COMM1822

Term 2 2022

Introduction to Databases for Business Analytics

Week 4 Normalisatio

Lecturer-in-Charge: Kam-Fung (Henry) Cheung
Email: kf.cheung@unsw.edu.au
Tutors: Theresa Tran
Liam Li Chen
Kathy Xu
PASS Leader: Srilekha Chandrashekara Kolaki

Copyright

- There are some file-sharing websites that specialise in buying and selling academic work to and from university students.

- If you upload your original wor other student downloads and presents it as their own eit **ight be found guilty of collusion — even years after graduatio**

- These file-sharing websites may also accept purchase of course materials, **such as copies of lecture slides and tutorial handouts. By law, the copyright on course materials, developed by UNSW staff in the course of their employment, belongs to UNSW. It constitutes copyright infringement, if not academic misconduct, to trade these materials.**

Acknowledgement of Country

UNSW Business School acknowledges the Bidjigal (Kensington campus) and Gadigal (City campus) the traditional custodians of the lands where each campus is located.

We acknowledge all Aboriginal and Torres St Islander Elders, past and present and thei communities who have shared and practiced their teachings over thousands of years including business practices.

We recognise Aboriginal and Torres Strait Islander people's ongoing leadership and contributions, including to business, education and industry.

UNSW Business School. (2022, May 7). *Acknowledgement of Country* [online video]. Retrieved from https://vimeo.com/369229957/d995d8087f

Chapter 6

ation of Database Tables

Plan: W4 Learnings

- **Normalisation** (or Normali**z**ation)
- **Functional Dependencies**
- **Normal Forms**
  - 1NF
  - 2NF
  - 3NF
  - BCNF
- **Denormalisation**

Conceptual Modelling and Logical Modelling

- A **conceptual data model** (e.g., ER model) represents the **conceptual view** of organisational data.
- A **logical data model** (e.g., relational model) describes the organisational data in **implementation** in a DBMS. (the logical m nt of any particular DBMS)
- So far,
  - We have learned how to develop ER models (conceptual).
  - We have learned how to convert ER models to relational schema (logical).
- **The question remains: How good are the attributes in the relational schema?**

The Needs and Outcomes of Normalisation

- **<u>Need</u> the process of normalisation is when you need to design a new database structure**
  - Analyse the relationship among the attributes within each entity
  - Determine if the stru
- **Improve the existing da** **n 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.

Normalisation (1)

- **Normalisation** is a process for evaluating and correcting table structures to **minimise data redundancies**, thereby **reducing the likelihood of data anomalies**.
- **Normalisation** is …
  - a process for converting a relati
  - a **process** that is accomplished in **stages**.
  - a technique that is used to **define** "**goodness**" (or "b lation.
  - 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).**

Normalisation (2)

Table name: STUDENT

zID	…	Sec_Email

Table name: COURSE_ENROL

zID	CourseID	…	Sec_Email

Redundancy

- **Redundancy** occurs when data about a **one entity is recorded more than once** in a database.
- Database designers aim to **r** abase should not store same data several times) to save s
- Evaluating and correcting ta ta redundancies.

If **data redundancy exists**, **be anomalies**.

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

Normalisation (3)

A Normal Form…

- …is a certain **state** of a **relation**.
- …can be determined by applying **rules regarding dependencies**.
- …uses a concept calle **ey**.

Normal forms

- First normal form (1NF)
- Second normal form (2NF)
- Third normal form (3NF)
- Boyce-Codd normal form (BCNF)
- [Fourth normal form (4NF)]

Normalisation

De-normalisation

Why denormalisation?

To improve greater performance with greater data redundancy. (More will be covered in Big Data)

Normal Forms

Table 6.2: Normal Forms		
Normal Form	**Characteristic**	**Section**
First normal form (1NF)	s, and PK identified	6-3a
Second normal form (2NF)	1NF and no partial	6-3b
Third normal form (3NF)	2NF and no transiti	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

Our focus

Functional Dependency

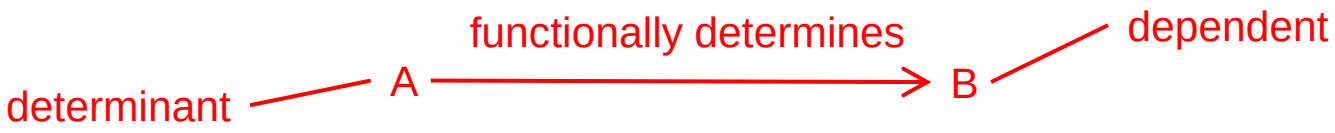
UNSW SYDNEY

Functional Dependency (FD)

Functional Dependencies …

- …are **relationships between attributes** in a relation.
- …are the **semanti** in a relation.
- …**can be inferred** **y** by applying a set of **inference rules** (next slides).

