SIT103/SIT772 Database Fundamentals

Week 4

Normalisation

Dr Iynkaran Natgunanathan,

email: iynkaran.natgunanathan@deakin.edu.au, Phone: +61 3 924 68825.

Last Week

- Types of Attributes
- Strengths of Relationships
- Types of Entities
- Implementing Relationships
- Extended/Advanced ERD concepts
- Modelling historical time-variant data
- Case studies

Last Week’s OnTrack Tasks

- Task 3.1P Modelling database in terms of ERD

  - Using Lucid Chart or MS Visio

  - Due this Friday

- Task 3.2HD Research Report and Presentation

  - On a topic of your interest related to database and/or data management

Questions?

Any questions/comments so far

Last week's content

OnTrack tasks

Anything in general about the unit

This week

- Database Anomalies

  - Insertion/Update/Deletion Anomalies

- Functional Dependencies
- Normalisation

  - 1NF, 2NF, and 3NF

- Denormalisation

Anomalies in databases

- **What is an Anomaly in a database?**

  - Problems/issues that can occur because of poor database design

  * Redundant data

  * Inconsistence/inaccurate data

  * Data loss

  * Limitations to add/delete data

  - Type of anomalies

  1. Insertion Anomalies

  2. Update Anomalies

  3. Delete Anomalies

Insertion Anomaly

- In ability to add data in a database due to the absence of other data

  - because of unnecessary coupling of data

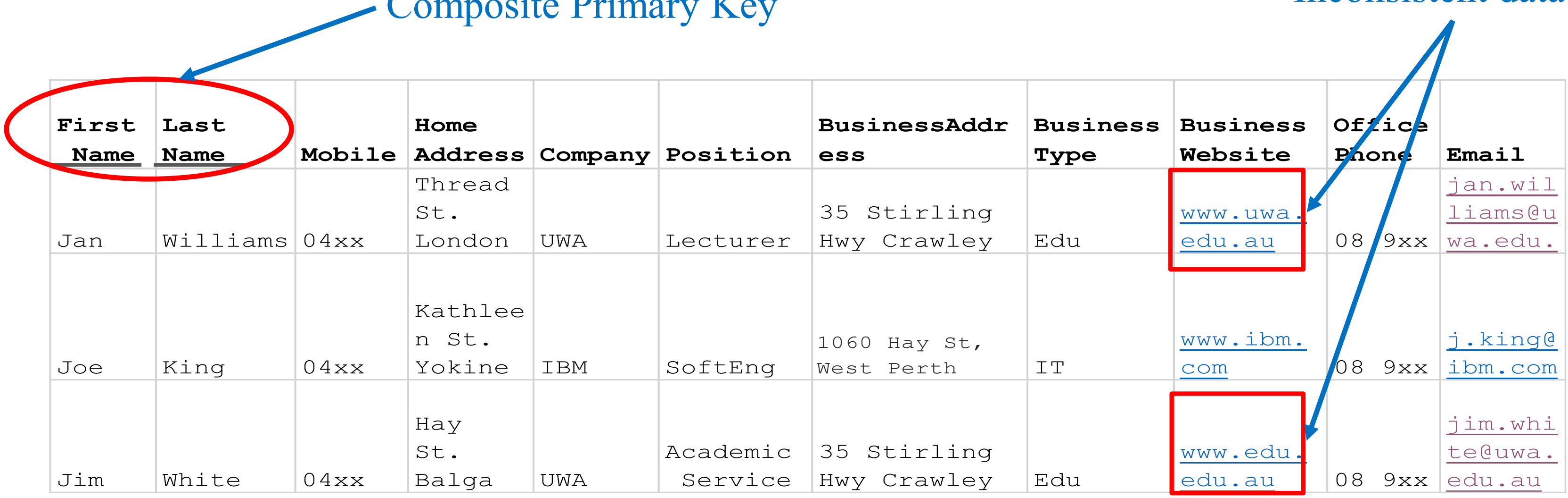
- Same information has to be inserted again and again

  - results in data inconsistency if mistakes are made

Example:

Composite Primary Key

Inconsistent data



First Name	Last Name	Mobile	Home Address	Company	Position	BusinessAddress	Business Type	Business Website	Office Phone	Email
Jan	Williams	04xx	Thread St. London	UWA	Lecturer	35 Stirling Hwy Crawley	Edu	www.uwa.edu.au	08 9xx	jan.williams@uwa.edu.
Joe	King	04xx	Kathleen St. Yokine	IBM	SoftEng	1060 Hay St, West Perth	IT	www.ibm.com	08 9xx	j.king@ibm.com
Jim	White	04xx	Hay St. Balga	UWA	Academic Service	35 Stirling Hwy Crawley	Edu	www.edu.edu.au	08 9xx	jim.white@uwa.edu.au

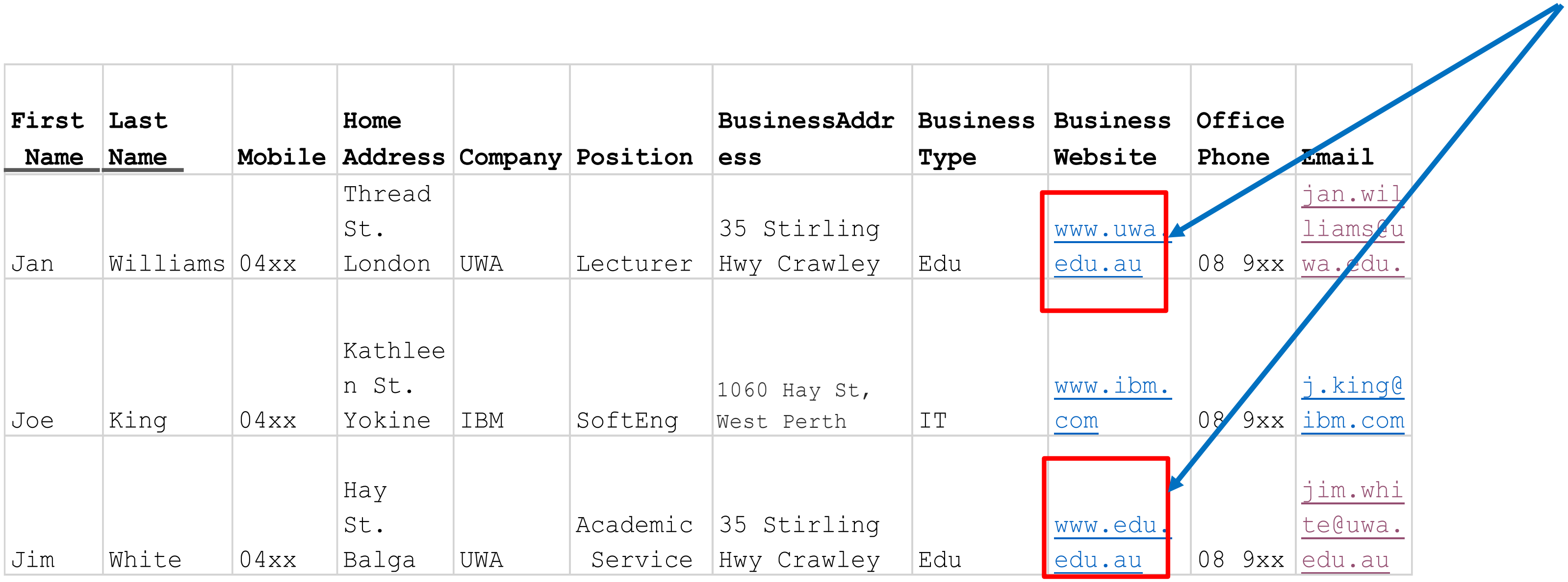
A new company cannot be added until an Employee is available to add

Update Anomaly

- Changing one incorrect data could involve updating many records, leading to the possibility of some changes being made incorrectly
  - can result in data inconsistencies
  - because of data redundancy

