DATABASE NORMALIZATION

S.PALANIVEL
CSE /SOC

LECTURE PLAN

- ⇒ Definition of Normalization
- ⇒ Redundancy and Data Anomalies
- ⇒ Repeating Groups
- ⇒ Functional Dependency
- ⇒ Transitive Dependency
- ⇒ Stages of Normalisation
- ⇒ Example

DEFINITION

- **Database normalization** is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency.

- Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them.

REDUNDANCY AND DATA ANOMALIES

Redundant data is where we have stored the same 'information' more than once. i.e., the redundant data could be removed without the loss of information.

Example: We have the following relation that contains staff and department details:

staffNo	job	dept	dname	city
SL10	Salesman	10	Sales	Stratford
SA51	Manager	20	Accounts	Barking
DS40	Clerk	20	Accounts	Barking
OS45	Clerk	30	Operations	Barking

Such 'redundancy' could lead to the following 'anomalies'

Insert Anomaly: We can't insert a dept without inserting a member of staff that works in that department

Update Anomaly: We could change the name of the dept that SA51 works in without simultaneously changing the dept that DS40 works in.

Deletion Anomaly: By removing employee SL10 we have removed all information pertaining to the Sales dept.

REPEATING GROUPS

A repeating group is an attribute (or set of attributes) that can have more than one value for a primary key value.

Example: We have the following relation that contains staff and department details and a list of telephone contact numbers for each member of staff.

staffNo	**job**	**dept**	**dname**	**city**	**contact number**
SL10	Salesman	10	Sales	Stratford	**018111777, 018111888, 079311122**
SA51	Manager	20	Accounts	Barking	**017111777**
DS40	Clerk	20	Accounts	Barking	
OS45	Clerk	30	Operations	Barking	**079311555**

Repeating Groups are not allowed in a relational design, since all attributes have to be ‘atomic’ - i.e., there can only be one value per cell in a table!

NEED & SOLUTION

- A formal **tool** for analysis of relational schemas that enables us to detect the above-mentioned problems.
- The single most important concept in relational schema design theory is that of a **functional dependency**.

FUNCTIONAL DEPENDENCY

Formal Definition: Attribute B is functionally dependant upon attribute A *(or a collection of attributes)* if a value of A determines a single value of attribute B at any one time.

Formal Notation: **A → B** This should be read as **'A determines B'** or **'B is functionally dependant on A'**. A is called the *determinant* and B is called the *object of the determinant*.

Example:

staffNo	job	dept	dname
SL10	Salesman	10	Sales
SA51	Manager	20	Accounts
DS40	Clerk	20	Accounts
OS45	Clerk	30	Operations

Functional Dependencies

staffNo → job
staffNo → dept
staffNo → dname
dept → dname

FUNCTIONAL DEPENDENCY

Compound Determinants: If more than one attribute is necessary to determine another attribute in an entity, then such a determinant is termed a composite determinant.

Full Functional Dependency: Only of relevance with composite determinants. This is the situation when it is necessary to use all the attributes of the composite determinant to identify its object uniquely.

Example:

order#	line#	qty	price
A001	001	10	200
A002	001	20	400
A002	002	20	800
A004	001	15	300

Full Functional Dependencies

(Order#, line#) $\rightarrow$ qty

(Order#, line#) $\rightarrow$ price

FUNCTIONAL DEPENDENCY

Partial Functional Dependency: This is the situation that exists if it is necessary to only use a subset of the attributes of the composite determinant to identify its object uniquely.

Example:

student#	unit#	room	grade
9900100	A01	TH224	2
9900010	A01	TH224	14
9901011	A02	JS075	3
9900001	A01	TH224	16

Repetition of data!

Full Functional Dependencies

(student#, unit#) $\rightarrow$ grade

Partial Functional Dependencies

unit# $\rightarrow$ room

TRANSITIVE DEPENDENCY

Definition: A transitive dependency exists when there is an intermediate functional dependency.

Formal Notation: If **A $\rightarrow$ B** and **B $\rightarrow$ C**, then it can be stated that the following transitive dependency exists: **A $\rightarrow$ B $\rightarrow$ C**

Example:

staffNo	job	dept	dname
SL10	Salesman	10	Sales
SA51	Manager	20	Accounts
DS40	Clerk	20	Accounts
OS45	Clerk	30	Operations

Repetition of data!

