implicit in our definition of the relational model.
- Although some of the newer database systems are relaxing this requirement
- 2NF is mainly of historical interest.
- 3NF and BCNF are important from a database design standpoint.

1st Normal Form

The Requirements

13

- The requirements to satisfy the 1st NF:
 - ▣ Each table has a primary key: minimal set of attributes which can uniquely identify a record
 - ▣ The values in each column of a table are atomic (No multi-value attributes allowed).
 - ▣ There are no repeating groups: two columns do not store similar information in the same table.
- 1st NF definition:
 1. A relation is in the First Normal Form, if it does not contain any repeating elements or groups.
 2. A relation is in the First Normal Form only if all underlying domains contain only atomic values.

1st Normal Form

- The objective of the **First Normal Form** is to divide the base data into logical units called tables.
- Once each table has been designed, a **Primary Key** is assigned to most or all tables.
- A **Primary Key** in a table is one or more columns that make every row of data in a table unique.
- “The **First Normal Form** deals with the *shape* of a record type”.
- Under the **First Normal Form**, all occurrences of a record type must contain the same no. of fields.

1st Normal Form

15

- **First Normal Form** excludes variable repeating fields and groups.
- This is not so much a design guideline as a matter of definition.
- Relational database theory doesn't deal with records having a variable no. of fields.

Drawback of First Normal Form:

- The main drawback of First Normal Form is redundancy of data.

1st Normal Form

Example - 1

16

□ ORDER:

1. Order_No
2. Order_Date
3. Customer_No
4. Item_No
5. Item_Name
6. Qty_Ordered
7. Rate_Per_Unit
8. Item_Value

As many as items ordered in the ORDER

1st Normal Form

Example - 1 (cont.)

17

- In this example (ORDER) contains the repeating groups in item details. So it is not in the **First Normal Form**.
- To convert a relation into the **First Normal Form**, “ remove all repeating (or multivalued) attributes to another (child) relation”.
- We perform the operation on ORDER relation and arrive at the following two relations:

ORDER

1. Order_No (PK)
2. Order_Date
3. Customer_No

ORDER_ITEM

1. Order_No (PK)
2. Item_No (PK)
3. Item_Name
4. Qty_Ordered
5. Rate_Per_Unit
6. Item_Value

1st Normal Form

Example - 1 (cont.)

18

ORDER

1. Order_No (PK)
2. Order_Date
3. Customer_No

ORDER_ITEM

1. Order_No (PK)
2. Item_No (PK)
3. Item_Name
4. Qty_Ordered
5. Rate_Per_Unit
6. Item_Value

- If we examine the relations ORDER and ORDER_ITEM, we find that there are no repeating elements or groups in both the relations and can therefore say that both these relations are in the **First Normal Form**.
- The **First Normal Form** is also called the **Normalized Form** (as against fully normalized).

1st Normal Form

Example - 2

19

Un-normalized Students table:

<u>Student#</u>	AdvID	AdvName	AdvRoom	Class1	Class2
123	123A	James	555	102-8	104-9
124	123B	Smith	467	209-0	102-8

Normalized Students table:

<u>Student#</u>	AdvID	AdvName	AdvRoom	Class#
123	123A	James	555	102-8
123	123A	James	555	104-9
124	123B	Smith	467	209-0
124	123B	Smith	467	102-8

1st Normal Form

Example - 3

20

A table is considered to be in 1NF if all the fields contain only scalar values (as opposed to list of values).

Example (Not 1NF)

ISBN	Title	AuName	AuPhone	PubName	PubPhone	Price
0-321-32132-1	Balloon	Sleepy, Snoopy, Grumpy	321-321-1111, 232-234-1234, 665-235-6532	Small House	714-000-0000	\$34.00
0-55-123456-9	Main Street	Jones, Smith	123-333-3333, 654-223-3455	Small House	714-000-0000	\$22.95
0-123-45678-0	Ulysses	Joyce	666-666-6666	Alpha Press	999-999-9999	\$34.00
1-22-233700-0	Visual Basic	Roman	444-444-4444	Big House	123-456-7890	\$25.00

Author and AuPhone columns are not scalar

1NF - Decomposition

21

1. Place all items that appear in the repeating group in a new table
2. Designate a primary key for each new table produced.
3. Duplicate in the new table the primary key of the table from which the repeating group was extracted or vice versa.