Example: updating Business website for Jan Williams

Inconsistent data



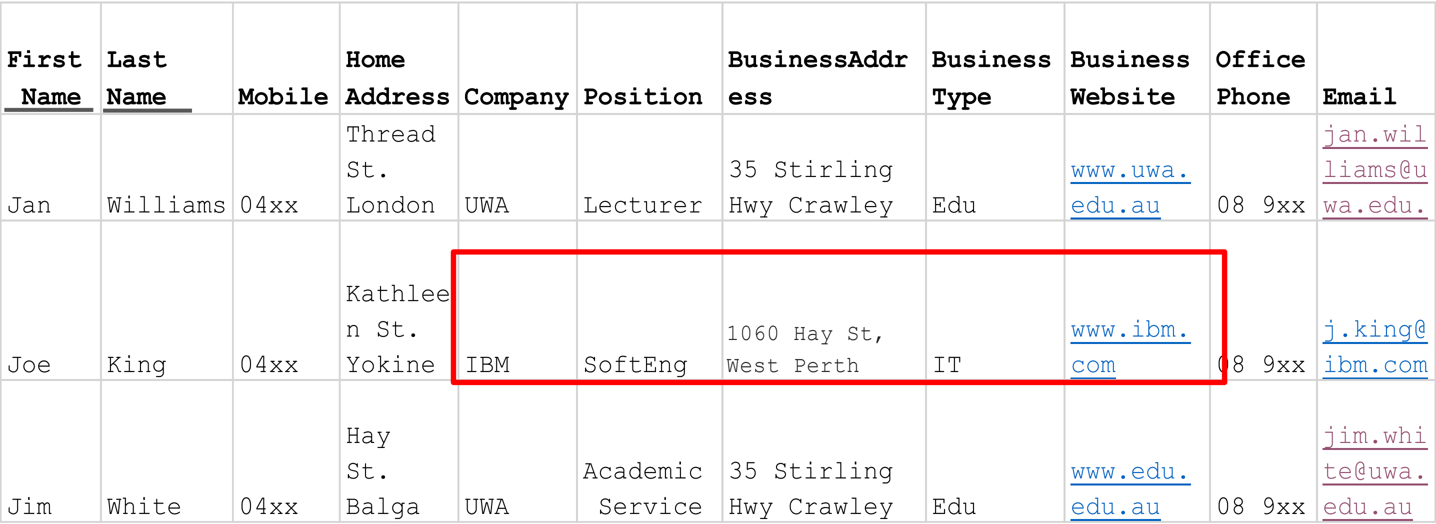
First Name	Last Name	Mobile	Home Address	Company	Position	BusinessAddress	Business Type	Business Website	Office Phone	Email
Jan	Williams	04xx	Thread St. London	UWA	Lecturer	35 Stirling Hwy Crawley	Edu	www.uwa.edu.au	08 9xx	jan.williams@uwa.edu.
Joe	King	04xx	Kathleen St. Yokine	IBM	SoftEng	1060 Hay St, West Perth	IT	www.ibm.com	08 9xx	j.king@ibm.com
Jim	White	04xx	Hay St. Balga	UWA	Academic Service	35 Stirling Hwy Crawley	Edu	www.edu.edu.au	08 9xx	jim.white@uwa.edu.au

Deletion Anomaly

- Unintended loss of data due to deletion of other data

  - because of unnecessary coupling of data

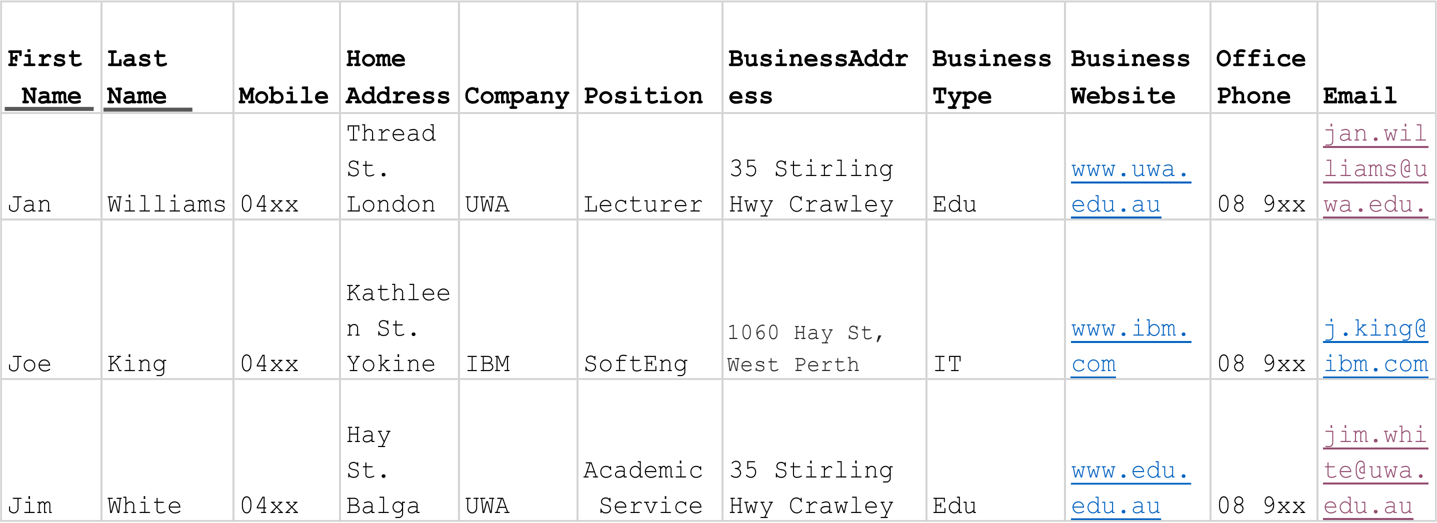
Example: deleting Joe King's detail will also delete IBM's information from the database



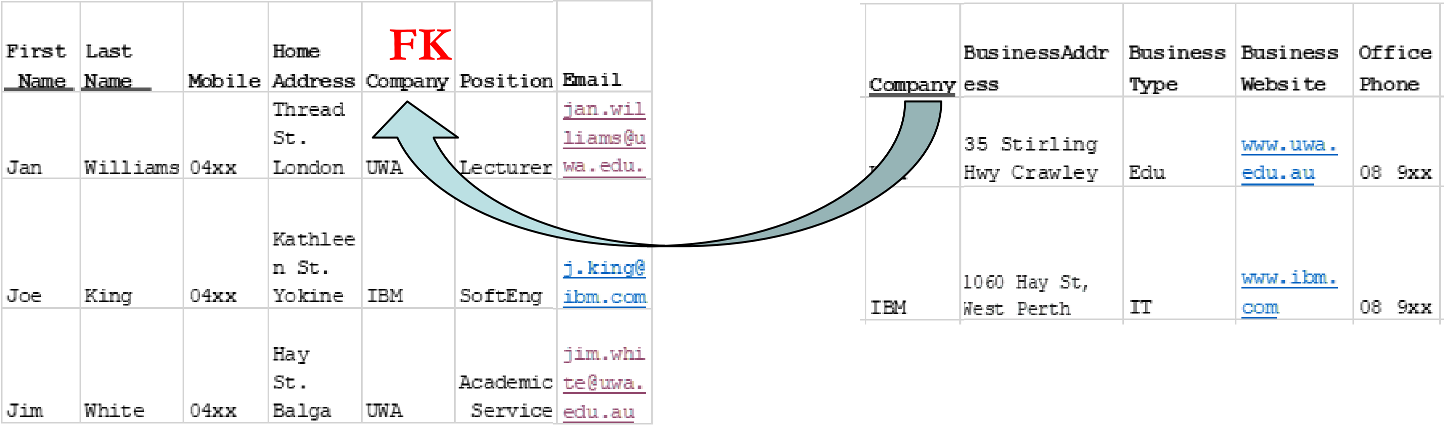
First Name	Last Name	Mobile	Home Address	Company	Position	BusinessAddress	Business Type	Business Website	Office Phone	Email
Jan	Williams	04xx	Thread St. London	UWA	Lecturer	35 Stirling Hwy Crawley	Edu	www.uwa.edu.au	08 9xx	jan.williams@uwa.edu.
Joe	King	04xx	Kathleen St. Yokine	IBM	SoftEng	1060 Hay St, West Perth	IT	www.ibm.com	08 9xx	j.king@ibm.com
Jim	White	04xx	Hay St. Balga	UWA	Academic Service	35 Stirling Hwy Crawley	Edu	www.edu.edu.au	08 9xx	jim.white@uwa.edu.au

What is the problem?

