MONASH University

MONASH
INFORMATION
TECHNOLOGY

Normalisation

Lindsay Smith

GROUP OF EIGHT AUSTRALIA

Sample Data

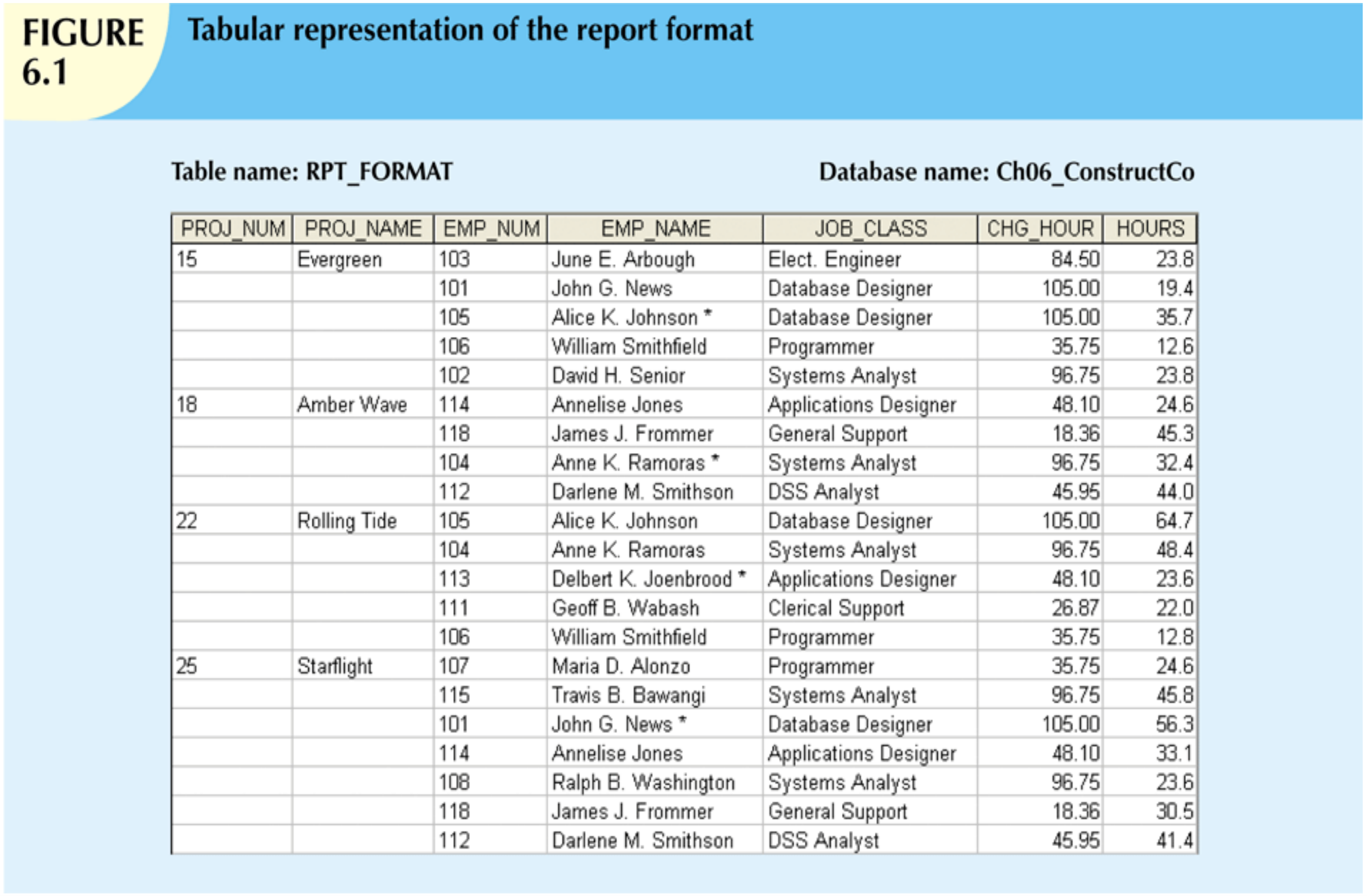


FIGURE 6.1 Tabular representation of the report format

Table name: RPT_FORMAT

Database name: Ch06_ConstructCo

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

