INFS5710 IT Infra. for BA

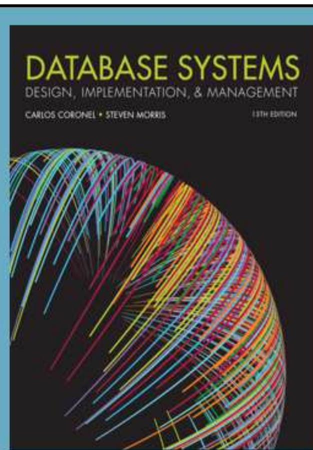
Normalisation

Agenda

- **Normalisation**
- **Functional Dependencies**
- **Normal Forms**
  - 1NF
  - 2NF
  - 3NF
  - BCNF
  - 4NF
- **De-normalisation**

This week, we will be looking at more on Normalisation of Database Tables.

Normalisation and Normalisation are both fine – one is English and the other is American English – it is ‘s’ or ‘z’



Chapter 6

Normalisation of Database Tables

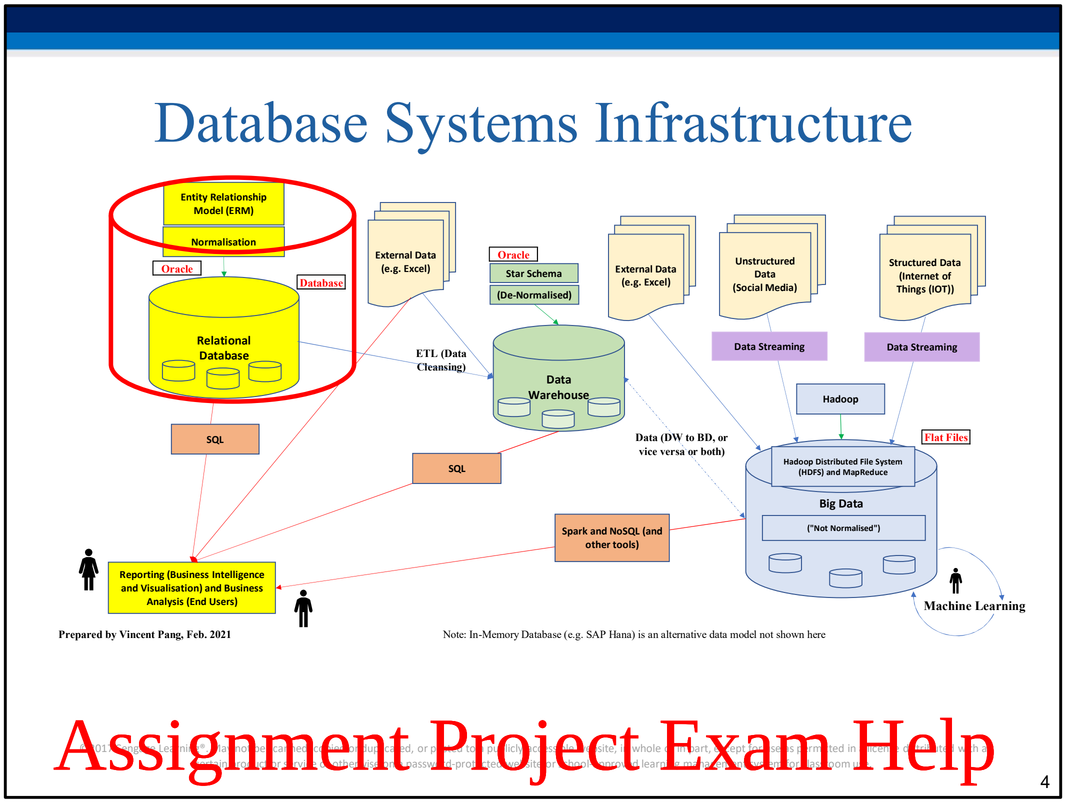
6-1 to 6-8

pp. 200-230

3

This week, we will be looking at more on Normalisation of Database Tables.

Normalisation and Normalisation are both fine – one is English and the other is American English – it is 's' or 'z'



We are looking at Normalisation. Normalisation and ERD go together when designing a database. We have look at conceptual data modelling, i.e. ERD.

Remember, **Entity-Relationship (ER) modelling** is Top-down approach. Begins by looking for the data groups in the system.

On the other hand, **Normalisation** is a Bottom-up approach. Begins by looking at the smallest individual items of data recorded by the system.

When I was in my doing my undergraduate, I asked myself who will use normalisation. Three weeks into my first job, my boss asked me to evaluate all the databases the company had at that point in time to see if we need to do anything about normalisation.

So normalisation is an important part of database design.

The Needs and Outcomes of Normalisation

- **You <u>need</u> the process of normalisation is when your design a new database structure**
  - Analyzes the relationship among the attributes within each entity
  - Determines if the structure can be improved
- **Improves the existing data structure and creates an appropriate database design**

- **The <u>outcome</u> of normalisation will result in a well-structured relation.** A well-structured relation is:
  - a relation that contains minimal data redundancy and
  - allows users to insert, delete, and update rows without causing data inconsistencies and anomalies, i.e. reduce data anomalies

5

The key point of normalisation is to help building a database structure. So you will go from one normal form to the next normal form based on the rules.

The need for normalisation include identifying business rules, identifying and defining business and data constraints, defining functional dependencies, identifying entities and relationships and eliminating multivalued attributes

You need to Improve the existing data structure and create an appropriate database design.

The outcome is what you want from normalisation **as stated on slide**. You can insert, delete and update rows without any issue.

We will now cover in more depth.

Normalisation (1)

Normalisation is a process for evaluating and correcting table structures to **minimize data redundancies**, thereby **reducing the likelihood of data anomalies**.

- **Normalisation** is …
  - a process for converting a relation to a **standard (normal) form**.
  - a **process** that is accomplished in **stages**.
  - a technique that is used to **define "goodness"** (or "badness") of a relation.
  - to minimise or **eliminate redundancy** (duplication of data).
  - to **prevent data inconsistencies** from update, deletion, and insertion **anomalies**.
  - to decompose a relation/table into **smaller components.**
  - to **recapture the precise content** of the original relation/table.
  - to build data structures that have some **desirable** ("good") **properties**.
  - Based on paper: **Codd (1971).**

6

The key points of normalisation are shown.

The keys point are **to reduce or minimised data redundancies, and prevent data inconsistencies**. We examine the table structures to see if we could minimise or further minimise data redundancies.

We have to ensure the data redundancy is eliminated or minimised (we will talk why there might be data redundancy on the next slide).

Normalisation (2)

- **Redundancy**
  - **Redundancy** occurs when data about a **one entity is recorded more than once** in a database.
  - Database designers aim to **reduce redundancy** (i.e., database should not store same data several times) to save space and prevent problems.
  - Evaluating and correcting table structures to minimise data redundancies

- **Anomalies**
  - **Insertion Anomaly** – adding new rows forces user to create duplicate data
  - **Deletion Anomaly** – deleting rows may cause a loss of data that would be needed for other future rows
  - **Modification (Update) Anomaly** – changing data in a row forces changes to other rows because of duplication

7

Data redundancy can be generated across different tables if they are not controlled, i.e. normalised.