- Unnecessary coupling of data
- Too much information in the table
- Better solution would be to divide the information into two tables
- This leads us to the concept of Normalisation



First Name	Last Name	Mobile	Home Address	Company	Position	BusinessAddress	Business Type	Business Website	Office Phone	Email
Jan	Williams	04xx	Thread St. London	UWA	Lecturer	35 Stirling Hwy Crawley	Edu	www.uwa.edu.au	08 9xx	jan.williams@uwa.edu.
Joe	King	04xx	Kathleen St. Yokine	IBM	SoftEng	1060 Hay St, West Perth	IT	www.ibm.com	08 9xx	j.king@ibm.com
Jim	White	04xx	Hay St. Balga	UWA	Academic Service	35 Stirling Hwy Crawley	Edu	www.edu.edu.au	08 9xx	jim.white@uwa.edu.au



First Name	Last Name	Mobile	Home Address	Company	Position	Email
Jan	Williams	04xx	Thread St. London	UWA	Lecturer	jan.williams@uwa.edu.
Joe	King	04xx	Kathleen St. Yokine	IBM	SoftEng	j.king@ibm.com
Jim	White	04xx	Hay St. Balga	UWA	Academic Service	jim.white@uwa.edu.au

Company	BusinessAddress	Business Type	Business Website	Office Phone
[illegible]	35 Stirling Hwy Crawley	Edu	www.uwa.edu.au	08 9xx
IBM	1060 Hay St, West Perth	IT	www.ibm.com	08 9xx

Normalisation

- A process that organises entities/relations/tables, attributes and relationships in a database such that data anomalies are eliminated

  - divides entities/tables and assign attributes to minimise data dependencies and redundancies

  - involves assessment of attributes in relations based on the concept of Functional Dependency and reassign them

  - works through a series of stages to satisfy certain condition(s) called Normal Forms

- Normalisation is a process to evaluate and correct a database design to eliminate unnecessary/unwanted data dependencies and/or redundancies

Normalization Objectives

- Objective is to ensure that each table conforms to the concept of well-formed relations
  - Each table represents a single subject
  - Each row/column intersection contains only one value and not a group of values
  - No data item will be unnecessarily stored in more than one table
  - All non-key attributes in a table are dependent on the PK
  - Each table has no insertion, update, or deletion anomalies

Normalisation Process

- Assess each relation/table in a database against criteria for a series of Normal Forms

  - the First Normal Form (1NF), the Second Normal Form (2NF), and so on.

  - generally, a higher normal form is better than a lower normal form, *i.e.*,

    (*n+1*)th NF is better than (*n*)th NF

- Ensure that each relation/table are in at least 3NF (satisfy conditions for 3NF) – sufficient to prevent anomalies
  - Identifying the dependencies of attributes in a relation
  - Progressively breaking the relation up into a new set of relations

Normal Forms

1st Normal Form (1NF)

2nd Normal Form (2NF)

3rd Normal Form (3NF)

In practice, most database systems are normalized up to here, sufficient for preventing anomalies. Our Focus in this unit

Boyce-Codd Normal Form (BCNF)

An extension of 3NF

4th Normal Forms (4NF)

5th Normal Form (5NF)

Domain/Key Normal Form (DKNF)

6th Normal Form (6NF)

Theoretical, and may be not reasonable design goals in practice

Normal Forms (2)

- A relation is in 1NF if and only if
  - There are **no repeating groups**
  - Each record must be **unique**
  - It has a **PK**
- A relation is in 2NF if and only if
  - It is in 1NF
  - There is **no partial dependency**
- A relation is in 3NF if and only if
  - It is in 2NF
  - There is **no transitive dependency**

Functional Dependency

- A relationship between two attributes, where knowing the value of one attribute makes it possible to determine the value of another.
- **Formal definition**: 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*, represented as $A \rightarrow B$. *A* is called determinant and *B* is called dependent. For any two tuples *t1* and *t2*, if *t1* and *t2* have same values for attribute *A*, then they must have same values for attribute *B*.

$$t1[A] = t2[A] \text{ implies } t1[B] = t2[B]$$

Example:

– ISBN → Book Title

– Postcode → State

Because of the Entity Integrity constraint all non-key attributes are functionally dependent on the PK

ISBN & Postcode are determinants, Book Title & State are dependents.

Functional Dependency (2)

- **Partial dependency**: A type of functional dependency in which **the determinant is a part of the primary key**

  For example, if $(A, B) \rightarrow (C, D)$, $B \rightarrow C$, and (A, B) is the primary key, then the functional dependence $B \rightarrow C$ is a partial dependency because only part of the primary key (B) is needed to determine the value of C.

- **Transitive dependency**: An attribute (not a PK or not a part of PK) is dependent on another attribute (determinant) that is not part of the primary key
  - More difficult to identify among a set of data
  - Occur only when a functional dependence exists among non-key attributes,
  - Determinant and dependents both are non-key

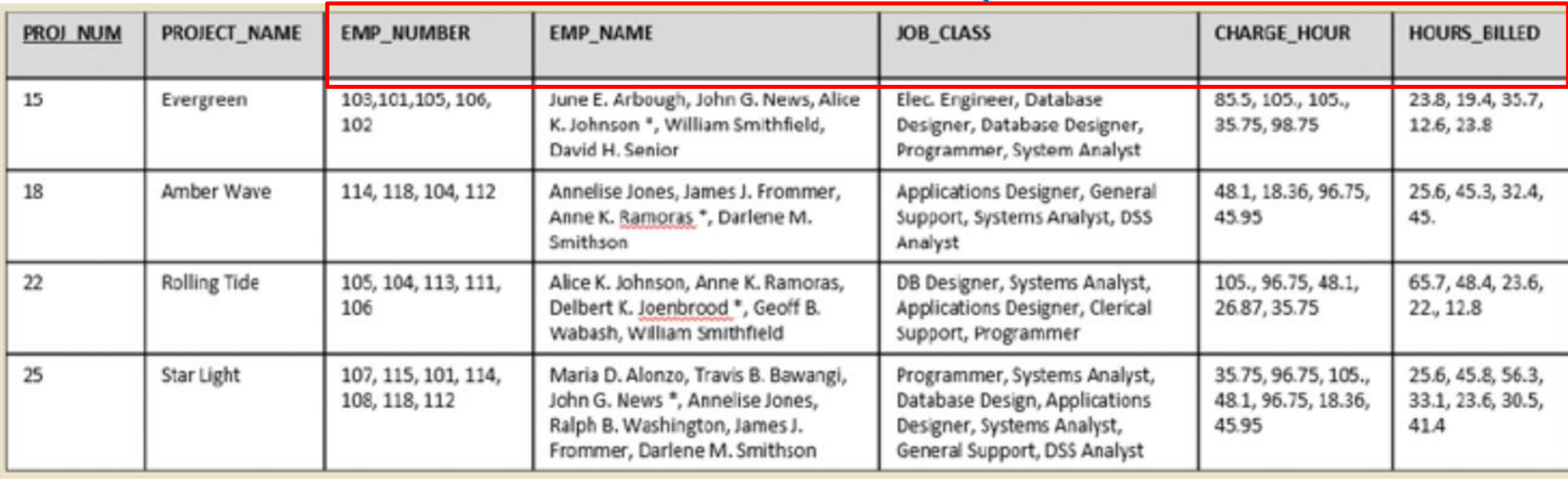
Normalisation Process

- Let's look at the normalisation process with a **Case Study**
- BCG company manages building projects
- Purpose of BCG's data

  1. Indicate the number of hours spent by each employee on different projects
  2. Indicate the hourly rate for each employee
  3. Generate report about each project

BCG – Base Data

Multivalued attributes, repeating groups



PROJ_NUM	PROJECT_NAME	EMP_NUMBER	EMP_NAME	JOB_CLASS	CHARGE_HOUR	HOURS_BILLED
15	Evergreen	103,101,105, 106, 102	June E. Arbough, John G. News, Alice K. Johnson *, William Smithfield, David H. Senior	Elec. Engineer, Database Designer, Database Designer, Programmer, System Analyst	85.5, 105., 105., 35.75, 98.75	23.8, 19.4, 35.7, 12.6, 23.8
18	Amber Wave	114, 118, 104, 112	Annelise Jones, James J. Frommer, Anne K. Ramoras *, Darlene M. Smithson	Applications Designer, General Support, Systems Analyst, DSS Analyst	48.1, 18.36, 96.75, 45.95	25.6, 45.3, 32.4, 45.
22	Rolling Tide	105, 104, 113, 111, 106	Alice K. Johnson, Anne K. Ramoras, Delbert K. Joenbrood *, Geoff B. Wabash, William Smithfield	DB Designer, Systems Analyst, Applications Designer, Clerical Support, Programmer	105., 96.75, 48.1, 26.87, 35.75	65.7, 48.4, 23.6, 22., 12.8
25	Star Light	107, 115, 101, 114, 108, 118, 112	Maria D. Alonzo, Travis B. Bawangi, John G. News *, Annelise Jones, Ralph B. Washington, James J. Frommer, Darlene M. Smithson	Programmer, Systems Analyst, Database Design, Applications Designer, Systems Analyst, General Support, DSS Analyst	35.75, 96.75, 105., 48.1, 96.75, 18.36, 45.95	25.6, 45.8, 56.3, 33.1, 23.6, 30.5, 41.4

- Violates a basic rule of relational database – each row-column intersection must have a single value
- Suffers from insertion, deletion, and update anomalies.