Example (1NF)

ISBN	Title	PubName	PubPhone	Price
0-321-32132-1	Balloon	Small House	714-000-0000	\$34.00
0-55-123456-9	Main Street	Small House	714-000-0000	\$22.95
0-123-45678-0	Ulysses	Alpha Press	999-999-9999	\$34.00
1-22-233700-0	Visual Basic	Big House	123-456-7890	\$25.00

ISBN	AuName	AuPhone
0-321-32132-1	Sleepy	321-321-1111
0-321-32132-1	Snoopy	232-234-1234
0-321-32132-1	Grumpy	665-235-6532
0-55-123456-9	Jones	123-333-3333
0-55-123456-9	Smith	654-223-3455
0-123-45678-0	Joyce	666-666-6666
1-22-233700-0	Roman	444-444-4444

2nd Normal Form

The Requirements

22

- The requirements to satisfy the 2nd NF:
 - All requirements for 1st NF must be met.
 - Redundant data across multiple rows of a table must be moved to a separate table.
 - The resulting tables must be related to each other by use of foreign key.
- 2nd NF definition:
 - A relation is in the **Second Normal Form** if it is in the **First Normal Form** and all non-key attributes are fully functionally dependent on the primary key.
 - A relation is in **2NF** if all the non-key attributes are dependant on all of the Key attributes.

2nd Normal Form

23

- 2nd NF is based on the concept of Full Functional dependency and removal of the partial functional dependency:
- Full FD:
 - ▣ A FD $X \rightarrow Y$ is a Full FD if removal of any attribute from X means that the dependency does not hold any more; i.e for any attribute $A \in X$ ($X - \{A\}$) does not Functionally determines Y
- Partial FD:
 - ▣ A FD $X \rightarrow Y$ is a Partial FD if some attribute $A \in X$ ($X - \{A\}$) does Functionally determines Y

2nd Normal Form (cont.)

- The conversion to **Second Normal Form** has taken place by removing attributes that are not dependent on the **Full Primary Key** attributes.
- A relation schema **R** is in **2NF** if every non-prime attribute is fully functionally dependent on any key of R.
- **Prime Attribute**: A attribute that is part of some key
- **Non-prime Attribute**: An attribute that is not part of any key

2nd Normal Form

Example - 1

25

ORDER

1. Order_No (PK)
2. Order_Date
3. Customer_No

ORDER_ITEM

1. Order_No (PK)
2. Item_No (PK)
3. Item_Name
4. Qty_Ordered
5. Rate_Per_Unit
6. Item_Value

- If we examine the relation ORDER_ITEM, we find that the Item_Name is not Fully Functionally dependent on the Fully Primark Key(Order_No + Item_No) as it is Functionally dependent on a part of the Primary Key i.e Item_No.
- In otherwords, we do not need to know the Order_No to determine the Item_Name; we can determine it from Item_No alone.

2nd Normal Form

Example - 1 (cont.)

26

- The disadvantage of having such a relation is that if the name of an Item changes, it has to be changed in all the ORDER_ITEM rows where it occurs.
- To remove this disadvantage, we split the relation ORDER_ITEM into the following two relations:

ORDER_ITEM

1. Order_No (PK)
2. Item_No (PK)
3. Qty_Ordered
4. Rate_Per_Unit
5. Item_Value

ITEM

1. Item_No (PK)
2. Item_Name

Both these relations are now in the **Second Normal Form** (since we have assumed that the rate_per_unit cannot be derived from the Item_No).