If **data redundancy exists, then there will be anomalies.**

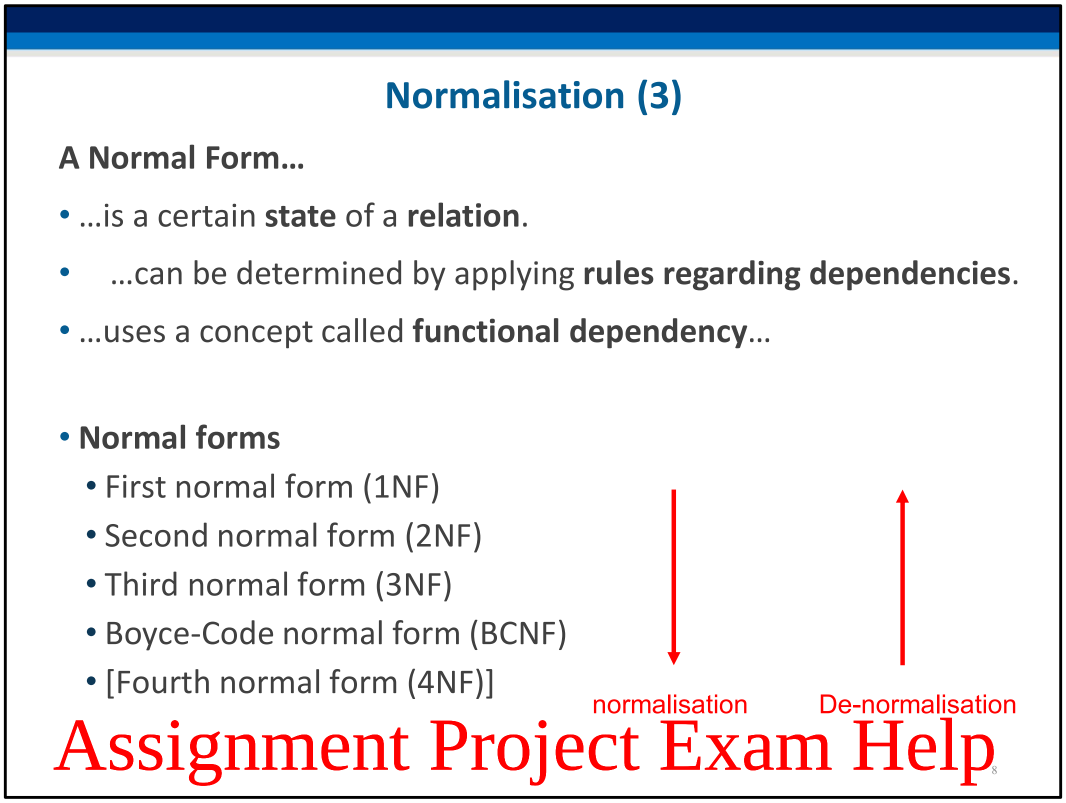
For example, if tables are not normalised, then data redundancies will happen, and you will have issues with anonmalies.

Let’s say, your have a private email attribute/column in your student table. You also have your private email address in course_enrol table for all the courses you have enrolled. Thus, you have data redundancy because you have entered private email address column in two tables, and you have to enter the private email address every time you enrol a new course.

Thus, an anomaly refers to the situation when there is a change of an employee information, there are many columns/rows about this employee that must be manually changed/updated.

There are three types of anomalies as stated on the slide.

If you change your private email address, then you have to change the student table and all the course records in the course_enrol table. This is an example of **modification anomaly**.



Normalisation is a structural point of view of normal forms.

A Normal Form…

a) …is a certain **state** of a **relation**.
b) …can be determined by applying **rules regarding dependencies**.
c) …uses a concept called **functional dependency.**

The last two points, (b) and (c), are the foundation of normalisation, which will be covered later.

A normal form is up to about sixth normal form and there are other normal forms such as BCNF. In this course, we only interested in the first 3 normal forms and Boyce-Codd normal form, and we also touch on fourth normal form. **Higher normal forms are better than lower normal forms**.

Data and business rules determine the normal form we will adopt.

To **normalise** the data, we go from 1NF to 3NF etc., that is to reduce data redundancies. Most of the normalised tables are in 3NF and BCNF.

Denormalisation, on the hand, we go the other way round from bottom to top. The data becomes more redundant.

Why? You might ask. The reason is when the data is denormalized, you can access the data much faster. Therefore, The first half of this course, we teach you how to reduce data redundancies, when it comes to data warehouse and big data, we will then teach about demormalisation to become greater data redundancy. There are good reasons why we want to take this action! **Denormalisation** helps to improve greater performance with greater data redundancy!

Sometimes, we might just do that for a few tables, purely to improve run-time performance.

Normal Forms

Table 6.2: Normal Forms		
Normal Form	**Characteristic**	**Section**
First normal form (1NF)	Table format, no repeating groups, and PK identified	6-3a
Second normal form (2NF)	1NF and no partial dependencies	6-3b
Third normal form (3NF)	2NF and no transitive dependencies	6-3c
Boyce-Codd normal form (BCNF)	Every determinant is a candidate key (special case of 3NF)	6-6a
Fourth normal form (4NF)	3NF and no independent multivalued dependencies	6-6b

These are the rules of for Normal forms:

We will go through some of these terms such as partial dependencies and transitive dependencies on the next slide

1. In the **First normal form (1NF)**, you will try to ensure the data is cleaned and table is formatted, no repeating groups, and PK identified
2. In the **Second normal form (2NF)**, it has to be in 1NF and no partial dependencies
3. In the **Third normal form (3NF)**, it has to be in 2NF and no transitive dependencies
4. In **Boyce-Codd normal form (BCNF)**, every determinant is a candidate key, which is a special case of 3NF
5. In **Fourth normal form (4NF)**, it has to be in 3NF and no independent multivalued dependencies

Functional Dependency

Functional Dependency (FD)

Functional Dependencies …

…are **relationships between attributes** in a relation.

…are the **semantics of the attributes** in a relation.

…**can be inferred in a systematic way** by applying a set of **inference rules** (next slides).

As stated on slide …

We will now go through FD…

Table 6.3: Functional Dependence Concepts	
Concept	**Definition**
Functional dependence	The attribute B is fully functionally dependent on the attribute A if each value of A determines one and only one value of B. Example: PROJ_NUM S PROJ_NAME (read as PROJ_NUM functionally determines PROJ_NAME) In this case, the attribute PROJ_NUM is known as the determinant attribute, and the attribute PROJ_NAME is known as the dependent attribute.
Functional dependence (generalized definition)	Attribute A determines attribute B (that is, B is functionally dependent on A) if all (generalized definition) of the rows in the table that agree in value for attribute A also agree in value for attribute B.
Fully functional dependence (composite key)	If attribute B is functionally dependent on a composite key A but not on any subset of that composite key, the attribute B is fully functionally dependent on A.

A $\xrightarrow{\text{functionally determines}}$ B

determinant

dependent

12

Now, we look at the term **functional dependence (FD)**

As stated in the slide…

You can say,

A functionally determines B

Or

B is functionally dependent on A

If **A→B,** i.e. **A determines B**:

- if A then B; if the premise A holds, then the conclusion B holds;
- B can be inferred from A; A implies B.

In a relation R, an attribute A (or set of attributes) determines an attribute B (or set of attributes) if the values of A uniquely identify the values of B in all cases. In other words, B is "functionally dependent" on A (or A functionally determines B).

For example,

(1) Postcode → State

i.e. “2052”→“NSW”, but not “2052”→”VIC”

(1) PROJ_NUM -> PROJ_NAME

The attribute PROJ_NUM (e.g. 123) is known as the determinant attribute, and the attribute PROJ_NAME (“Project Terminator”) is known as the dependent attribute.

Functional Dependency & Normalisation

- Two types of functional dependencies:
- A **partial dependency** exists when there is a functional dependence in which <u>**the determinant is only part of the primary key**</u>.
  - For example, if (A, B) -> (C, D), B -> C, and (A, B) is the primary key, then the functional dependence B -> C is a partial dependency because only part of the primary key (B) is needed to determine the value of C. Partial dependencies tend to be straight-forward and easy to identify.
- A **transitive dependency** exists when there are functional dependencies such that X -> Y, Y -> Z, and X is the primary key. In that case, the dependency X -> Z is a transitive dependency because X determines the value of Z via Y.
  - Unlike partial dependencies, transitive dependencies are more difficult to identify among a set of data.
  - Fortunately, there is an effective way to identify transitive dependencies: they occur only when a functional dependence exists among nonprime attributes.