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 (generalised definition)	ttribute B (that is, B is on A) if all (generalised the table that agree in value for attribut value for attribute B.
Fully functional dependence (composite key)	If attribute dependent on a composite key A but not on any subset of that composite key, the attribute B is fully functionally dependent on A.



Postcode → *State*; e.g., “2052” → “NSW”, but not “2052” → ”VIC”

Functional Dependence & Normalisation

Two types of functional dependencies:

- A **partial dependency** exists when there is a functional dependence in which **the determinant is only 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 becaus eeded to determine the value of C. Partial dependencies tend to b fy.
- A **transitive dependency** exists when there are fu dencies such that $X \rightarrow Y$, $Y \rightarrow Z$, and X is the primary key. In that case, the depenate 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.

nonkey to nonkey

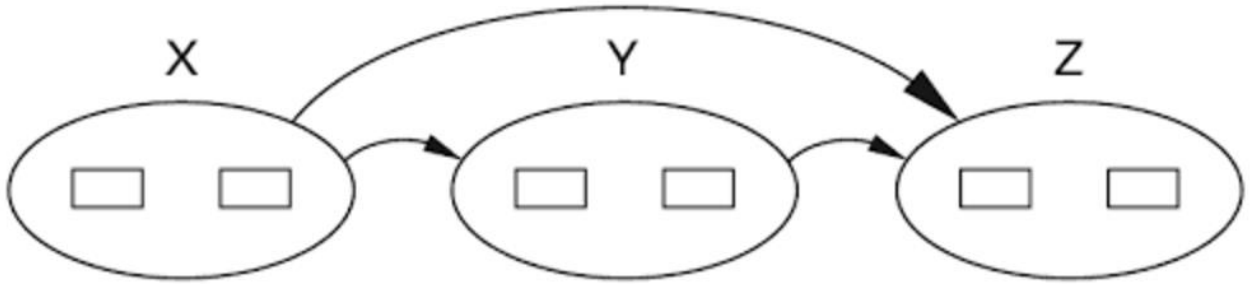
Transitivity and Transitive Dependency

If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

Example

If zID → MobileNumber and then zID → Name

z1234567 → 0466 772 123 and 0466 772 12 then z1234567 → Kaiser



Normalisation and Normal Forms

UNSW SYDNEY

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 tab y
  - Each table is void of insertion,
- Ensures that all tables are in at least 3NF (ru
- 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

Lossless Decomposition and Normal Forms

- Our aim is to **decompose** relations/tables so to **reduce size/redundancy**.

- We use **inferences rul** n **process**.

- We need to be sure that the decompo nents (tables/relations) have the **lossless** join property (i.e., d components could be joined back together to the original table/relation).

Decomposition Example

Which of the two decompositions of SUPPLIER relation is better? (i.e., which one could be joined back together to the original relation?)

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 s position.

The following slide shows a table with c espond to the reporting requirements but is not “normalised”.

Example: Table Problems

- The project number is intended to be (part of) a **PK**, but it **contains NULLs**.
- The table **has data**
- The table entries **in** **encies** and **anomalies** (addition, deletion, update ano

1 N

UNSW SYDNEY

Conversion to First Normal Form (1NF)

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

- A relation / table is in **1NF**
  - The key attributes are defined, i.e., (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 multivalue**

- 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 Partial Dependencies

Steps to Follow for 1NF

- **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 Primar**
  - All attributes are dependent on PROJ_NUM + EMP_N

- **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

So, which are the PK and dependencies?

Examine the Similarities and Differences of the Data

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

First Normal Form (1NF) 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

All attributes depend on the primary key.

Primary Key

Please note the notation

30

2N

UNSW SYDNEY

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 (E dependent on the entire PK).
  - The relation/table must be i

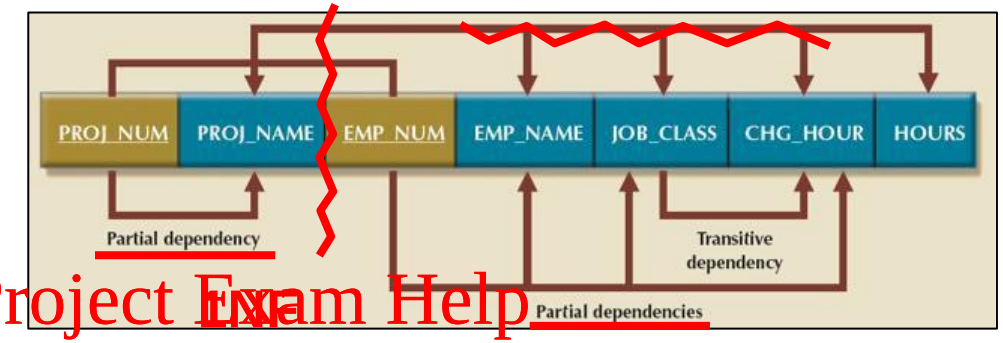
Hint: Look fo multiple times in *non-key fields*. This tells you that you have to single table. *In a well-designed database, the only data that is d fields used to connect tables.*

- 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).