First Normal Form (1NF)

A relation is in 1NF if and only if

- There are **no repeating groups**
- Each record must be **unique**
- It has a **PK**

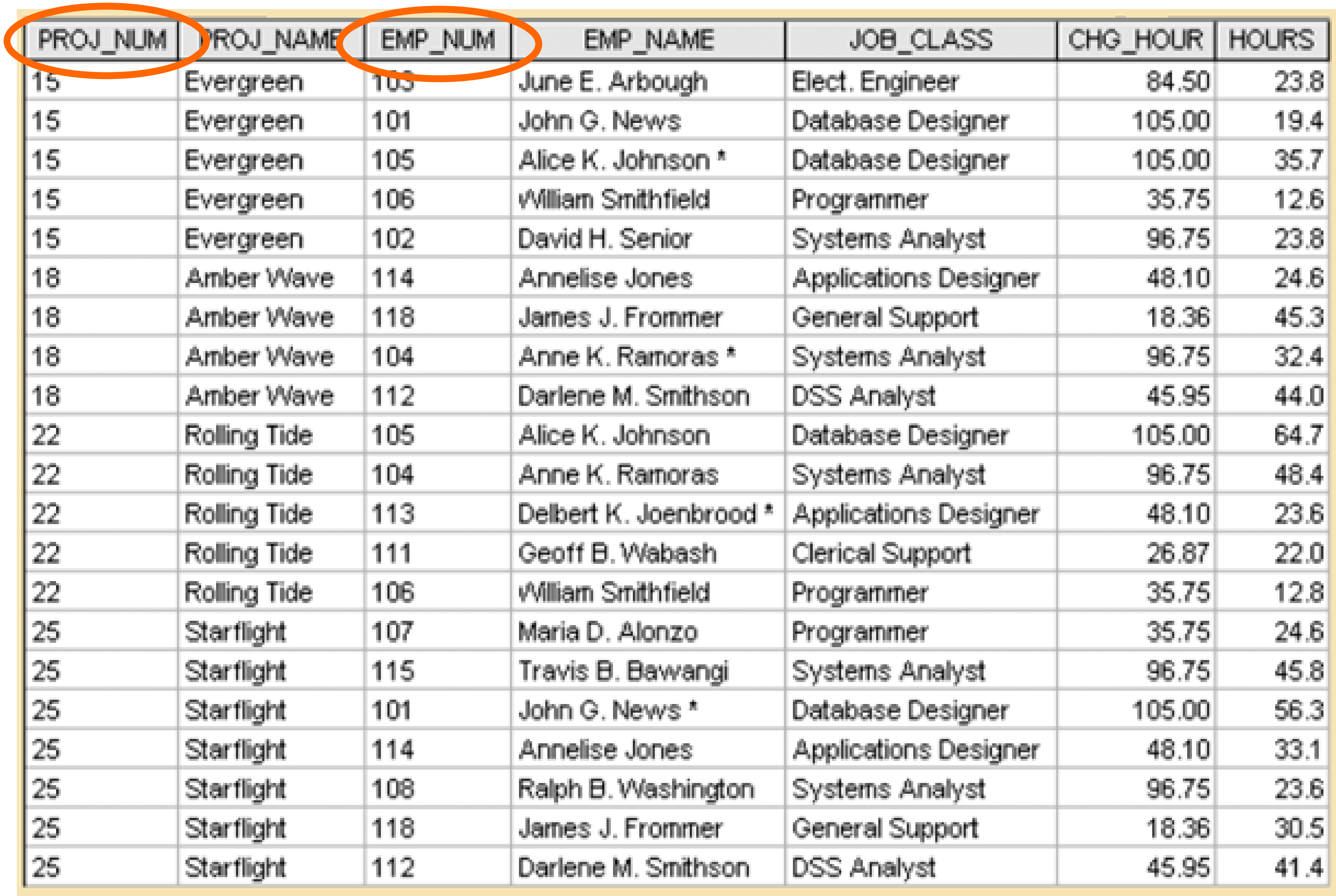
- Eliminate the repeating groups
- Convert the multivalued attributes into single-valued attributes.
  - they must be eliminated by making sure that each row defines a single entity instance, and
  - each row-column intersection has only a single value
- Identify the primary key

BCG Relation in 1NF

PK:

PROJ_NUM +

EMP_NUM



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	12.6
15	Evergreen	102	David H. Senior	Systems Analyst	96.75	23.8
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	32.4
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	48.4
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	26.87	22.0
22	Rolling Tide	106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
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	23.6
25	Starflight	118	James J. Frommer	General Support	18.36	30.5
25	Starflight	112	Darlene M. Smithson	DSS Analyst	45.95	41.4

Normalisation to 1NF

- Relation in the previous table is in 1NF
  - √ There are no repeating groups in the table, each cell has a single value
  - √ PK is defined

  All non-key attributes are dependent on the PK

- Repeat the same process for all relations in a database

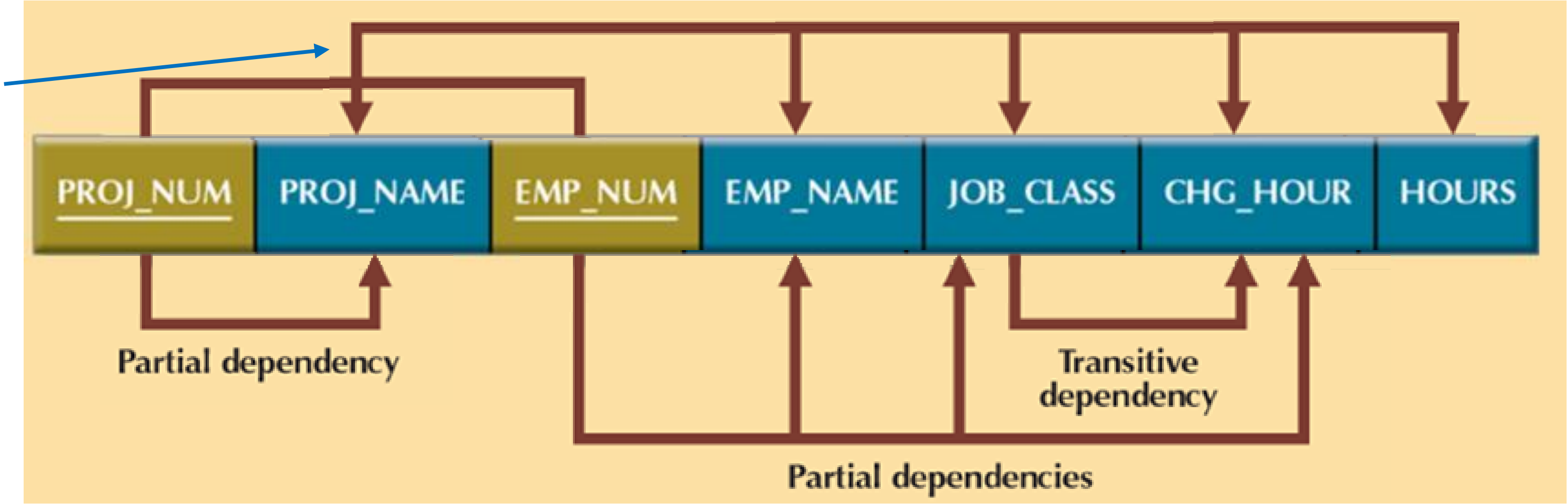
- What next?