1st Normal Form Example

27

Un-normalized Students table:

<u>Student#</u>	AdvID	AdvName	AdvRoom	Class1	Class2
123	123A	James	555	102-8	104-9
124	123B	Smith	467	209-0	102-8

Normalized Students table:

<u>Student#</u>	AdvID	AdvName	AdvRoom	Class#
123	123A	James	555	102-8
123	123A	James	555	104-9
124	123B	Smith	467	209-0
124	123B	Smith	467	102-8

2nd Normal Form

Example - 2

28

Students table

<u>Student#</u>	AdvID	AdvName	AdvRoom
123	123A	James	555
124	123B	Smith	467

Registration table

<u>Student#</u>	Class#
123	102-8
123	104-9
124	209-0
124	102-8

Steps to convert a table to its 2nd Normal Form

29

A table is in **2nd Normal Form** if:

- It is in 1st Normal Form
- It includes no partial dependencies (where an attribute is dependent on only a part of a primary key)

The steps to convert a table to its 2nd Normal Form:

- Find and remove fields that are related to the only part of the key.
- Group the removed items in the another table.
- Assign the new table with the key **i.e.** part of a whole **composite** key.

Steps to convert a table to its 2nd Normal Form

Example - 3

30

EmpProj

1. Project_number (PK)
2. Project_name
3. Employee_number (PK)
4. Employee_name
5. Rate_category
6. Hourly_rate

Going through all the fields reveals the following:

- Project_name is only dependent on Project_number
- Employee_name, Rate_category and Hourly_rate are dependent only on Employee_number

Steps to convert a table to its 2nd Normal Form

Example - 3 (cont..)

31

EmpProj

1. Project_number (PK)
2. Project_name
3. Employee_number (PK)
4. Employee_name
5. Rate_category
6. Hourly_rate

To convert the table into the **Second Normal Form** remove and place these fields in a separate table, with the key being that part of the original key they are dependent on.

This leads to the following 3 tables:

EmpProj

1. Project_number(PK)
2. Employee_number(PK)

Emp

1. Employee_number (PK)
2. Employee_name
3. Rate_category
4. Hourly_rate

Proj

1. Project_number(PK)
2. Project_name

2nd Normal Form Example - 4

32

1) Book (authorName, title, authorAffiliation, ISBN, publisher, pubYear)

Keys: (authorName, title), ISBN

Not in 2NF as authorName \rightarrow authorAffiliation (authorAffiliation is not fully functionally dependent on the first key)

2) Student (rollNo, name, dept, sex, hostelName, roomNo, admitYear)

Keys: rollNo, (hostelName, roomNo)

Not in 2NF as hostelName \rightarrow sex

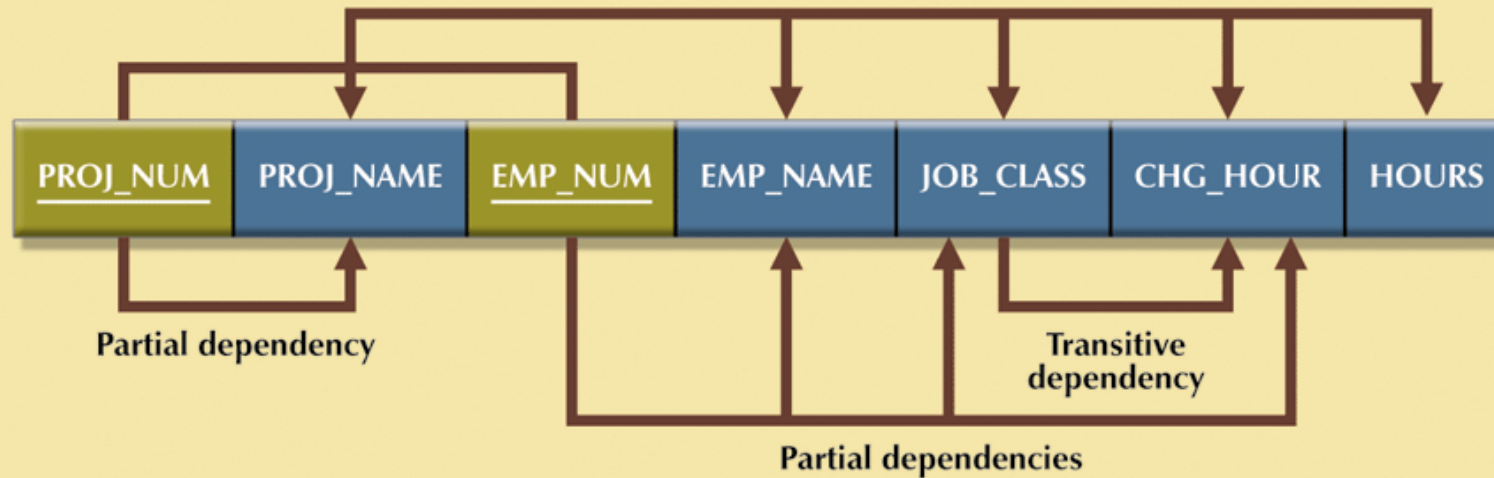
student (rollNo, name, dept, hostelName, roomNo, admitYear)