Transitive Dependencies

staffNo $\rightarrow$ dept
dept $\rightarrow$ dname

staffNo $\rightarrow$ dept $\rightarrow$ dname

NORMALISATION - RELATIONAL MODEL

In order to comply with the relational model it is necessary to 1) remove repeating groups and 2) avoid redundancy and data anomalies by remoting partial and transitive functional dependencies.

Relational Database Design: All attributes in a table must be atomic, and solely dependant upon the fully primary key of that table.

NORMALISATION ACHIEVES THIS!

Stages of Normalisation

- Unnormalised **(UDF)**
  - Remove repeating groups
- First normal form **(1NF)**
  - Remove partial dependencies
- Second normal form **(2NF)**
  - Remove transitive dependencies
- Third normal form **(3NF)**
  - Remove remaining functional dependency anomalies
- Boyce-Codd normal form **(BCNF)**
  - Remove multivalued dependencies
- Fourth normal form **(4NF)**
  - Remove remaining anomalies
- Fifth normal form **(5NF)**

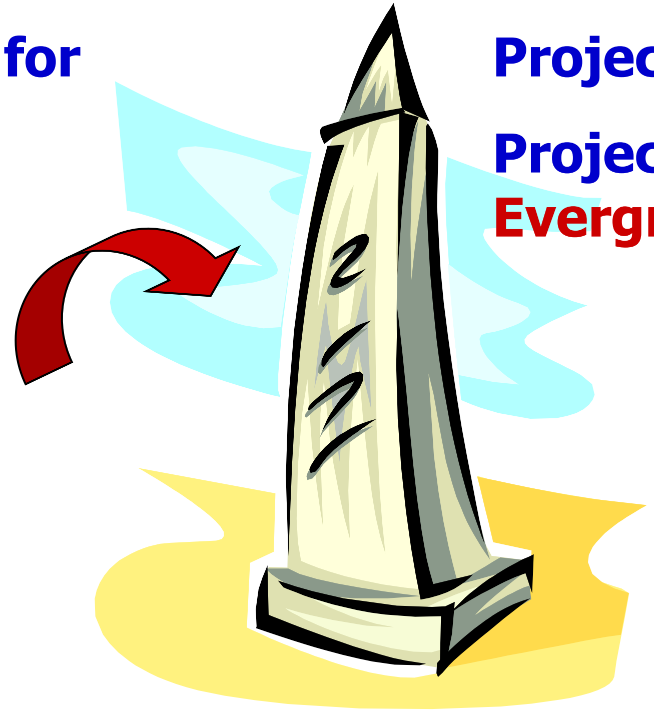
DATABASE TABLES AND NORMALIZATION

- **The Need for Normalization**
  - **Case of a Construction Company**
    - **Building project -- Project number, Name, Employees assigned to the project.**
    - **Employee -- Employee number, Name, Job classification**
    - **The company charges its clients by billing the hours spent on each project. The hourly billing rate is dependent on the employee's position.**
    - **Periodically, a report is generated.**
    - **The table whose contents correspond to the reporting requirements is shown in Table 5.1.**

SCENARIO

A few employees works for one project.

Employee Num : 101, 102, 103, 105



Project Num : 15

Project Name : Evergreen

SAMPLE FORM

Project Num : 15

Project Name : Evergreen

Emp Num	Emp Name	Job Class	Chr Hours	Hrs Billed	Total
101					
102					
103					
105					

TABLE 5.1 A SAMPLE REPORT LAYOUT

PROJ. NUM.	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS.	CHG/ HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$84.50	23.8	$2,011.10
		101	John G. News	Database Designer	$105.00	19.4	$2,037.00
		105	Alice K. Johnson *	Database Designer	$105.00	35.7	$3,748.50
		106	William Smithfield	Programmer	$35.75	12.6	$450.45
		102	David H. Senior	Systems Analyst	$96.75	23.8	$2,302.65
				Subtotal			**$10,549.70**
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6	$1,183.26
		118	James J. Frommer	General Support	$18.36	45.3	$831.71
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4	$3,134.70
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0	$2,021.80
				Subtotal			**$7,171.47**
22		105	Alice K. Johnson	Database Designer	$105.00	64.7	$6,793.50
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4	$4,682.70
		113	Delbert K. Joenbrood*	Applications Designer	$48.10	23.6	$1,135.16
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0	$591.14
		106	William Smithfield	Programmer	$35.75	12.8	$457.60
				Subtotal			**$13,660.10**
25		107	Maria D. Alonzo	Programmer	$35.75	24.6	$879.45
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8	$4,431.15
		101	John G. News *	Database Designer	$105.00	56.3	$5,911.50
		114	Annelise Jones	Applications Designer	$48.10	33.1	$1,592.11
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6	$2,283.30
		118	James J. Frommer	General Support	$18.36	30.5	$559.98
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4	$1,902.33
				Subtotal			**$17,559.82**
				Total			**48,941.09**