Steps to Follow for 2NF

Step 1: Identify all key FDs components, especially partial dependency before breaking into smaller tables.

Step 2: Eliminate partial dependency



HOURS

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:

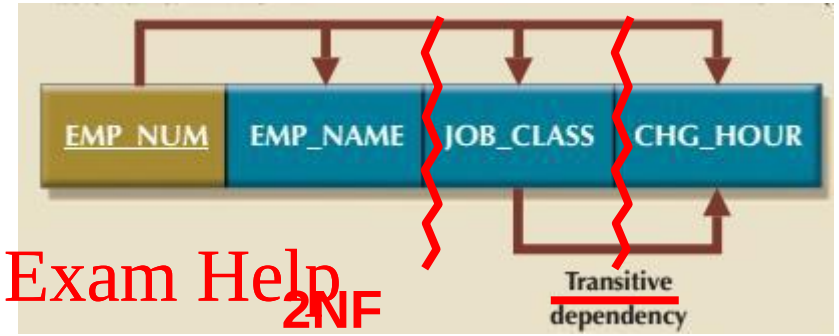
- It has **no transitive dependen** ed by other non-candidate-key attributes).
- The relation/table must be in 2NF

❑ 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

Steps to Follow for 3NF

Step 1: Analyse FDs, especially transitive dependencies (from 2NF)



Step 2: Remove transitive dependency.

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 **Codd (1974)**.
  - Sometimes called **3.5NF**
  - 3NF is always achievable, **BCNF is not always achievable (Beeri & Bernstein 1979)**.

- Candidate Key: Every dete e a candidate key
  - Same characteristics as primary key but not chos ary key
  - Equivalent to 3NF when the table contains only o y
  - 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

A Table That is in 3NF and NOT in BCNF

STUD_ID	STAFF_ID	CLASS

A **partial dependency:** The determinant is only part of the primary key.

Transitive dependency: An attribute functionally depends on another nonkey attribute (i.e., nonkey to nonkey)

Why is $C \rightarrow B$ not partial or transitive?

Not partial! Because C is the determinant of B, and not part of PK.
Not transitive! Because it involves a PK, i.e., B.
Thus, in 3NF

BCNF if, for every one of its dependencies $X \rightarrow Y$, one of the following conditions holds true:

- $X \rightarrow Y$ is a TRIVIAL functional dependency, i.e., Y is a subset of X
- X is a SUPERKEY

Why not in BCNF? (Hint: Look at $C \rightarrow B$)

- B is not part of C, i.e., B is NOT a subset of C
- C is not a superkey, as C CANNOT determines A or D on its own

A: STU_ID
B: STAFF_ID
C: CLASS_CODE
D: EN_GRADE

Change PK

STU_ID CLASS_CODE EN_GRADE

CLASS_CODE STAFF_ID

Normalisation and Database Design

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

Denormalisation

UNSW SYDNEY

Denormalisation

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

- Design goals
  - Creation of normalized relations
  - Processing requirements and speed
- Number of database ta es are decomposed to conform to normalisatio
  - Joining a larger number of tables:
    - Takes additional input/output (I/O) operati sing 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 (will be covered later)

Common Denormalisation Examples

(ZIP, CITY)

(Course, Credit)

storing WAM

Summary

- **Normalisation** is a **table design technique** aimed at minimising data redundancies.
- First three normal fo **3NF**) are most commonly used.
- Normalisation is an important pa **ly a part**—of the design process.
- Best practice: continue the iterative ER process until all entities and their attributes are defined and all equivalent **tables are in 3NF**.

W4 Learnings

- **Normalisation** (or Normalization)
- **Functional Dependencies**
- **Normal Forms**
  - 1NF
  - 2NF
  - 3NF
  - BCNF
- **Denormalisation**

Reference (Harvard)

Beeri, C. & Bernstein, P.A., 1979. ‘Computational problems related to the design of normal form relational schemas’, *ACM Transactions on Database Systems (TODS)*, vol. 4, no. 1, pp.30-59.

Codd, E.F., 1971. ‘Normalized data base structure: A brief tutorial’, In *Proceedings of the 1971 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control* (pp. 1-17).

Codd, E.F., 1974. ‘Recent investig , *Information Processing 74*, pp.1017-1021.

Questions

Source: keepmeme.com

Take-Home Exercise

A librarian has created the above table in an effort to create a “database”. However, there are several issues with the design.

1. Argue what potential problems there are with the table design.
2. Identify the PK(s) and draw the dependencies diagrams.
3. Normalise the relational model the 3NF.
4. Draw the ER diagram based on the 3NF.