13

Partial dependency is a functional dependence in which the determinant is <u>only part of the primary key.</u>

For example, (<u>A, B</u> → C, D) is Functional dependence whereas **B → C** is Partial dependency because **only part of the primary key (B)** is needed to determine **the value of C**.

Transitive dependency is **when an attribute, e.g. Z**, **functionally depends on another non-key attribute, e.g. Y., i.e. nonkey to nonkey)**

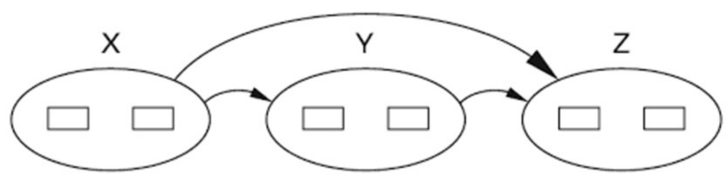
For instance, <u>X</u> → Y, Y → Z, then X → Z is transitive via a nonprime Y, so Y → Z (i.e. nonkey to nonkey) is a transitive dependency (to be removed to get to 3NF)

Transitivity andTransitive Dependency

if X → **Y** and **Y** → **Z** then **X** → **Z**

Example: if Suburb→Postcode and Postcode→State then Suburb→State

2052 → Kensington and Kensington → NSW,
then 2052→NSW



2052 → Kensington and Kensington → NSW then 2052→NSW

Normalisation

15

Normalisation Process

- Objective is to ensure that each table conforms to the concept of well-formed relations
  - Each table represents a single subject
  - No data item will be **unnecessarily** stored in more than one table
  - All **nonprime attributes** in a table are dependent on the primary key
  - Each table is void of insertion, update, and deletion **anomalies**
- Ensures that all tables are in at least 3NF (rule of thumb)
- Works one relation at a time
- Starts by:
  - Identifying the dependencies of a relation (table)
  - Progressively breaking the relation into new set of relations/tables

16

As stated in the slide…

In relational databases, tables are usually in 3NF or Boyce-Codd NF, and some are in 2NF (for performance).

In banks, it could be 4NF or 5NF, because of time of access to the database, such as from ATM. You might have a few person working on it to tune it up to a higher NF.

Lossless Decomposition and Normal Forms

- Our aim is to **decompose** relations/tables so to **reduce size/redundancy**.
- We use **inferences rules** for this decomposition **process**.
- We need to be sure that the decomposed components (tables/relations) have the **lossless** join property (i.e., decomposed components could be joined back together to the original table/relation).

TABLE 6.2

NORMAL FORMS

NORMAL FORM	CHARACTERISTIC
First normal form (1NF)	Table format, no repeating groups, and 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)
Fourth normal form (4NF)	3NF and no independent multivalued dependencies

17

As stated in the slide…

Construction Company Example

Scenario: database for reports for a **construction company.**

- Building project has: Project number, Name, Employees assigned to the project.
- Employee has: 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.

The following slide shows a table with contents correspond to the reporting requirements but is not "normalised."

TABLE 6.1

A SAMPLE REPORT LAYOUT

PROJECT NUMBER	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	CHARGE/ 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	Rolling Tide	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	Starflight	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
		[illegible]	Jam[illegible]	[illegible]	$ 18.36	30.[illegible]	$ [illegible]
		1[illegible]2	Darlene M. Smit[illegible]on	[illegible]	$ 45.95	41.[illegible]	$ [illegible]
				[illegible]ubt[illegible]			**$17,[illegible]9.8[illegible]**
				[illegible]tal			**$48,941.09**

Note: * indicates the project leader.

18

As stated in the slide…

1NF

19

Conversion to First Normal Form (1NF)

- **Aim**: creating a valid relation.

- A relation / table is in **1NF** if:
  - The key attributes are defined, i.e. PK attributes are defined and not NULL (i.e., **a valid PK**).
  - All attributes are dependent on the primary key
  - There are no repeating groups in the table
  - All attributes contain only atomic values (i.e., **no multivalued attributes**).

- Action to create/check 1NF:
  - Step 1: Cleaning & dealing with Repeating Groups and Multi-valued Attributes
  - Step 2: Identify the Primary Key
  - Step 3: Identify All Dependencies

20

All relational tables satisfy 1NF requirements as stated on the slide

It is OK if some tables contain partial dependencies in 1NF - subject to data redundancies and various anomalies

The actions to create/check 1NF are shown and discussed next.

Steps to Follow for 1NF

- **Draw Dependency Diagram**
- **Step 1: Cleaning & dealing with Repeating Groups and Multi-valued Attributes**
  - **Split multivalued attributes** and **split repeating groups of data** (i.e., transform multivalued attributes in additional columns, or, better, additional rows).
  - Add the **appropriate entry** in at least for the **primary keys column(s).**
- **Step 2: Identify the Primary Key**
  - All attributes are dependent on PROJ_NUM + EMP_NUM
- **Step 3: Identify All Dependencies**
  - Draw Dependency Diagram
  - Partial dependency: attributes are dependent on only a part of a composite PK
  - Transitive dependency: non-key (nonprime) attributes are dependent on another non-key attribute

21

Tips: Before going through steps of Normalisation, you want to remove all the derived attributes, i.e. calculated values.

TABLE 6.1

A SAMPLE REPORT LAYOUT

PROJECT NUMBER	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	CHARGE/ 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			[illegible]
18	Amber Wave	114	Annelise Jones	Applications Designer	$ 48.10	24.6	$ [illegible],183.2[illegible]
		118	James J. Frommer	General Support	$ 18.36	45.3	$ 831.[illegible]1
		104	Anne K. Ramoras *	Systems Analyst	$ 96.75	32.4	$ [illegible]
		112	Darlene M. Smithson	DSS Analyst	$ 45.95	44.0	$ [illegible]
				Subtotal			[illegible]
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7	$ [illegible]
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4	$ [illegible]
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6	$ [illegible]
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0	$ [illegible]
		106	William Smithfield	Programmer	$35.75	12.8	$ [illegible]
				Subtotal			[illegible]
25	Starflight	107	Maria D. Alonzo	Programmer	$ 35.75	24.6	$ [illegible]
		115	Travis B. Bawangi	Systems Analyst	$ 96.75	45.8	$ [illegible]
		101	John G. News *	Database Designer	$105.00	56.3	$ [illegible]
		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			[illegible]
				Total			[illegible]

Note: * indicates the project leader.

Remove derived attributes

TABLE 6.1

A SAMPLE REPORT LAYOUT

PROJECT NUMBER	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	CHARGE/ 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	Rolling Tide	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	Starflight	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 the project leader.

So, which are the PK and dependencies?

23

So what are the PKs and their dependencies?

The dependencies are built on data shown above. You have examine the data carefully and see if there are any association between them.

If you are good at spot the difference" between two pictures or numbers or Sudoku, then you should have no problem of picking up differences and similarities. Otherwise, not to worry, you still can do it but probably it will take you a bit longer.

Have a look at project number and project name – they are unique!

Have another look at:

(a) employee number and employee name;
(b) employee_number, job_class and charge/hour.
(c) etc.