Note: * indicates project leader

Table Structure Matches the Report Format

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
		118	James J. Frommer	General Support	$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0
		106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
		101	John G. News *	Database Designer	$105.00	56.3
		114	Annelise Jones	Applications Designer	$48.10	33.1
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6
		118	James J. Frommer	General Support	$18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

FIGURE 5.1 A TABLE WHOSE STRUCTURE MATCHES THE REPORT FORMAT

DATABASE TABLES AND NORMALIZATION

- **Problems with the Figure 5.1**
  - **The project number is intended to be a primary key, but it contains nulls.**

DATABASE TABLES AND NORMALIZATION

- **Conversion to First Normal Form**
  - **A relational table must not contain repeating groups.**
  - **Repeating groups can be eliminated by adding the appropriate entry in at least the primary key column(s).**

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8

FIGURE 5.2 THE EVERGREEN DATA

Data Organization: First Normal Form

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HO
15	Evergreen	03	June E. Arbough	Elect. Engineer	$84.50	
		01	John G. News	Database Designer	$105.00	
		05	Alice K. Johnson *	Database Designer	$105.00	
		06	William Smithfield	Programmer	$35.75	
		02	David H. Senior	Systems Analyst	$96.75	
18	Amber Wave	14	Annelise Jones	Applications Designer	$48.10	
		18	James J. Frommer	General Support	$18.36	
		04	Anne K. Ramoras *	Systems Analyst	$96.75	
		12	Darlene M. Smithson	DSS Analyst	$45.95	
22	Rolling Tide	05	Alice K. Johnson	Database Designer	$105.00	
		04	Anne K. Ramoras	Systems Analyst	$96.75	
		13	Delbert K. Joenbrood *	Applications Designer	$48.10	
		11	Geoff B. Wabash	Clerical Support	$26.87	
		06	William Smithfield	Programmer	$35.75	
25	Starflight	07	Maria D. Alonzo	Programmer	$35.75	
		15	Travis B. Bawangi	Systems Analyst	$96.75	
		01	John G. News *	Database Designer	$105.00	
		14	Annelise Jones	Applications Designer	$48.10	33.1
		08	Ralph B. Washington	Systems Analyst	$96.75	23.6
		18	James J. Frommer	General Support	$18.36	30.5
		12	Darlene M. Smithson	DSS Analyst	$45.95	41.4

FIGURE 5.1 A T **Before** HES THE REPORT FORMAT

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
15	Evergreen	101	John G. News	Database Designer	$105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	$105.00	35.7
15	Evergreen	106	William Smithfield	Programmer	$35.75	12.5
15	Evergreen	102	David H. Senior	Systems Analyst	$96.75	23.9
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
18	Amber Wave	118	James J. Frommer	General Support	$18.36	45.3
18	Amber Wave	104	Anne K. Ramoras *	Systems Analyst	$96.75	32.1
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	$96.75	48.9
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	$26.87	22.5
22	Rolling Tide	106	William Smithfield	Programmer	$35.75	12.1
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.7
25	Starflight	115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
25	Starflight	101	John G. News *	Database Designer	$105.00	56.3
25	Starflight	114	Annelise Jones	Applications Designer	$48.10	33.1
25	Starflight	108	Ralph B. Washington	Systems Analyst	$96.75	23.9
25	Starflight	118	James J. Frommer	General Support	$18.36	30.2
25	Starflight	112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

FIGURE 5.3 **After** RST NORMAL FORM

FIRST NORMAL FORM (1 NF)

- **1NF Definition**
  - **The term first normal form (1NF) describes the tabular format in which:**
    - **All the key attributes are defined.**
    - **There are no repeating groups in the table.**
    - **All attributes are dependent on the primary key.**