hostelDetail(hostelName, sex)

-These are both in 2NF

2nd Normal Form Example - 5

FIGURE 5.3 First normal form (1NF) dependency diagram



1NF (PROJ_NUM, EMP_NUM, PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOURS, HOURS)

PARTIAL DEPENDENCIES:

(PROJ_NUM \Rightarrow PROJ_NAME)

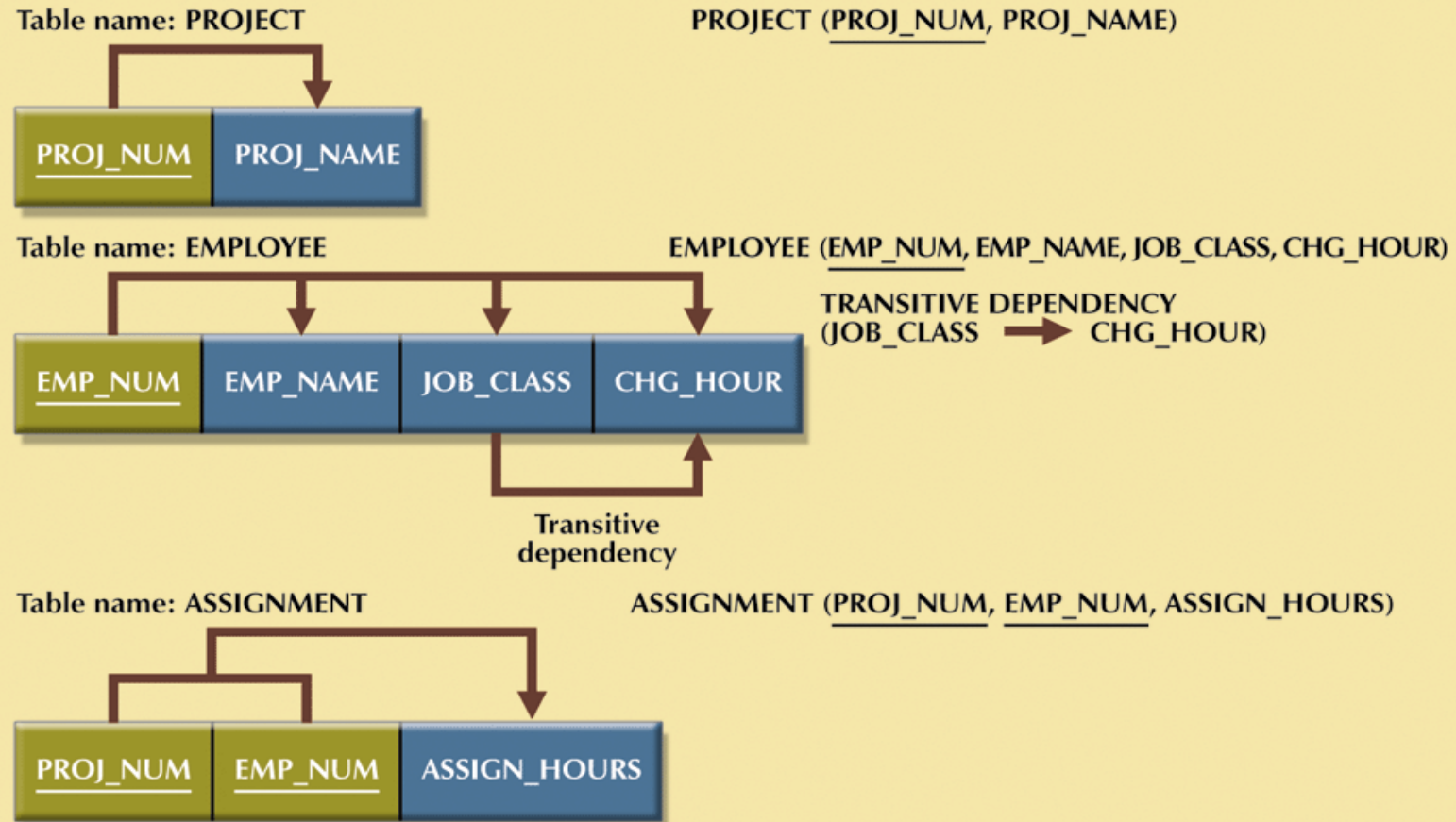
(EMP_NUM \Rightarrow EMP_NAME, JOB_CLASS, CHG_HOUR)