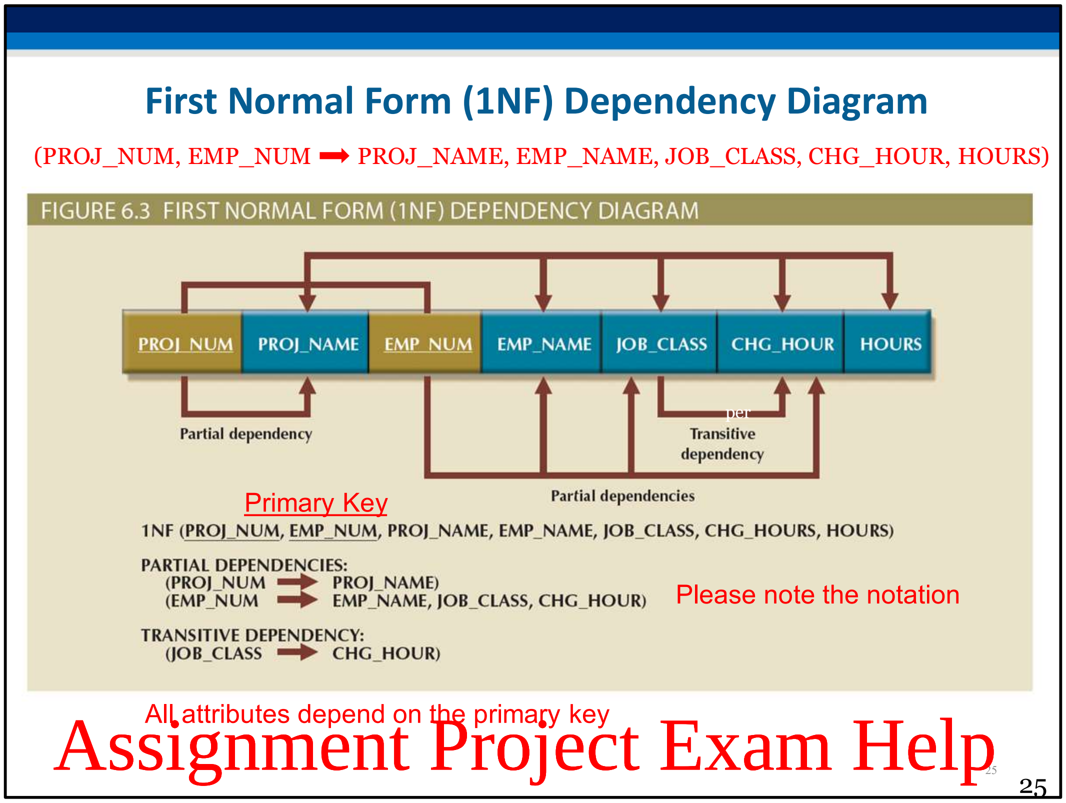
Examine the Similarities and Differences of the Data

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

ALL_IN_ONE (PROJ_NUM, PROJ_NAME, EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR, HOURS)

24

If the data come in a spreadsheet, then you can examine the similarities and differences of the Data to build up your PK and dependencies using the **functions** and **features** under the **Data** tab in **Excel**.



So, you draw like this, 1NF Dependency Diagram!

The **primary composite key (PK)** is **Project Number (proj_num)** and **Employee Number (emp_num)**

As stated in the slide…

Remember, Partial dependency is a functional dependence in which the determinant is only part of the primary key, in this case, Project Number is only part of PK Project Number and Employee Number

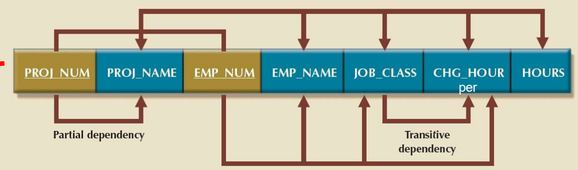
Same as Employee Number **(emp_num), emp_name, job_class, and chg_hour**

Remember Transitive dependency is an attribute, e.g. z, functionally depends on another non-key attribute, e.g. Y., i.e. nonkey to nonkey, so **Job Class** and **Change Hour**!

Explanation: First Normal Form (1NF) Dependency Diagram

- **Draw Dependency Diagram**
- **Identify the Primary Key**
  - All attributes are dependent on PROJ_NUM + EMP_NUM
- **Identify All Partial Dependencies**
  - Partial dependency: attributes are dependent on a part of a PK or composite PK
- **Identify All Transitive Dependencies**
  - Transitive dependency: attributes depend on other attributes but not the PK attributes

FIGURE 6.3 FIRST NORMAL FORM (1NF) DEPENDENCY DIAGRAM



Primary Key

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

PARTIAL DEPENDENCIES:
(PROJ_NUM → PROJ_NAME)
(EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR)

Please note the notation

TRANSITIVE DEPENDENCY:
(JOB_CLASS → CHG_HOUR)

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 DATA ORGANIZATION: FIRST NORMAL FORM

2NF

Conversion to Second Normal Form (2NF)

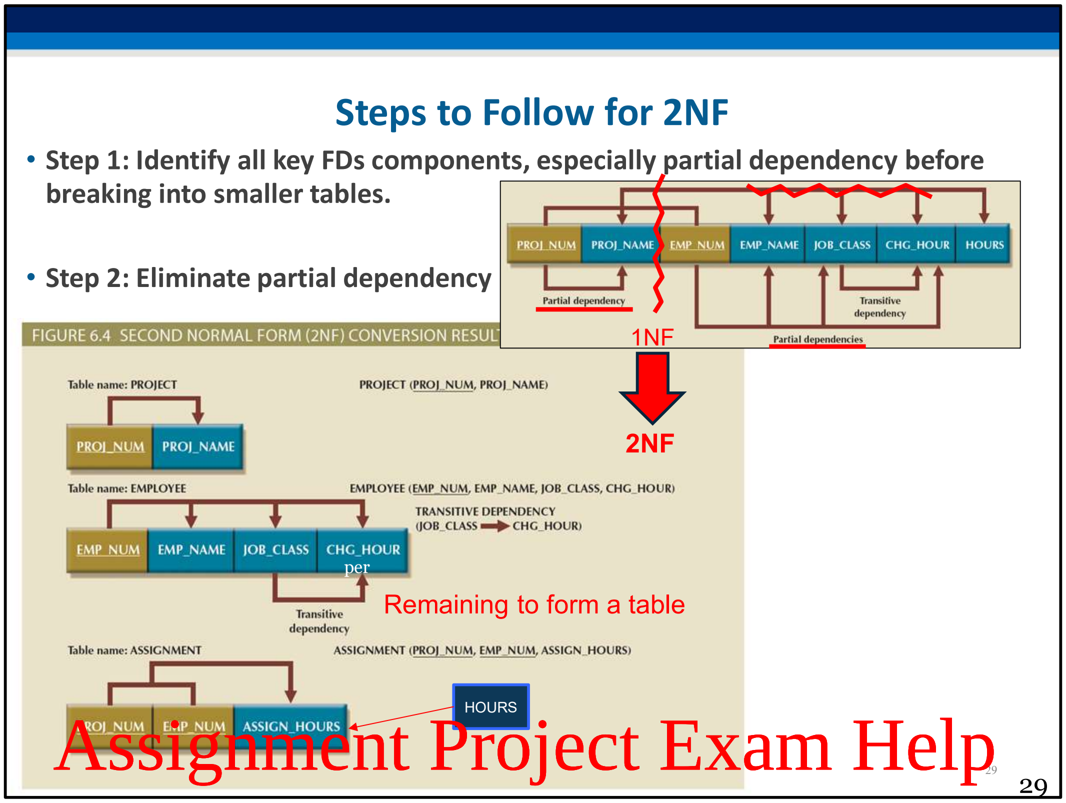
- **Aim**: remove **partial dependencies** (no repeating values in non-key fields).
- A relation / table is in **2NF** if:
  - No partial dependencies (Each non-key field is functionally dependent on the entire PK).
  - The relation/table must be in 1NF.
- Action to create/check **2NF**:
  - Step 1: **Analyse FDs, especially partial dependencies, and assign corresponding dependent attributes**
  - Step 2: Make new tables by eliminating **partial dependencies** (attributes not functionally dependent on the entire primary key) by separating the data items into a separate relation using appropriate PKs (may need bridge/junction table).

28

All relational tables satisfy 2NF requirements as stated on the slide.

The key process in 2NF is get rid of partial dependencies and assign other dependent attributes.

Hint: Look for values that occur multiple times in non-key fields. This tells you that you have too many fields in a single table. In a well-designed database, the only data that is duplicated is in key fields used to connect tables.



So, we want to get rid of partial dependencies!

Let's go back to the 1NF, so what can we do?