  - Relations may contain partial or transitive dependencies

  - Analyse dependencies of attributes – Dependency Diagram

Dependency Diagram

- Visual representation of dependencies between attributes of a relation/entity/table



PK Dependency

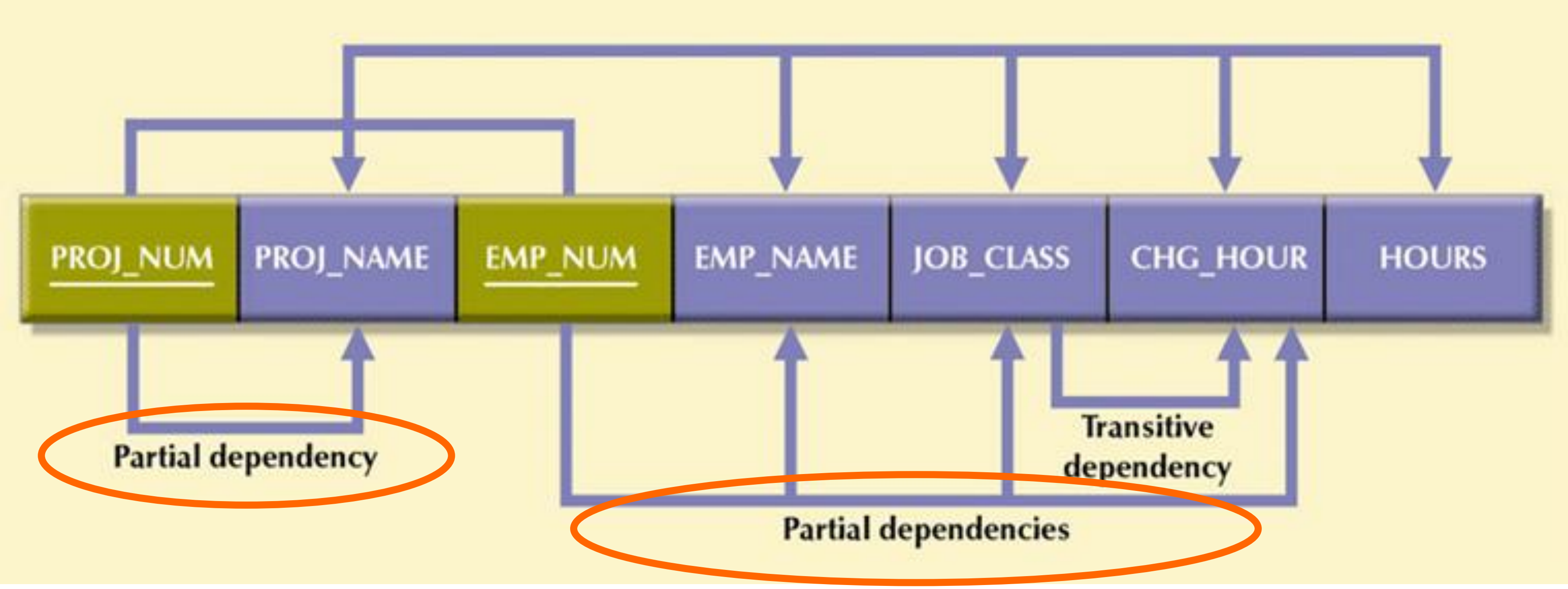
1NF (PROJ_NUM, EMP_NUM, PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOURS, HOURS)

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

TRANSITIVE DEPENDENCY:
(JOB CLASS → CHG_HOUR)

Second Normal Form (2NF)?

- What are the issues?



- Partial dependencies that can cause Insertion, Update and Deletion anomalies

Problems with the 1NF Relation

- **Update anomalies:** Modifying the JOB_CLASS for employee Annelise Jones requires updating many entries; otherwise, it will lead to data inconsistencies.
- **Insertion anomalies:** Adding a new employee requires the employee to be assigned to a project. If the employee is not yet assigned to a project, a phantom project must be created to complete the employee data entry.
- **Deletion anomalies:** Suppose that only one employee is associated with a given project. If that employee is deleted, the project information will also be deleted.

Second Normal Form (2NF)

A relation is in 2NF if and only if

- It is in **1NF**
- There is **no partial dependency**

- Check for partial dependencies in each relation/table
  - If any non-key attributes depend on a part of the PK
- Applicable only when the PK is a composite key.

  - relation is in 2NF if the PK is a single key, check for 3NF
- Goal of 2NF is to **remove partial dependencies**

Conversion to 2NF

- To achieve 2NF,

  1. Make new tables to eliminate partial dependencies
  2. Each partial dependency goes to a new table with **<u>key attribute of partial dependency</u>** and their corresponding **<u>dependent attributes</u>**
  3. **<u>Keep the key attributes of partial dependencies</u>** in the original table

BCG Database in 2NF

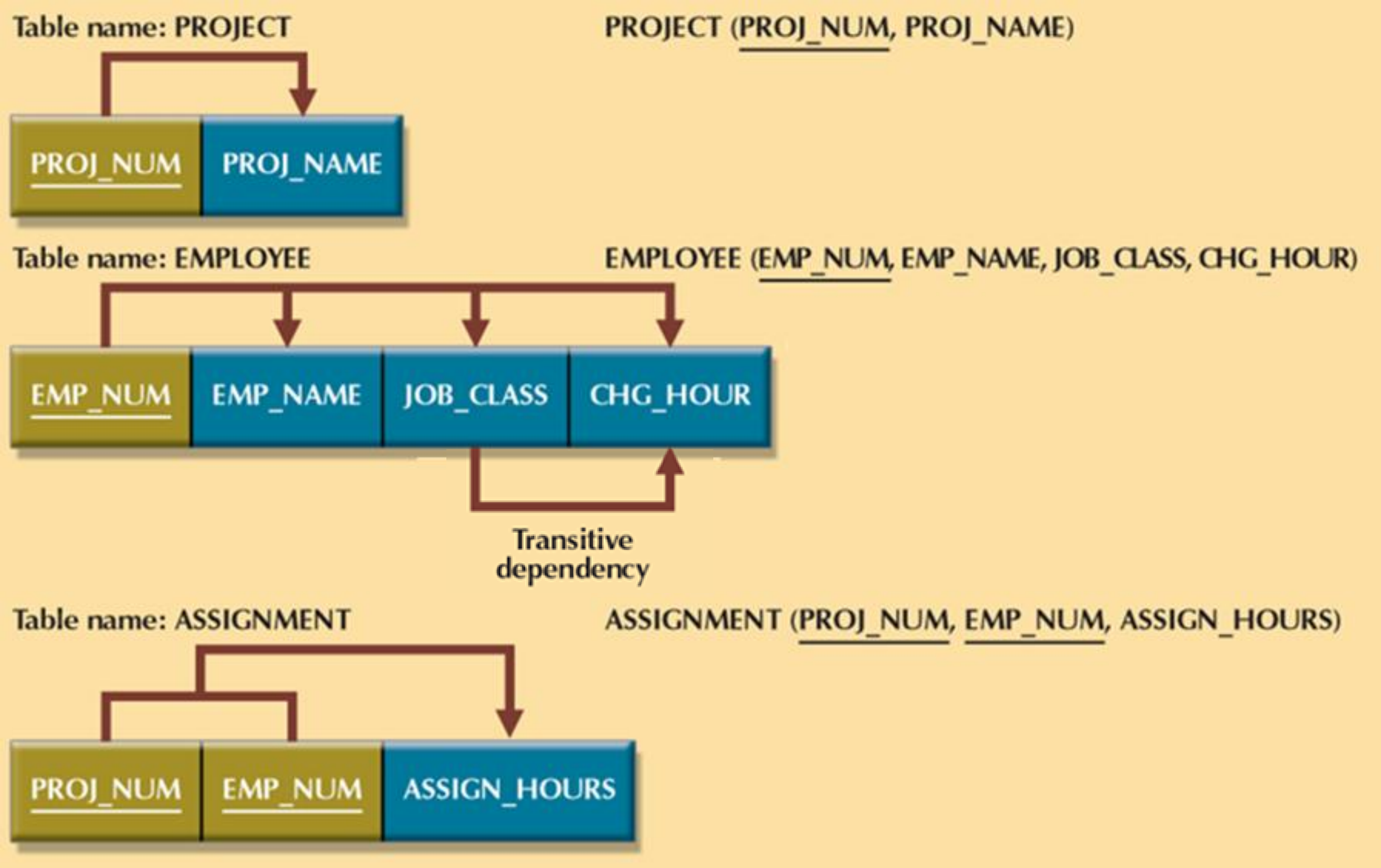


Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

PROJ_NUM | PROJ_NAME

Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR)

EMP_NUM | EMP_NAME | JOB_CLASS | CHG_HOUR

Transitive dependency

Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)

PROJ_NUM | EMP_NUM | ASSIGN_HOURS

2NF Summary

- **1NF is satisfied**
- **Includes no partial dependencies**
  - No attribute dependent on a portion of the PK
- Still possible to exhibit **transitive dependency**
  - Transitive dependency: an attribute (Not PK or part) is dependent on another attribute that is not part of the PK
  - Check if the relation is in 3NF

Third Normal Form (3NF)

- What are the issues?