TRANSITIVE DEPENDENCY:

(JOB CLASS \Rightarrow CHG_HOUR)

2nd Normal Form Example - 5

FIGURE 5.4 Second normal form (2NF) conversion results



3rd Normal Form

The Requirements

35

- The requirements to satisfy the 3rd NF:
 - ▣ All requirements for 2nd NF must be met.
 - ▣ Eliminate fields that do not depend on the primary key;
 - That is, any field that is dependent not only on the primary key but also on another field must be moved to another table.

3rd Normal Form Example

Students table:

<u>Student#</u>	AdvID	AdvName	AdvRoom
123	123A	James	555
124	123B	Smith	467

Student table:

<u>Student#</u>	<u>AdvID</u>
123	123A
124	123B

Advisor table:

<u>AdvID</u>	AdvName	AdvRoom
123A	James	555
123B	Smith	467

3rd Normal Form Example Cont.

37

Students table:

<u>Student#</u>	<u>AdvID</u>
123	123A
124	123B

Registration table:

<u>Student#</u>	<u>Class#</u>
123	102-8
123	104-9
124	209-0
124	102-8

Advisor table:

<u>AdvID</u>	<u>AdvName</u>	<u>AdvRoom</u>
123A	James	555
123B	Smith	467

3rd Normal Form Example Cont.

38

FIGURE
5.5

Third normal form (3NF) conversion results



Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

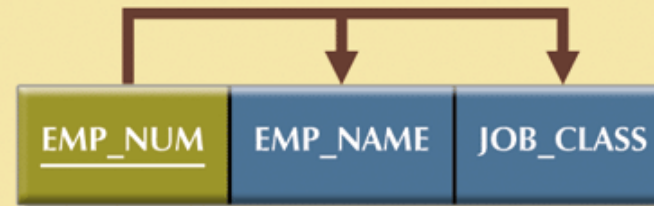


Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)



Table name: JOB

JOB (JOB_CLASS, CHG_HOUR)

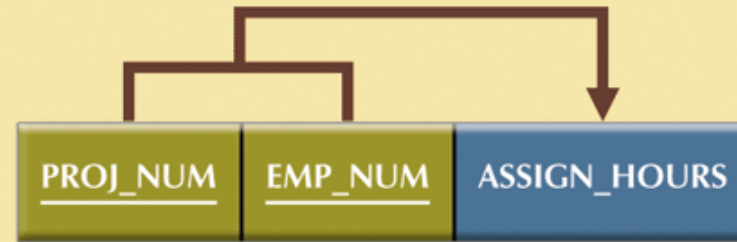


Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)