(1) The partial dependency of proj_num and proj_name can become a table, with proj_num the PK.
(2) The partial dependency of emp_num, emp_name, job_class and chg_hour can become another table, with emp_num is the PK.
(3) As for the remaining, we left with proj_num, emp_num and hours, eliminating all columns allocated to (2), and this can form the last table. The composite PK is proj_num and emp_num.

So in 2NF,
(a) We have a table Product table formed from the partial dependency in (1).
(b) We have Employee table formed from the partial dependency in (2).
(c) We have the remaining table to form Assignment table, and we change the column name from hours to assign_hours to make it more meaningful.

3NF

30

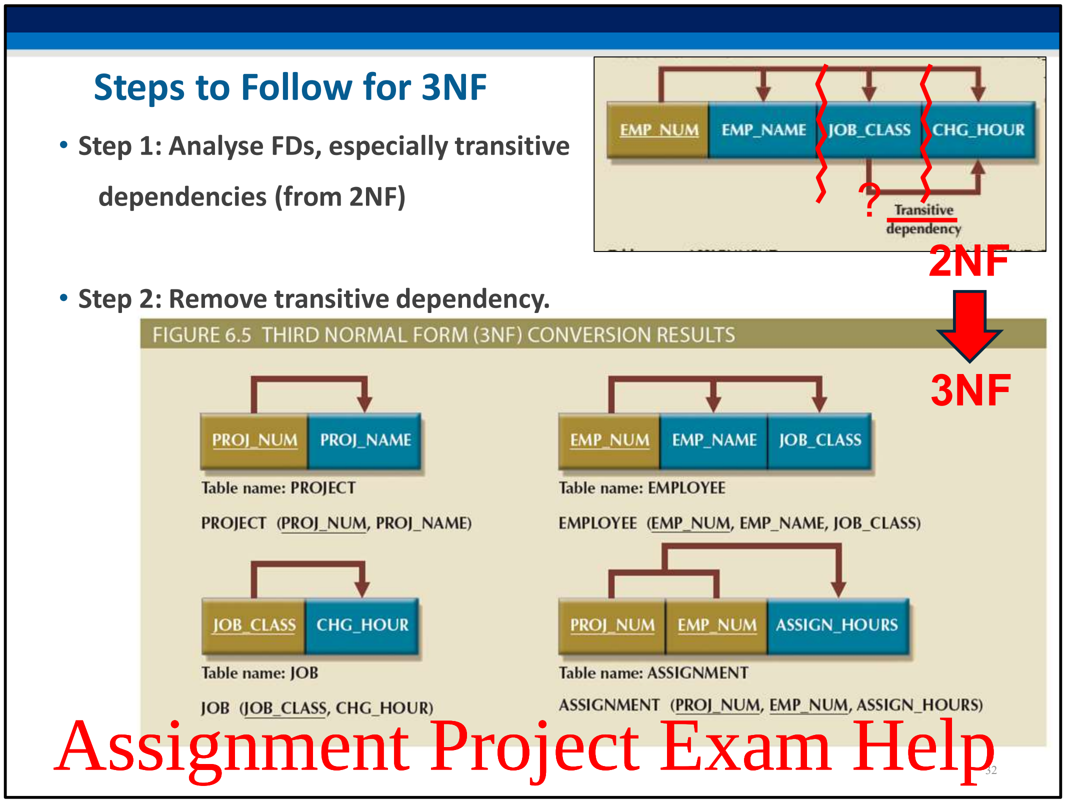
Conversion to Third Normal Form (3NF)

- **Aim**: **remove non-key dependencies**, data that is not dependent on other keys.
-
- A relation / table is in 3**NF** if:
  - It has **no transitive dependencies** (no non-key attributes determined by other non-candidate-key attributes).
  - The relation/table must be in 2NF.

- Action to create/check **3NF**:
  - Step 1: **Analyse FDs, especially transitive dependencies, and reassign corresponding dependent attributes**
  - Step 2: Make new tables to eliminate all **transitive dependencies**
    - **Determinant**: Any attribute whose value determines other values within a row

All relational tables satisfy 3NF requirements as stated on the slide

The key process in 3NF is get rid all the transitive dependencies.



Let's go back to the 2NF, so what else we need to do?

(1) Project and Assignment tables are OK, and nothing we need to do.
(2) We remove transitive dependency from the employee table into a new table called job. The transitive dependency of job_class and chg_hour are added to Job table, with job_class is the PK.
(3) The remaining columns are emp_num and emp_name, but we need job_class column because it is associated with the employee to say what job class the employee belongs to. Thus, job_class column is an FK in the Employee table.

Improving the Design

33

Improving the Design

- Evaluate PK assignments and naming conventions
- Refine attribute atomicity
  - **Atomic attribute**: Cannot be further subdivided
  - **Atomicity**: Characteristic of an atomic attribute
- **Identify new attributes and new relationships**
- **Refine primary keys as required for data granularity**
  - **Granularity**: Level of detail represented by the values stored in a table's row
- Maintain historical accuracy and evaluate using derived attributes

34

As stated in the slide…

Normalisation is valuable because its use helps eliminate data redundancies, next we want to see if we can improve the design.

More information can be found in Textbook 6-4.

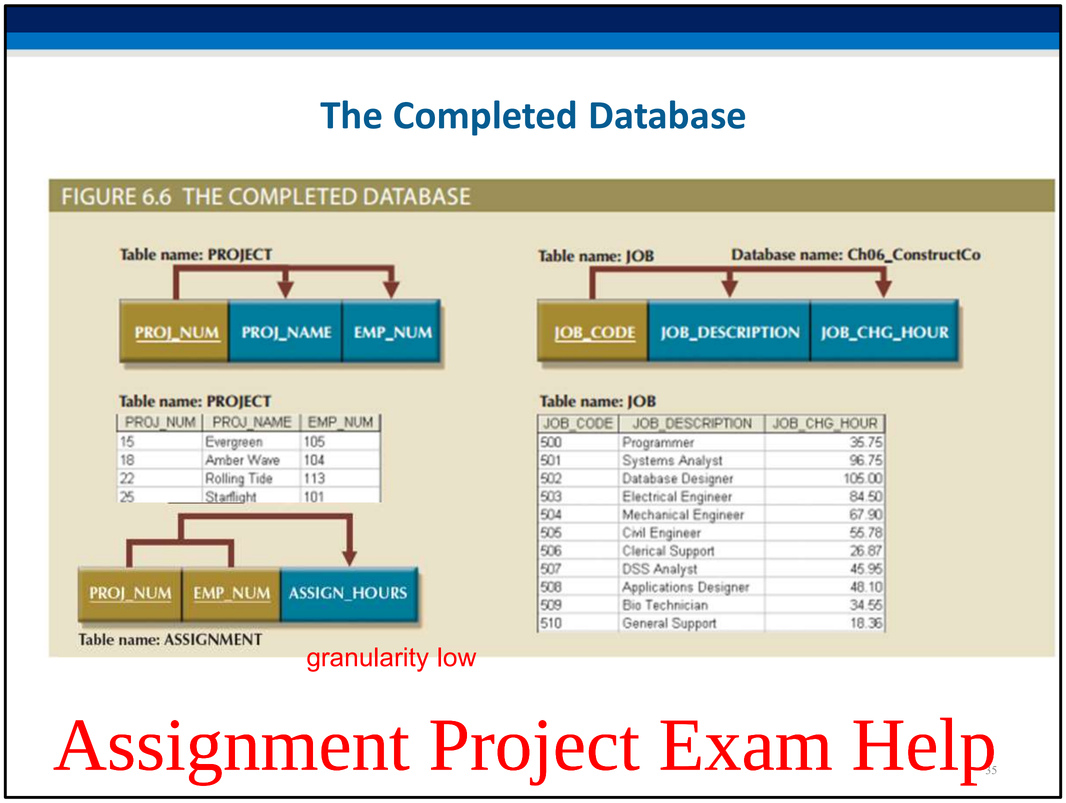


Table name: PROJECT

PROJ_NUM	PROJ_NAME	EMP_NUM
15	Evergreen	105
18	Amber Wave	104
22	Rolling Tide	113
25	Starflight	101