DEPENDENCY DIAGRAM

- **Dependency Diagram**
  - **The primary key components are bold, underlined, and shaded in a different color.**
  - **The arrows above entities indicate all desirable dependencies, i.e., dependencies that are based on PK.**
  - **The arrows below the dependency diagram indicate less desirable dependencies -- partial dependencies and transitive dependencies.**

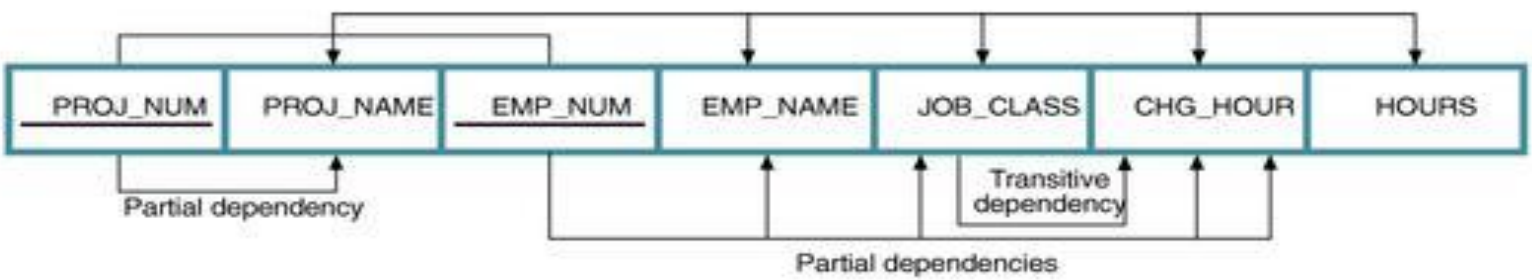


FIGURE 5.4 A DEPENDENCY DIAGRAM: FIRST NORMAL FORM (1NF)

SECOND NORMAL FORM (2 NF)

DEPENDENCY DIAGRAM

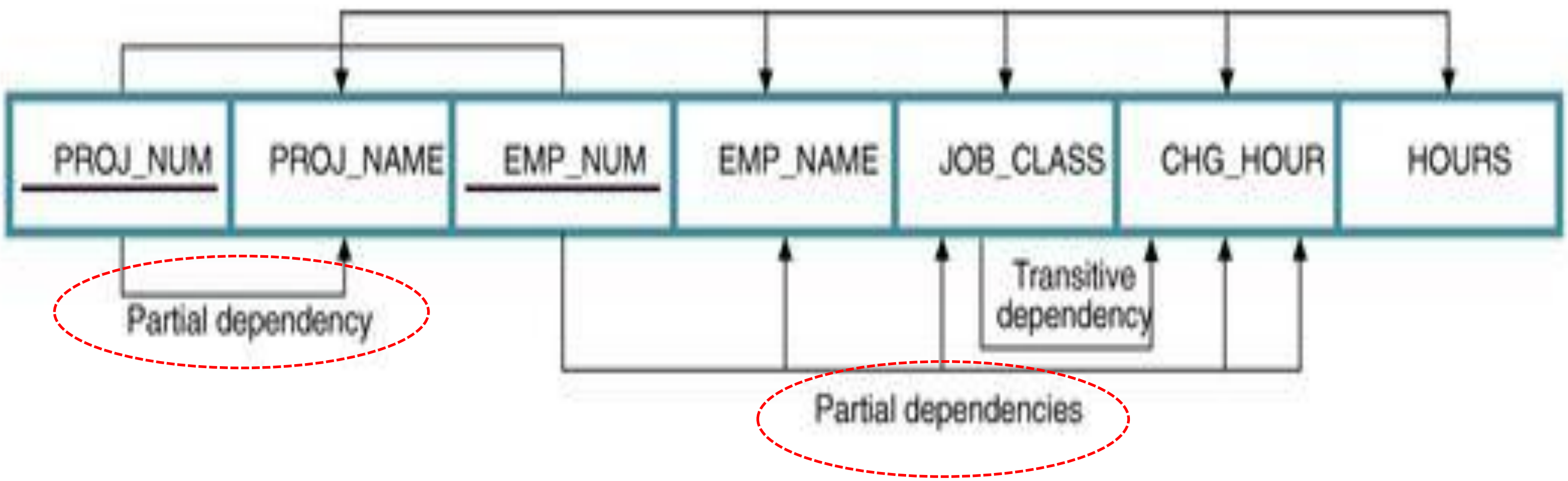


FIGURE 5.4 A DEPENDENCY DIAGRAM: FIRST NORMAL FORM (1NF)