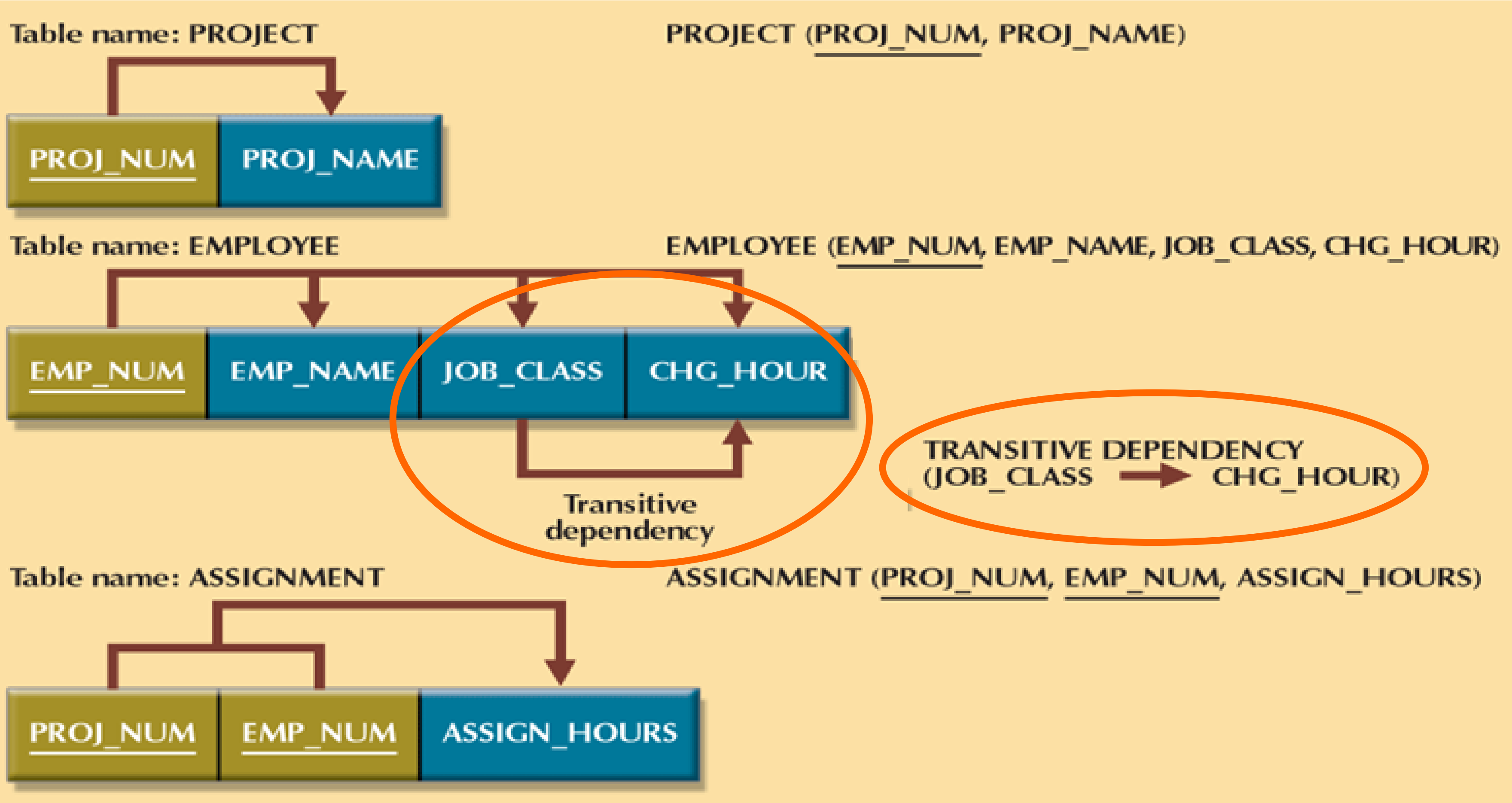


Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR)

TRANSITIVE DEPENDENCY
(JOB_CLASS → CHG_HOUR)

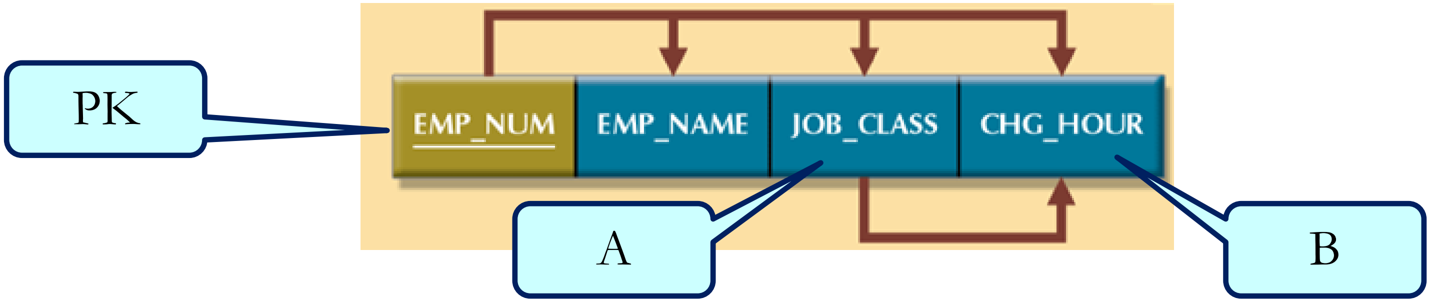
Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)

Third Normal Form (3NF) (2)

A relation is in 3NF if and only if

- It is in **2NF**
- There is **no transitive dependency**



- Here, attribute B is dependent on attribute A which is not part of the primary key
- Both (A and B) are non-key attributes (transitive).
- So it is not in 3NF,

Conversion to 3NF

- To achieve 3NF,
  - Identify the **transitive determinant** in the table
  - For each transitive determinant, identify the **dependent attributes**
- What to do …
  1. **Remove dependent attributes** from the table
  2. Place them into a **new table** with the ***transitive determinant as the PK***
  3. Leave the determinant in the original table