Table name: JOB

JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
500	Programmer	35.75
501	Systems Analyst	96.75
502	Database Designer	105.00
503	Electrical Engineer	84.50
504	Mechanical Engineer	67.90
505	Civil Engineer	55.78
506	Clerical Support	26.87
507	DSS Analyst	45.95
508	Applications Designer	48.10
509	Bio Technician	34.55
510	General Support	18.36

As stated in the slide…

For the Project table, you add emp_num for the roject leader.
For the Job table, you add job description to make it more meaningful to the user.
Etc…

Surrogate Key Considerations

- Used by designers when the primary key is considered to be unsuitable
- System-defined attribute
- Created a managed via the DBMS
- Have a numeric value which is automatically incremented for each new row

TABLE 5.4

DATA USED TO KEEP TRACK OF EVENTS

DATE	TIME_START	TIME_END	ROOM	EVENT_NAME	PARTY_OF
6/17/2016	11:00a.m.	2:00p.m.	Allure	Burton Wedding	60
6/17/2016	11:00a.m.	2:00p.m.	Bonanza	Adams Office	12
6/17/2016	3:00p.m.	5:30p.m.	Allure	Smith Family	15
6/17/2016	3:30p.m.	5:30p.m.	Bonanza	Adams Office	12
6/18/2016	1:00p.m.	3:00p.m.	Bonanza	Boy Scouts	33
6/18/2016	11:00a.m.	2:00p.m.	Allure	March of Dimes	25
6/18/2016	11:00a.m.	12:30p.m.	Bonanza	Smith Family	12

These are required fields to uniquely identify an event. It is too cumbersome to use them all as a foreign key. Programmer may like to assign an "event_id" for each event, not known to the user, but only to the programmer for convenience, which is a surrogate PK.

36

As stated in the slide…

What is surrogate key? It is when the number of columns to define a composite primary key. Sometimes it made difficult to reference from another table if we want to reference as an FK. Instead a unique key is created to substitute the composite primary key.

In this example, we can create an **event_id as** a **surrogate key** can consist of a composite primary **key date**, **time_start** and **room**, but **event_id** cannot guarantee the uniqueness of the row, so if **event_id** is a PK, we need to test for uniqueness of combination of columns **date**, **time_start** and **room**, which we will call it an **unique key**.

Note: there are technically two keys to ensure a row is unique – the **surrogate key** and the **unique key**. Now, you can reference **event_id** as an **FK** from another table.

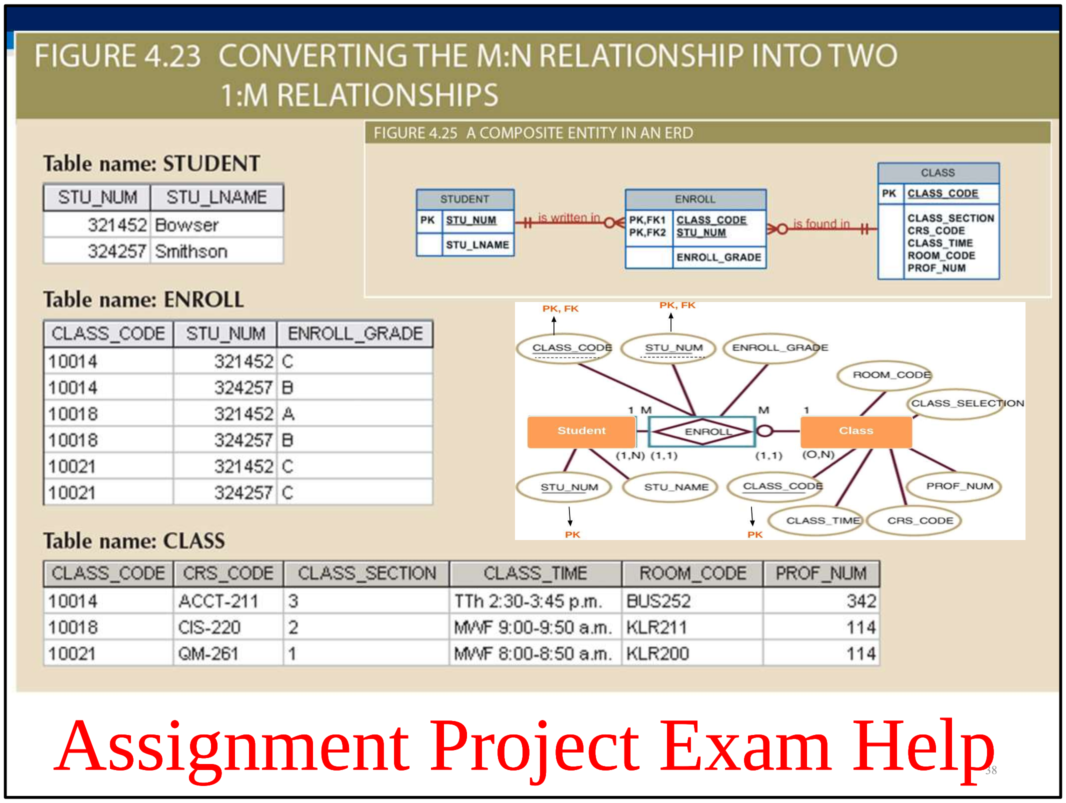
TABLE 6.4		
DUPLICATE ENTRIES IN THE JOB TABLE		
JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
511	Programmer	$35.75
512	Programmer	$35.75

- Nothing to prevent job code entered twice,
- Job_code is technically a surrogate key
- To ensure it is unique, we need to test job_description to ensure the description is unique, or it can be defined as **unique key**, i.e. you can only have one Programmer in the Job Description.

37

As stated in the slide…

This is a simple example that when we define **job_code** as a **surrogate key**, and if we do not test if *job_description* is unique or not, then a problem such as above will happen. To address this problem, we test ***job_description*** to ensure the description is **unique**. That is, we define ***job_description*** column as **unique key.**



STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

As stated in the slide

Remember, this slide! If you were to design using ERD or use normalisation. in this example, **class_code** is a **surrogate key**. The class_code might be made up of, say, crs_code, class_section and class_time.

BCNF

Boyce-Codd Normal Form (BCNF)

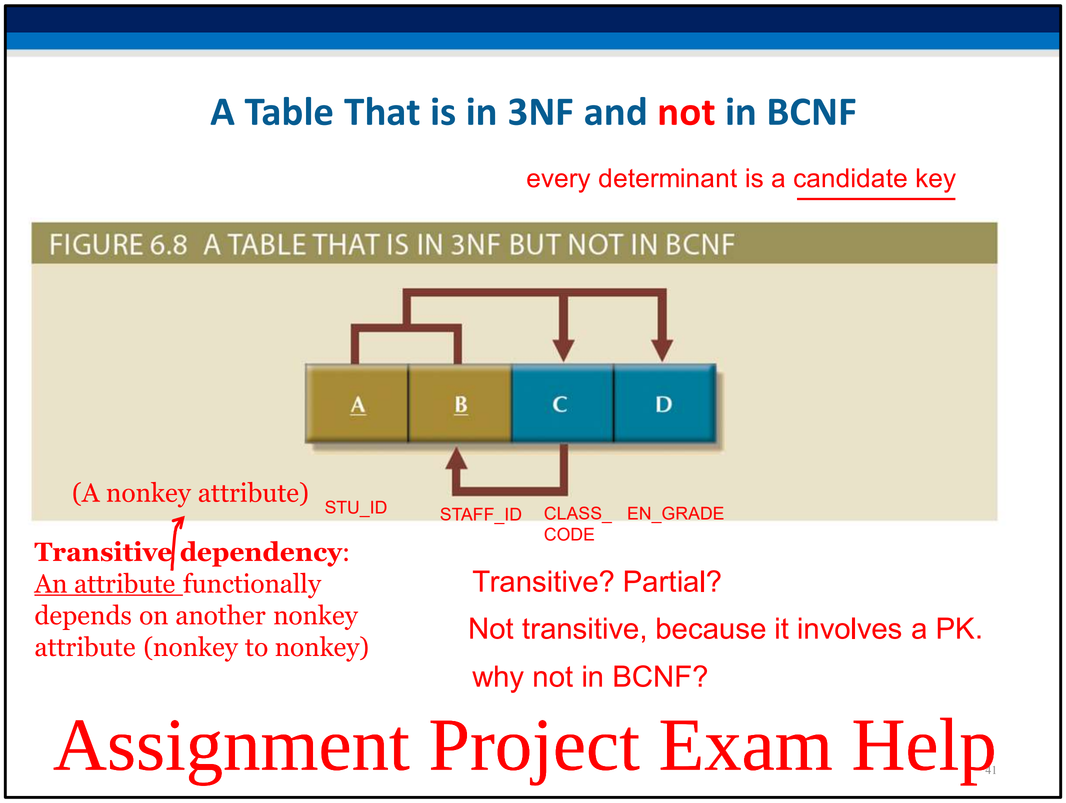
- Aim: higher normal forms such as BCNF do cover some specific aspects and problems with the 3NF
  - Based on paper **Boyce & Codd (1974).**
  - Sometimes called **3.5NF.**
  - 3NF is always achievable, **BCNF is not always achievable (Beeri & Bernstein 1979).**