SECOND NORMAL FORM (2 NF)

- **Conversion to Second Normal Form**
  - **Starting with the 1NF format, the database can be converted into the 2NF format by**
    - **Writing each key component on a separate line, and then writing the original key on the last line and**
    - **Writing the dependent attributes after each new key.**

PROJECT (<u>PROJ_NUM</u>, PROJ_NAME)

EMPLOYEE (<u>EMP_NUM</u>, EMP_NAME, JOB_CLASS, CHG_HOUR)

ASSIGN (<u>PROJ_NUM, EMP_NUM</u>, HOURS)

Dependency Diagram

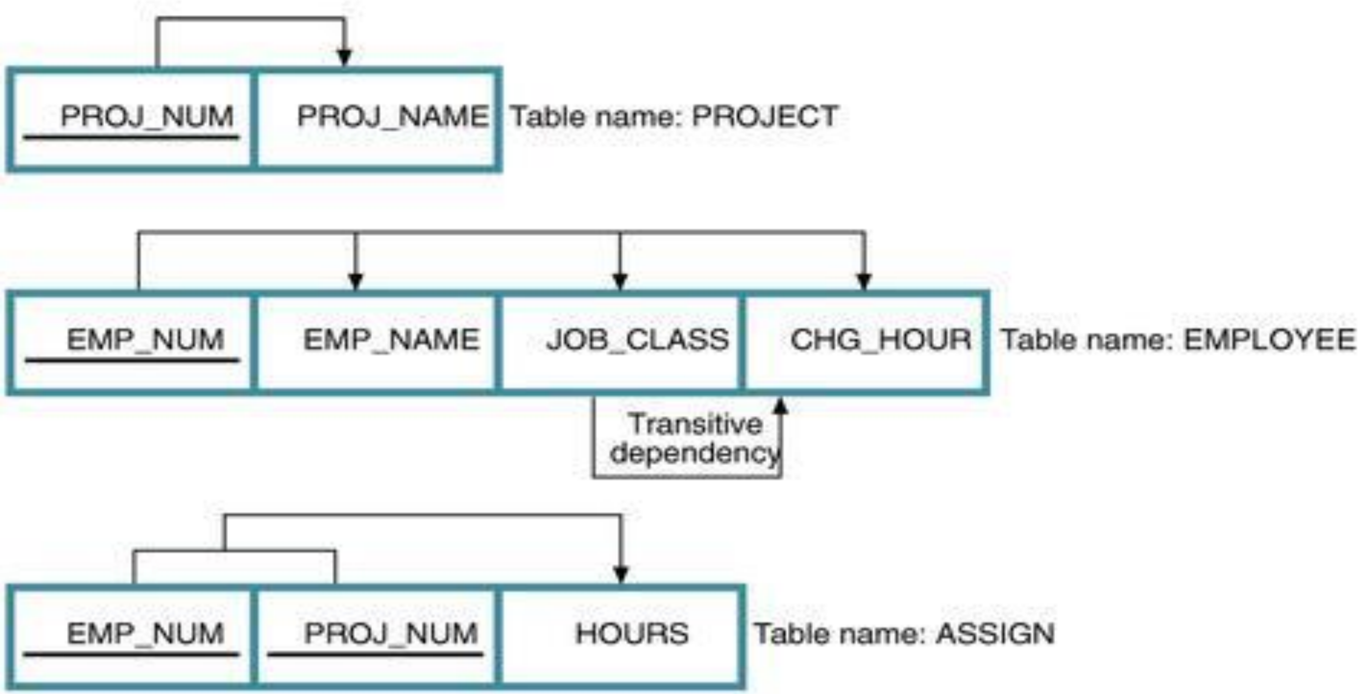


FIGURE 5.5 SECOND NORMAL FORM (2NF) CONVERSION RESULTS

SECOND NORMAL FORM (2 NF)

A table is in 2NF if:

- **It is in 1NF and**
- **It includes no partial dependencies; that is, no attribute is dependent on only a portion of the primary key.**

  (It is still possible for a table in 2NF to exhibit transitive dependency; that is, one or more attributes may be functionally dependent on nonkey attributes.)