BCG Database in 3NF

FIGURE 6.5 THIRD NORMAL FORM (3NF) CONVERSION RESULTS

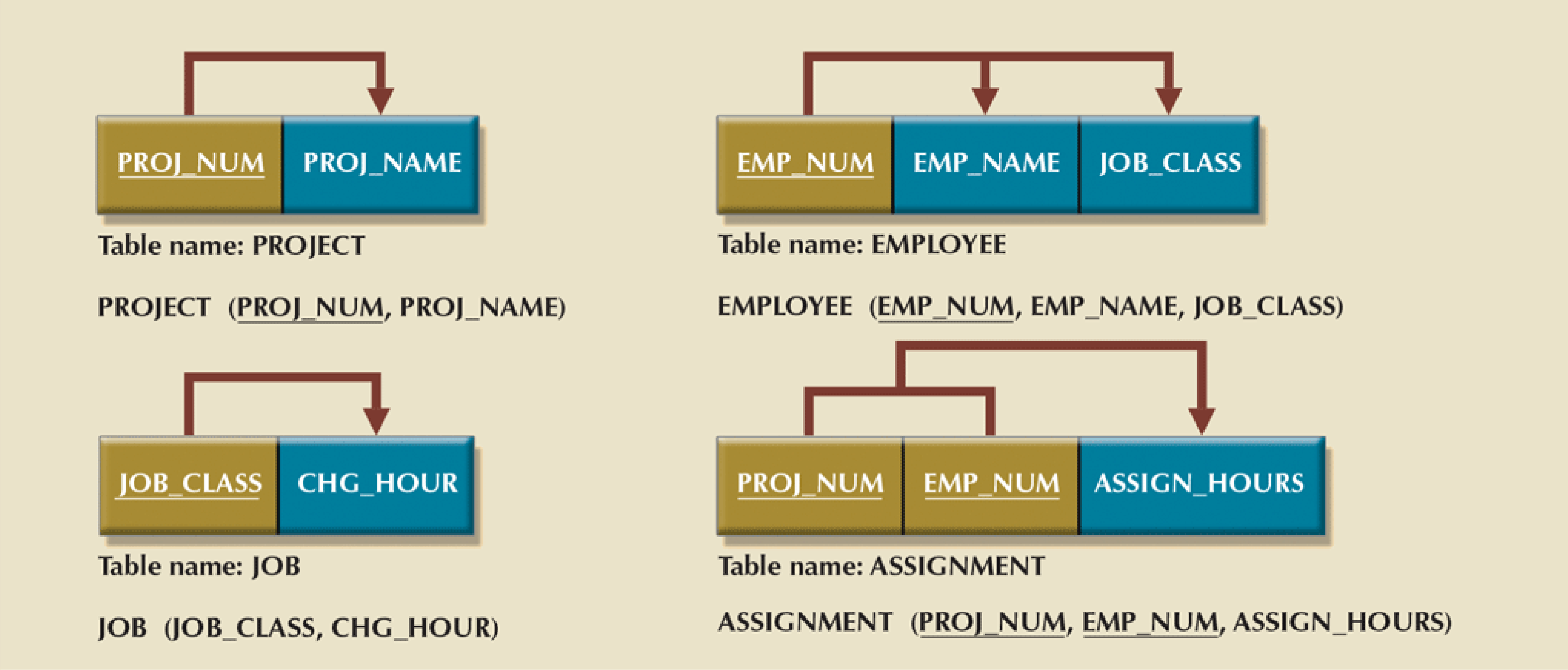


Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

Table name: EMPLOYEE

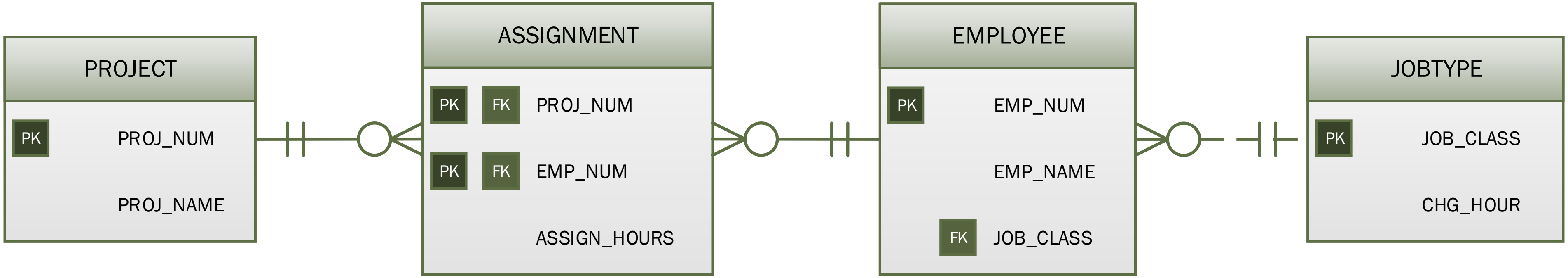
EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)

Table name: JOB

JOB (JOB_CLASS, CHG_HOUR)

Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)



More on Normalisation

- Part of **design and maintenance** process
  - Prevents the creation of improper table structures
  - Many real-world DBs are improperly designed or modified over time
- Conceptual ERD provides the big picture (macro view)
- Normalisation focused on specific entities (micro view)
- ERD and normalization are used concurrently
- Logical ERD (to-be implemented) includes relations and relationships after normalisation

More on Normalisation (2)

- There are <u>higher normal forms</u> (BCNF, 4NF, 5NF, 6NF)
- Higher normal forms are:
  - useful on occasions
  - not always desirable
  - not covered in this unit

More on Normalisation (3)

- Effective DB design is about
  - Correctness
  - Consistency
  - Reduced redundancy
  - Efficiency
- Normalisation creates more tables to ensure correctness, consistency and reduced redundancy
- More tables require more "joins" to answer queries

  - affects efficiency in retrieving data from the database
- A trade-off is required, too much normalisation may not be good – up to 3NF is enough in most real-world databases.

Denormalisation

- A process of changing a relation from higher-level NF to lower-level NF

  - to improve processing speed

  - potentially yields data anomalies

- Denormalisation implies redundancy
  - Reduces “joins” to achieve efficiency but may cause difficulty in insertion, update and/or deletion.

Normalisation to 2NF (Review)

A relation is in 2NF if and only if

- It is in **1NF**
- There is **no partial dependency**

- Partial dependency occurs only when the PK is a composite key (includes more than one attributes)
- What if we add an artificial key as ID and make it the PK?

  - the relation will be in 2NF (as the PK is not a composite key) but can result in many transitive dependencies

Natural vs Artificial PK

- **Natural PK**: is formed of attributes that already exist in the real world, such as (Name + DoB).
- **Artificial PK**: is created by certain rules, such as automatically increment numbers. They have no real meaning. E.g., (EmpID)

Natural PK

Pros:

- Has real meaning

Cons:

- Can be difficult to choose good key
- Cost more space, the combination can be big

Artificial PK

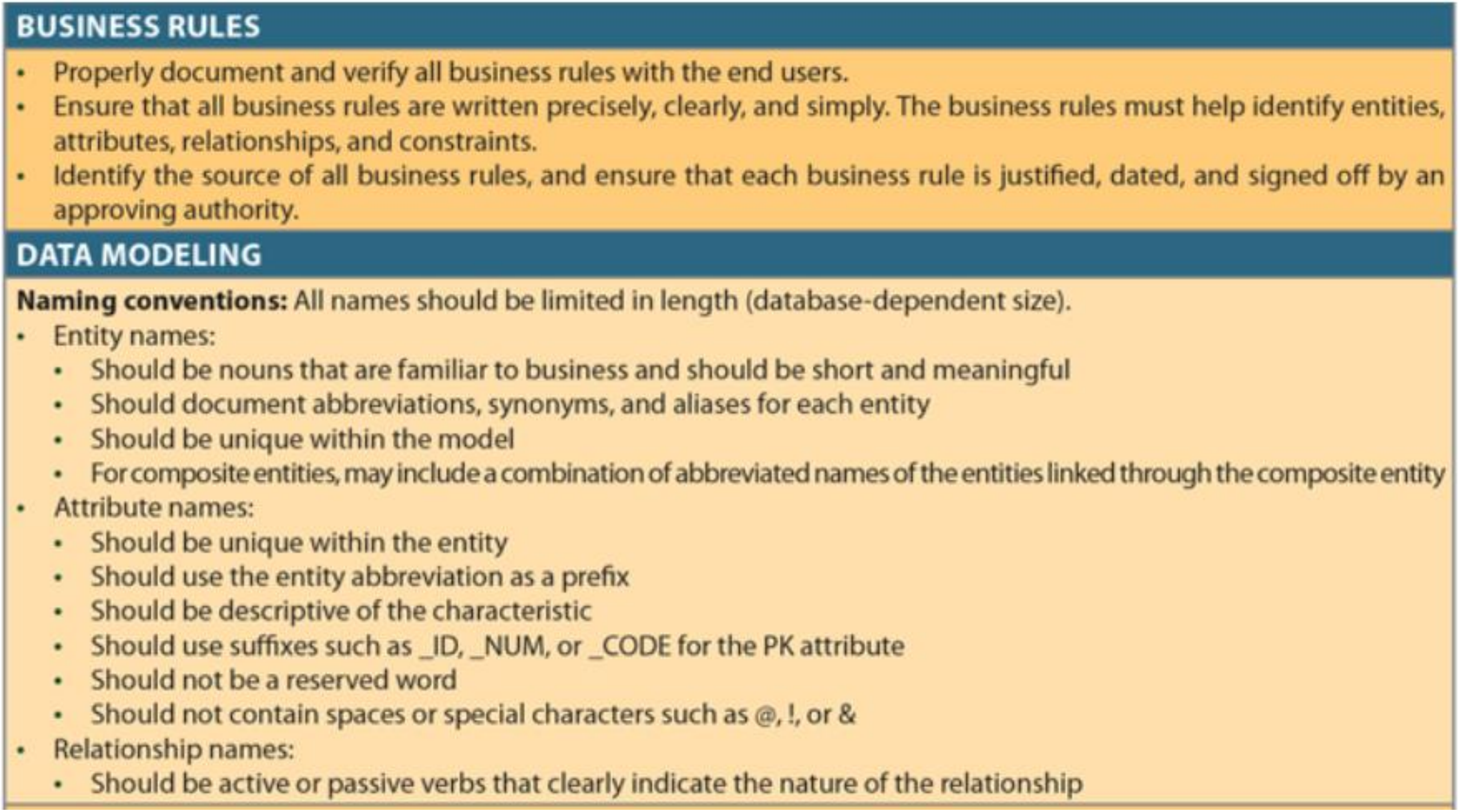
Pros:

- Doesn’t carry no real meaning.
- Easy to choose, e.g. auto increment

Cons:

- Artificially added with no meaning, i.e. you usually need to fetch more columns or join tables to obtain the information

Data-Modeling Checklist



BUSINESS RULES

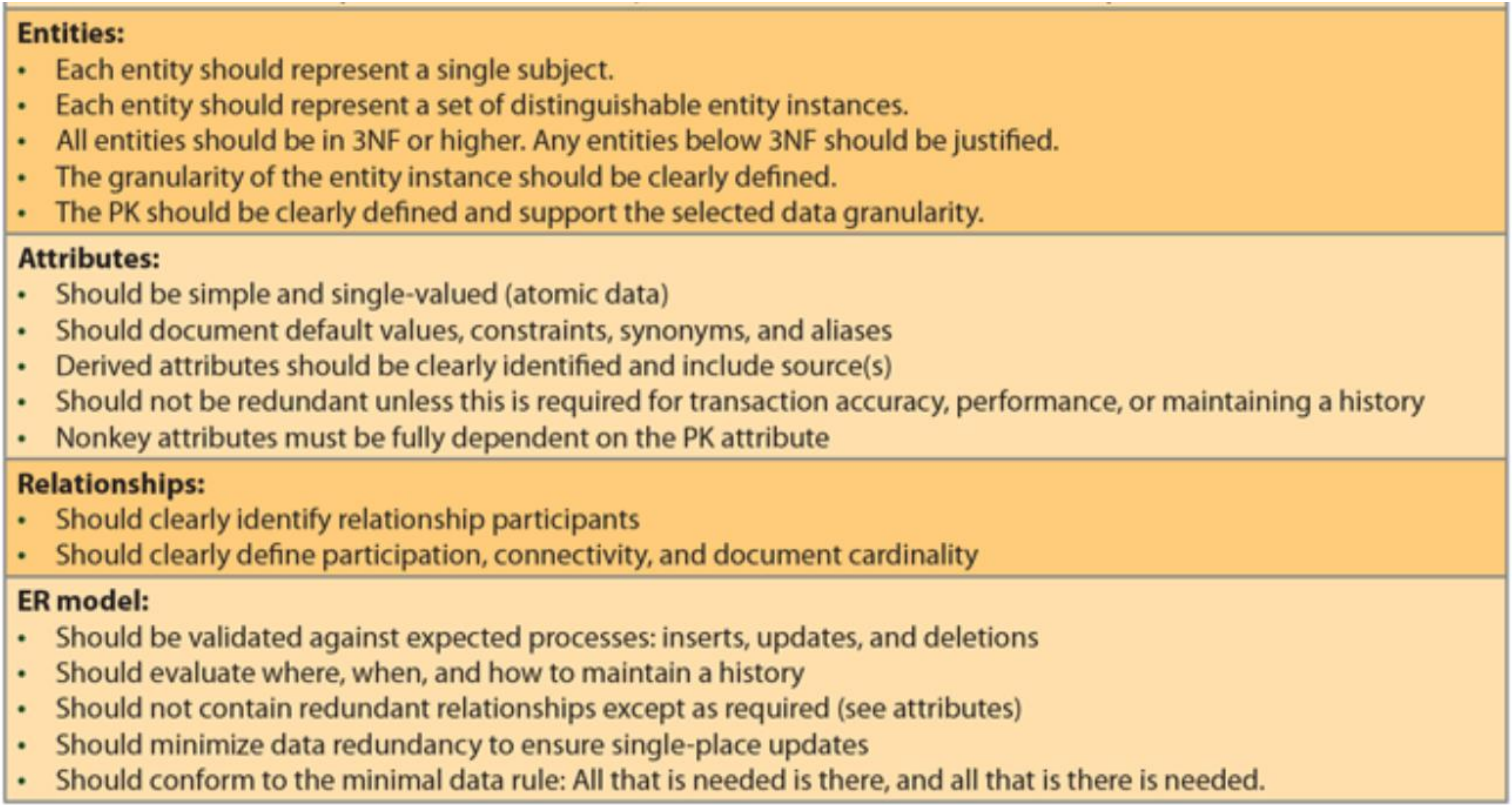
- Properly document and verify all business rules with the end users.
- Ensure that all business rules are written precisely, clearly, and simply. The business rules must help identify entities, attributes, relationships, and constraints.
- Identify the source of all business rules, and ensure that each business rule is justified, dated, and signed off by an approving authority.

DATA MODELING

Naming conventions: All names should be limited in length (database-dependent size).

- Entity names:
  - Should be nouns that are familiar to business and should be short and meaningful
  - Should document abbreviations, synonyms, and aliases for each entity
  - Should be unique within the model
  - For composite entities, may include a combination of abbreviated names of the entities linked through the composite entity
- Attribute names:
  - Should be unique within the entity
  - Should use the entity abbreviation as a prefix
  - Should be descriptive of the characteristic
  - Should use suffixes such as _ID, _NUM, or _CODE for the PK attribute
  - Should not be a reserved word
  - Should not contain spaces or special characters such as @, !, or &
- Relationship names:
  - Should be active or passive verbs that clearly indicate the nature of the relationship

Data-Modeling Checklist (2)



Entities:

- Each entity should represent a single subject.
- Each entity should represent a set of distinguishable entity instances.
- All entities should be in 3NF or higher. Any entities below 3NF should be justified.
- The granularity of the entity instance should be clearly defined.
- The PK should be clearly defined and support the selected data granularity.

Attributes:

- Should be simple and single-valued (atomic data)
- Should document default values, constraints, synonyms, and aliases
- Derived attributes should be clearly identified and include source(s)
- Should not be redundant unless this is required for transaction accuracy, performance, or maintaining a history
- Nonkey attributes must be fully dependent on the PK attribute

Relationships:

- Should clearly identify relationship participants
- Should clearly define participation, connectivity, and document cardinality

ER model:

- Should be validated against expected processes: inserts, updates, and deletions
- Should evaluate where, when, and how to maintain a history
- Should not contain redundant relationships except as required (see attributes)
- Should minimize data redundancy to ensure single-place updates
- Should conform to the minimal data rule: All that is needed is there, and all that is there is needed.

Summary

- A relation is in 1NF if and only if
  - There is no repeating group
  - All attributes are dependent on the PK
- A relation is in 2NF if and only if
  - It is in 1NF
  - There is no partial dependency
- A relation is in 3NF if and only if
  - It is in 2NF
  - There is no transitive dependency
- Normalize tables to <u>reduce data redundancies</u>
- Denormalize to <u>reduce joins</u>

This Week’s OnTrack Tasks

- 4.1P Database Normalisation

  - Dependency Diagram

  - 1NF, 2NF and 3NF

- 4.2C Miniproject Part-1 - Database Design and Normalisation

  - Database modelling for a business organisation of your choice

  - Opportunity to experience data modelling in real-world

- Please check the task sheets and start working on them

Next Week

- Introduction to SQL

Thank you

See you next week

Any questions/comments?

Readings and References:

- Chapter 6

Database Systems : Design, Implementation, & Management 13TH EDITION, by Carlos Coronel, Steven Morris