- Candidate Key: Every determinant in the table should be a candidate key
  - Same characteristics as primary key but not chosen to be the primary key
  - Equivalent to 3NF when the table contains only one candidate key
  - Violated only when the table contains more than one candidate key
  - Considered to be a special case of 3NF

- A relation/table is in **BCNF** if, for every one of its dependencies X → Y, one of the following conditions holds true:
  - X → Y is a TRIVIAL FUNCTIONAL DEPENDENCY (i.e., Y is a subset of X)
  - X is a SUPERKEY

40

As state on slide…

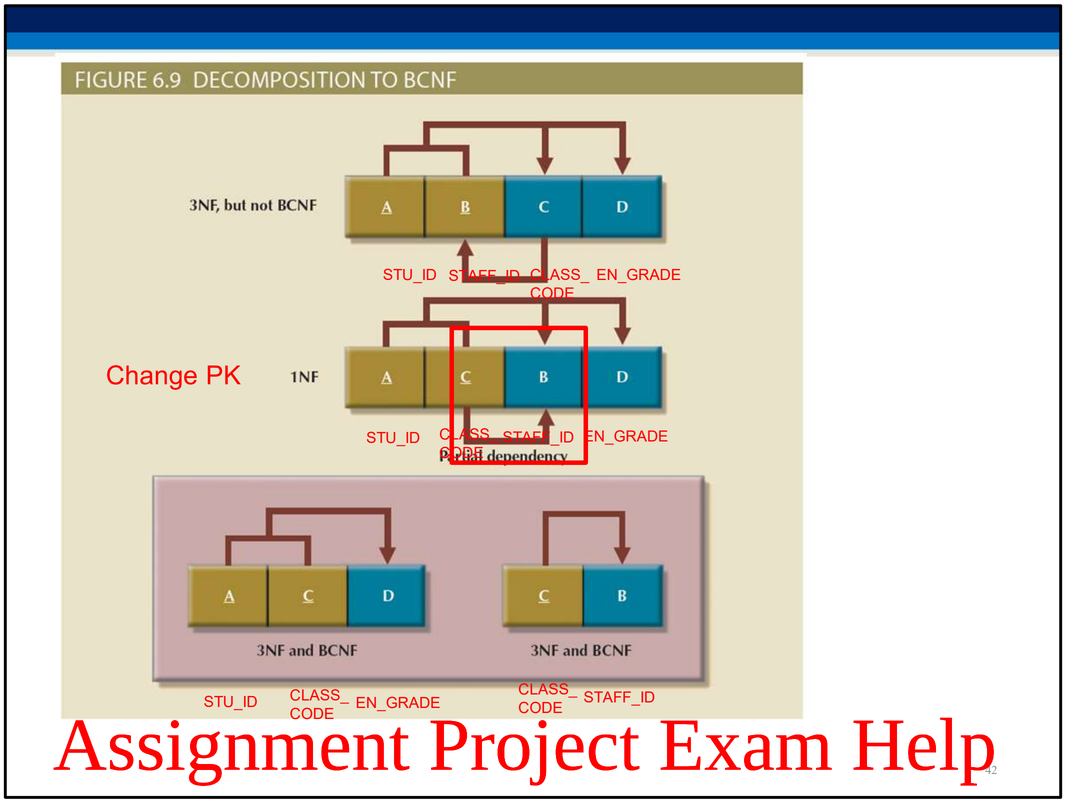


Class Code and Staff Id – A class needs a staff to teach that class.

Is it a Transitive key? Not quite because Staff Id or B is part of a PK. Remember, Transitive dependency is an attribute functionally depends on another nonkey attribute!

It is **not Partial** because **partial dependency** is when functional dependence in which the determinant is only part of the primary key. **C** is not part of a **PK** whereas **B** is part of a **PK**.

Thus, this leads to BCNF!



What next is interesting!

You swap the attributes in the composite PK, that is Staff Id or B, and Class Code or C are changed and swapped! Now, it becomes Partial Dependency

You then create a new table for Partial Dependency, and the remaining as the remaining table.

Sometimes, it might not reach here if ERD is done first and attributes/columns were assigned before normalisation is performed to ensure data consistency.

4NF

43

Fourth Normal Form (4NF)

- Table is in 4NF when it:
  - Is in 3NF
  - Has no multivalued dependencies
- Rules
  - All attributes must be dependent on the primary key, but they must be independent of each other
  - No row may contain two or more multivalued facts about an entity

- **Multivalued attributes**: Attributes that have many values e.g., car color; diploma – BS, MS, MBA, etc.

44

4NF is as far as we will go in this course!

The key problem with some of the tables are there are multi-values dependencies – there are lots of combinations.

As stated in the slide…

4th Normal Form Example (1)

Pizza Chain Table		
Pizza Chain	**Pizza**	**Store**
Domino	Hot & Spicy	Kingsford
Domino	Super Supreme	Kingsford
Domino	Hawaiian Pizza	Kingsford
Domino	Hawaiian Pizza	Randwick
Pizza Hut	Hot & Spicy	Randwick
Pizza Hut	Hot & Spicy	Kensington
Pizza Hut	Hot & Spicy	Kingsford
Pizza Hut	Hawaiian Pizza	Randwick
Pizza Hut	Hawaiian Pizza	Kensington
Pizza Hut	Hawaiian Pizza	Kingsford
Vincenzo's Pizza	Emperor Supreme	Randwick
Vincenzo's Pizza	Emperor Supreme	Kensington
Vincenzo's Pizza	Hot & Spicy	Randwick
Vincenzo's Pizza	Hot & Spicy	Kensington

Pizza Chain Table		
Pizza Chain	**Pizza**	**Store**
Domino	Hot & Spicy	Kingsford
Domino	Super Supreme	Kingsford
Domino	Hawaiian Pizza	Kingsford
Domino	Hawaiian Pizza	Randwick
Pizza Hut	Hot & Spicy	Randwick
Pizza Hut	Hot & Spicy	Kensington
Pizza Hut	Hot & Spicy	Kingsford
Pizza Hut	Hawaiian Pizza	Randwick
Pizza Hut	Hawaiian Pizza	Kensington
Pizza Hut	Hawaiian Pizza	Kingsford
Vincenzo's Pizza	Emperor Supreme	Randwick
Vincenzo's Pizza	Emperor Supreme	Kensington
Vincenzo's Pizza	Hot & Spicy	Randwick
Vincenzo's Pizza	Hot & Spicy	Kensington
…	…	…
Pizza Hut	Super Supreme	Randwick
Pizza Hut	Super Supreme	Kensington
Pizza Hut	Super Supreme	Kingsford

45

When you have a table with multivalues, you have a combination of values such as the Pizza Chain.