THIRD NORMAL FORM (3 NF)

Dependency Diagram

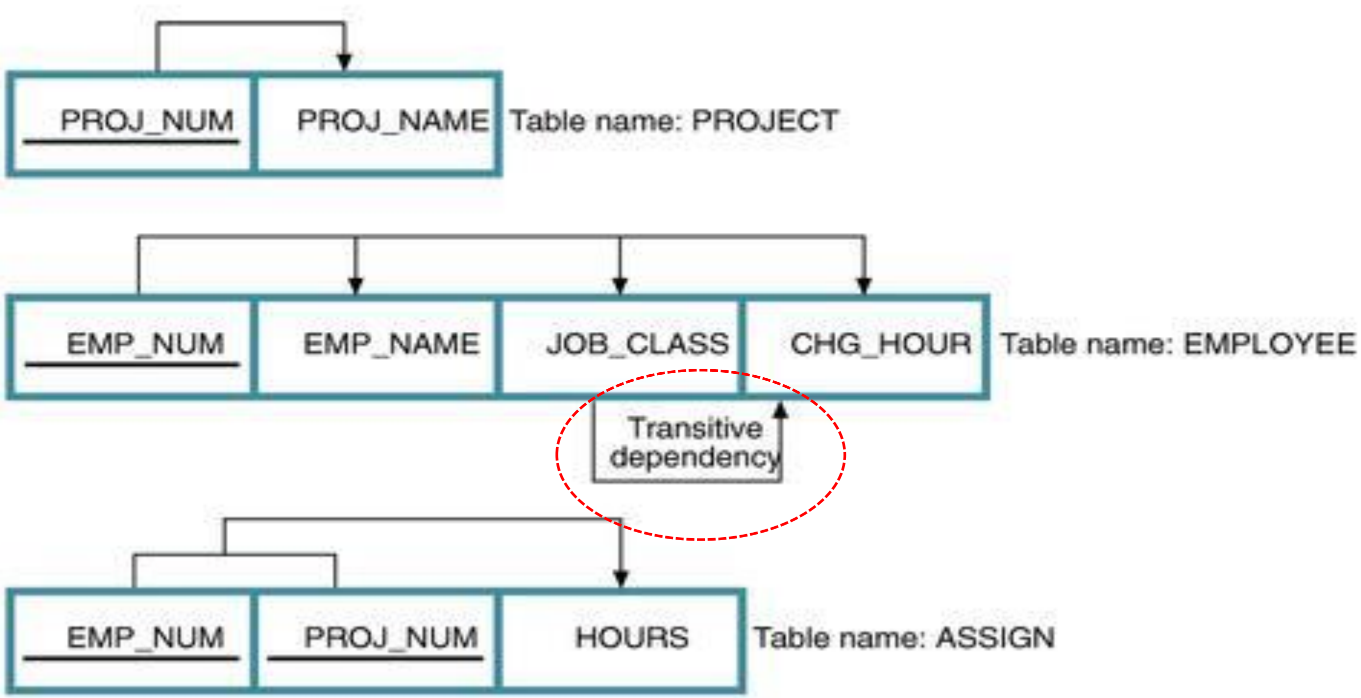


FIGURE 5.5 SECOND NORMAL FORM (2NF) CONVERSION RESULTS

THIRD NORMAL FORM (3 NF)

- Conversion to Third Normal Form
  - **Create a separate table with attributes in a transitive functional dependence relationship.**

PROJECT (PROJ_NUM, PROJ_NAME)

ASSIGN (PROJ_NUM, EMP_NUM, HOURS)

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)

JOB (JOB_CLASS, CHG_HOUR)

THIRD NORMAL FORM (3 NF)

- **3NF Definition**
  - **A table is in 3NF if:**
    - **It is in 2NF and**
    - **It contains no transitive dependencies.**

FIGURE 5.6 THE COMPLETED DATABASE

The Completed Database

THIRD NORMAL FORM

- No Repeating Groups

- No Partial Dependencies

- No Transitive Dependencies.

TERMINOLOGIES

- Super Key
- Candidate Key
- Prime Attribute
- NonPrime Attribute

SUPER KEY

SUPER KEY

- **Definition:**

- A super key
  - is a set of one or more columns (attributes)
  - to uniquely identify rows in a table.