The table is in 3NF and BCNF. However, if I were to add a new pizza called Super Supreme to Pizza Hut chain, I have to add to all three stores, namely Kingsford, Randwick and Kensington, i.e. I have to get the combination correct!

The issue here is Pizza Chain depends on Pizza and Store, but we can break it down to two tables (Pizza and Store) instead of one table (Pizza Chain)!!!

4th Normal Form Example

Pizza Chain Table		
Pizza Chain	**Pizza**	**Store**
Domino	Hot & Spicy	Kingsford
Domino	Super Supreme	Kingsford
Domino	Hawaiian Pizza	Kingsford
Domino	Hawaiian Pizza	Randwick
Pizza Hut	Hot & Spicy	Randwick
Pizza Hut	Hot & Spicy	Kensington
Pizza Hut	Hot & Spicy	Kingsford
Pizza Hut	Hawaiian Pizza	Randwick
Pizza Hut	Hawaiian Pizza	Kensington
Pizza Hut	Hawaiian Pizza	Kingsford
Vincenzo's Pizza	Emperor Supreme	Randwick
Vincenzo's Pizza	Emperor Supreme	Kensington
Vincenzo's Pizza	Hot & Spicy	Randwick
Vincenzo's Pizza	Hot & Spicy	Kensington

Store	
Pizza Chain	**Store**
Domino	Kingsford
Domino	Randwick
Pizza Hut	Kensington
Pizza Hut	Kingsford
Pizza Hut	Randwick
Vincenzo's Pizza	Randwick
Vincenzo's Pizza	Kensington

Pizza Chain - Pizza	
Pizza Chain	**Pizza**
Domino	Hot & Spicy
Domino	Super Supreme
Domino	Hawaiian Pizza
Pizza Hut	Hot & Spicy
Pizza Hut	Hawaiian Pizza
Vincenzo's Pizza	Emperor Supreme
Vincenzo's Pizza	Hot & Spicy

One table into Two tables
Pizza Chain → Pizza Chain Pizza
Pizza Chain → Store

46

The issue here is Pizza Chain depends on Pizza and Store, but we can break it down to two tables (Pizza and Store) instead of one table (Pizza Chain)!!!

Based on Pizza Chain table, I created two tables: **Pizza Chain Pizza**, and **Store**!

Pizza Chain →**Pizza Chain Pizza** and
Pizza Chain →**Store**

Store table has Pizza Chain and Store.
Pizza Chain Pizza table has Pizza Chain and Pizza.

4th Normal Form Example

Pizza Chain Table		
Pizza Chain	**Pizza**	**Store**
Domino	Hot & Spicy	Kingsford
Domino	Super Supreme	Kingsford
Domino	Hawaiian Pizza	Kingsford
Domino	Hawaiian Pizza	Randwick
Pizza Hut	Hot & Spicy	Randwick
Pizza Hut	Hot & Spicy	Kensington
Pizza Hut	Hot & Spicy	Kingsford
Pizza Hut	Hawaiian Pizza	Randwick
Pizza Hut	Hawaiian Pizza	Kensington
Pizza Hut	Hawaiian Pizza	Kingsford
Vincenzo's Pizza	Emperor Supreme	Randwick
Vincenzo's Pizza	Emperor Supreme	Kensington
Vincenzo's Pizza	Hot & Spicy	Randwick
Vincenzo's Pizza	Hot & Spicy	Kensington

Store	
Pizza Chain	**Store**
Domino	Kingsford
Domino	Randwick
Pizza Hut	Kensington
Pizza Hut	Kingsford
Pizza Hut	Randwick
Vincenzo's Pizza	Randwick
Vincenzo's Pizza	Kensington

Pizza Chain - Pizza	
Pizza Chain	**Pizza**
Domino	Hot & Spicy
Domino	Super Supreme
Domino	Hawaiian Pizza
Pizza Hut	Hot & Spicy
Pizza Hut	Hawaiian Pizza
Vincenzo's Pizza	Emperor Supreme
Vincenzo's Pizza	Hot & Spicy
…	…
Pizza Hut	Super Supreme

Now, I only have to add Super Supreme to Pizza Hut in Pizza Chain Pizza table. Thus, I just have to add one row to one table instead of trying to get the combinations correct like previously!

Normalisation and Database Design

- Normalisation should be part of the design process
- Proposed entities must meet required the normal form before table structures are created
- Principles and normalisation procedures to be understood to redesign and modify databases
  - ERD is created through an iterative process
  - Normalisation focuses on the characteristics of specific entities

As stated in the slide…

In summary, you will need to do ERD and normalisation for better database design. However, today, a, database designer might not able to do both because of time constraint.

Denormalisation

- Design goals
  - Creation of normalized relations
  - Processing requirements and speed
- Number of database tables expands when tables are decomposed to conform to normalisation requirements
  - Joining a larger number of tables:
    - Takes additional input/output (I/O) operations and processing logic
    - Reduces system speed
- Defects in unnormalized tables
  - **Data updates are less efficient because tables are larger**
  - Indexing is more cumbersome
  - No simple strategies for creating virtual tables known as views

Data is redundant but access will be much faster – this is in data warehousing and big data!

Common Denormalisation Examples

TABLE 6.6		
COMMON DENORMALIZATION EXAMPLES		
CASE	**EXAMPLE**	**RATIONALE AND CONTROLS**
Redundant data (ZIP, CITY)	Storing ZIP and CITY attributes in the AGENT table when ZIP determines CITY (see Figure 2.2)	Avoid extra join operations Program can validate city (drop-down box) based on the zip code
Derived data (Course, credit hr)	Storing STU_HRS and STU_CLASS (student classification) when STU_HRS determines STU_CLASS (see Figure 3.28)	Avoid extra join operations Program can validate classification (lookup) based on the student hours
Preaggregated data (also derived data) storing avg grade point	Storing the student grade point average (STU_GPA) aggregate value in the STUDENT table when this can be calculated from the ENROLL and COURSE tables (see Figure 3.28)	Avoid extra join operations Program computes the GPA every time a grade is entered or updated STU_GPA can be updated only via administrative routine
Information requirements	Using a temporary denormalized table to hold report data; this is required when creating a tabular report in which the columns represent data that are stored in the table as rows (see Figures 6.17 and 6.18)	Impossible to generate the data required by the report using plain SQL No need to maintain table Temporary table is deleted once report is done Processing speed is not an issue

50

As stated in the slide…

Class Exercise

Class Exercise

Memb ID	Memb Name	Call No	Copy ID	Book Title	Book Author	Author ID	Date Borrow	Date Return
10	A. Hope	SQ231.215	4	Jack Sprat's Hat – was it for real?	JK Spratt	A1	1/2/04	17/3/04
42	B. Marcy	S14.143	1C	Knowing what you know and knowing what to do with it.	K Nowles	A4	2/2/99	12/2/99
		AV127.143	5	Life and Times of the Iguana	IG Uana	A7	2/2/99	12/2/99
		S14.143	3C	Knowing what you know and knowing what to do with it.	K Nowles	A4	1/5/03	
24	C. Sam	PJ234.234	4	The Tech Heads guide to Technology	IM Smart	A9	3/4/99	
		S14.143	1C	Knowing what you know and knowing what to do with it.	K Nowles	A4	3/4/99	
56	E. Bronwyn	SQ231.215	2	Jack Sprat's Hat – was it for real?	JK Spratt	A1	3/3/99	5/3/04
67	F. Mac	AV127.143	5	Life and Times of the Iguana	IG Uana	A7	4/4/99	

1. Based on the above report, create tables to the 3NF or BCNF using Normalisation.
2. Based on the above report, create an ERD (if you have time)