This is an example !

EXAMPLE - EMPLOYEE

Emp_SSN	Emp_Number	Emp_Name
123456789	226	Steve
999999321	227	Ajeet
888997212	228	Chaitanya
777778888	229	Robert

SUPER KEYS

- {Emp_SSN}
- {Emp_Number}
- {Emp_SSN, Emp_Number}
- {Emp_SSN, Emp_Name}
- {Emp_Number, Emp_Name}
- {Emp_SSN, Emp_Number, Emp_Name}

- All of the above sets are able to uniquely identify rows of the employee table.

CANDIDATE KEY

CANDIDATE KEY

- Candidate keys are selected from the set of super keys.
- The only thing we take care while selecting candidate key is: It should not have any redundant data.

CANDIDATE KEYS

- They are the minimal super keys with no redundant Data.

- {Emp_SSN}
- {Emp_Number}

- Only these two sets are candidate keys
- All other sets are having redundant Data.

PRIME AND NONPRIME ATTRIBUTES

PRIME AND NONPRIME ATTRIBUTES

- Prime
  - Attributes that are chosen to uniquely identify any records in a table.
  - The values cannot be duplicated
- NonPrime
  - Attributes other than the prime attributes.
  - They can store a value many times.

BOYCE CODD NORMAL FORM (BCNF)

BOYCE CODD NORMAL FORM (BCNF)

- It is an advance version of 3NF
- It is also referred as 3.5NF.
- BCNF is stricter than 3NF.

BOYCE CODD NORMAL FORM (BCNF)

- A table complies with BCNF
  - If it is in 3NF
  - For every **functional dependency** X->Y
  - X should be the super key of the table.

EXAMPLE

- Suppose there is a company
- Wherein
- employees work in **more than one department**.

- They store the data in the following format:

COMPANY TABLE

emp_id	emp_nationality	emp_dept	dept_type	dept_no_of_emp
1001	Austrian	Production and planning	D001	200
1001	Austrian	stores	D001	250
1002	American	design and technical support	D134	100
1002	American	Purchasing department	D134	600

BCNF

- **Functional dependencies in the table above:**
  emp_id -> emp_nationality
  emp_dept -> {dept_type, dept_no_of_emp}

- **key**: {emp_id, emp_dept}

- The table is not in BCNF.

BCNF

- To make the table comply with BCNF we can break the table in three tables as follows:

Emp_Nationality Table:

emp_id	emp_nationality
1001	Austrian
1002	American

EMP_DEPT TABLE

emp_dept	dept_type	dept_no_of_emp
Production and planning	D001	200
stores	D001	250
design and technical support	D134	100
Purchasing department	D134	600

EMP_DEPT_MAPPING TABLE

emp_id	emp_dept
1001	Production and planning
1001	stores
1002	design and technical support
1002	Purchasing department

BCNF

- **Functional dependencies:**
  emp_id -> emp_nationality
  emp_dept -> {dept_type, dept_no_of_emp}

- **keys:**
  For first table: emp_id
  For second table: emp_dept
  For third table: {emp_id, emp_dept}

- This is now in **BCNF**
- as in both the functional dependencies **left side part is a key**.

FOURTH NORMAL FORM

FOURTH NORMAL FORM

- **Fourth normal form** (4NF) is Introduced by Ronald Fagin in 1977

- 4NF is the next level of normalization after Boyce–Codd **normal form** (BCNF).

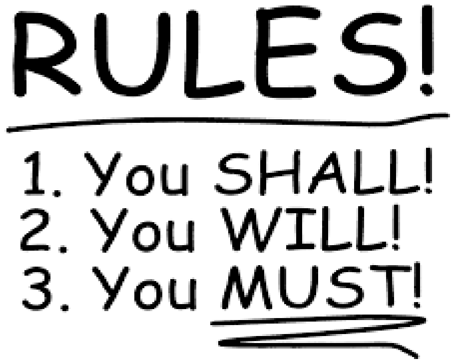
MULTI-VALUED DEPENDENCY

- Fourth Normal Form comes into picture when **Multi-valued Dependency** occur in any relation.

- Multi-valued Dependency – What?

- How to remove it and make any table satisfy the fourth normal form?

RULES FOR 4TH NORMAL FORM



- For a table to satisfy the Fourth Normal Form, it should satisfy the following two conditions:

- It should be in the **Boyce-Codd Normal Form**.

- The table should not have any **Multi-valued Dependency**.

WHAT IS MULTI-VALUED DEPENDENCY?

- A table is said to have multi-valued dependency, if the following conditions are true.

- For a dependency A → B,
  - if for a single value of A,
  - multiple value of B exists,
  - then the table may have multi-valued dependency.

WHAT IS MULTI-VALUED DEPENDENCY?

- Also, a table should <u>**have at-least 3 columns**</u> for it to have a multi-valued dependency.

- And, for a relation R(A,B,C), if there is a multi-valued dependency between, A and B, then <u>**B and C should be independent of each other.**</u>

- If all these conditions are true for any relation(table), it is said to have multi-valued dependency.

EXAMPLE
COLLEGE ENROLLMENT TABLE

just
another
example

- college enrolment table with columns s_id, course and hobby.

s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
2	C#	Cricket
2	Php	Hockey

EXAMPLE
COLLEGE ENROLLMENT TABLE

- In the table

- Student with s_id **1** has opted
  - for two courses, **Science** and **Maths**, and
  - has two hobbies, **Cricket** and **Hockey**.

- S_id->->Course
- S_id->->Hobby

- ->-> Notation for MultivaluedDependancy just another example

EXAMPLE

- Well the two records for student with s_id 1, will give rise to two more records, as shown below, because for one student, two hobbies exists, hence along with both the courses, these hobbies should be specified.

s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
1	Science	Hockey
1	Maths	Cricket

And, in the table above, there is no relationship between the columns **course and hobby**.

They are independent of each other.

So there is multi-value dependency, which leads to un-necessary repetition of data and other anomalies as well.

HOW TO SATISFY 4TH NORMAL FORM?

- To make the above relation satisfy the 4th normal form
- we can decompose the table into 2 tables.

CourseOpted Table

s_id	course
1	Science
1	Maths
2	C#
2	Php

HOW TO SATISFY 4TH NORMAL FORM?

And, **Hobbies Table**,

s_id	hobby
1	Cricket
1	Hockey
2	Cricket
2	Hockey

FIFTH NORMAL FORM

FIFTH NORMAL FORM

- 5NF is also known as Project-Join normal form (PJNF).

- A relation is in 5NF if it is in 4NF
- It cannot be further non loss decomposed (join dependency)

FIFTH NORMAL FORM

- A relation

- Decomposed into two relations must have **loss-less join Property**

- Which ensures that **no spurious or extra tuples** are generated

- When relations are reunited through a **natural join**.

FIFTH NORMAL FORM

- **Example –** Consider the Company schema, with a case as "if a company makes a product and an agent is an agent for that company, then he always sells that product for the company".

- Under these circumstances, the ACP table is shown as follows:

FIFTH NORMAL FORM

Table – ACP

AGENT	COMPANY	PRODUCT
A1	PQR	Nut
A1	PQR	Bolt
A1	XYZ	Nut
A1	XYZ	Bolt
A2	PQR	Nut

- The relation ACP is again decompose into 3 relations

FIFTH NORMAL FORM

Table – R1

AGENT	COMPANY
A1	PQR
A1	XYZ
A2	PQR

FIFTH NORMAL FORM

Table – R2

AGENT	PRODUCT
A1	Nut
A1	Bolt
A2	Nut

FIFTH NORMAL FORM

Table – R3

COMPANY	PRODUCT
PQR	Nut
PQR	Bolt
XYZ	Nut
XYZ	Bolt

FIFTH NORMAL FORM

- Result of Natural Join of R1 and R3 over ‘Company’ and then Natural Join of R13 and R2 over ‘Agent’ and ‘Product’ will be table **ACP**.

FIFTH NORMAL FORM

- Hence, in this example,
- The decomposition of ACP is a lossless join decomposition.

- Therefore, the relation is in 5NF as it does not violate the property of lossless join.

DENORMALIZATION

DENORMALIZATION

- **Normalization is only one of many database design goals.**

- **Normalized (decomposed) tables require additional processing, reducing system speed.**

- **Normalization purity is often difficult to sustain in the modern database environment.**

- **The conflict between design efficiency, information requirements, and processing speed are often resolved through compromises that include denormalization.**