Problems with data in Figure 6.1

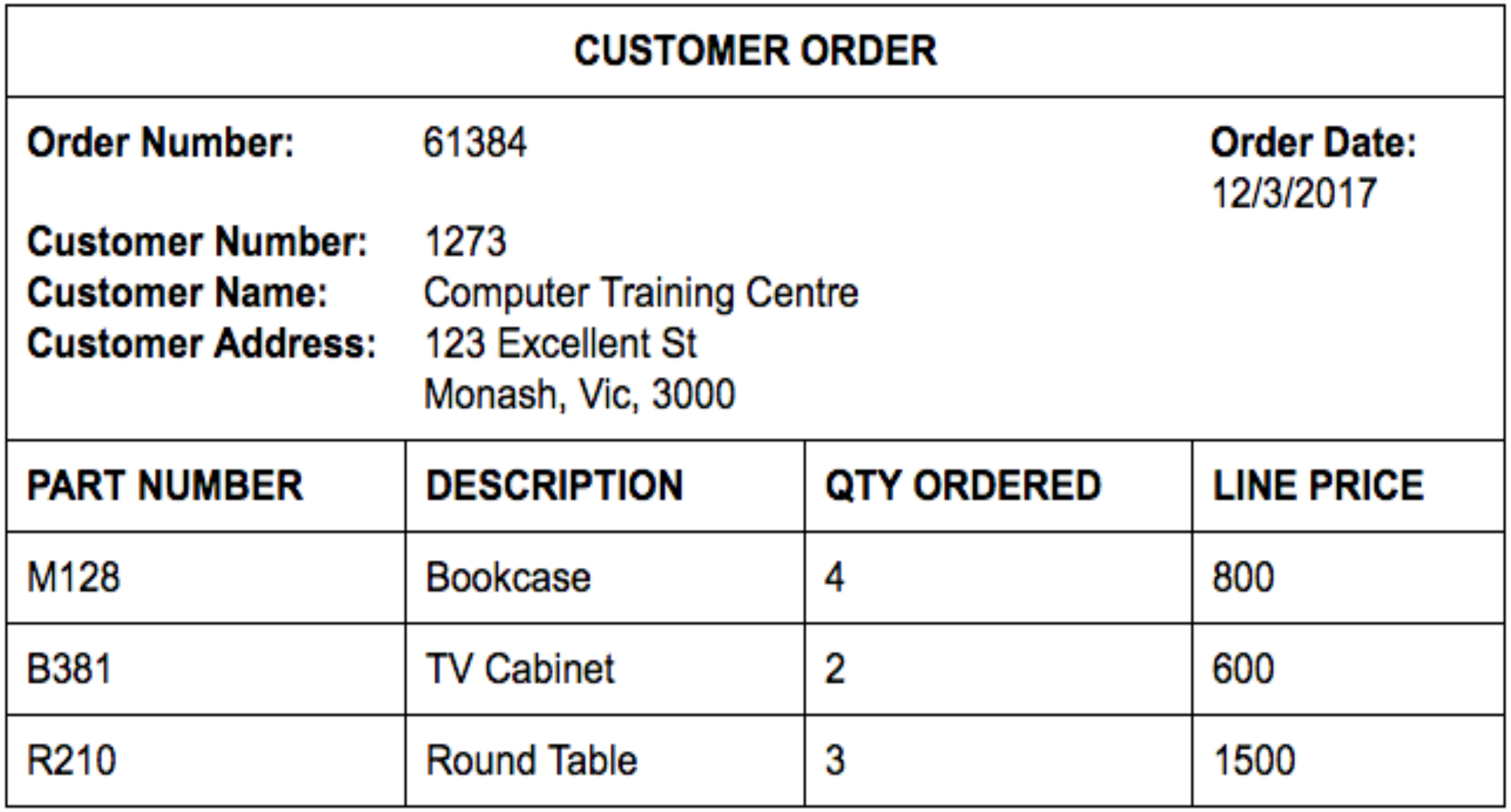
- PROJ_NUM intended to be **primary key**, but it contains nulls
- JOB_CLASS invites **entry errors** eg. Elec. Eng. vs Elect. Engineer vs E.E.
- Table has **redundant data**
  – Details of a charge per hour are repeated for every occurrence of job class
  – Every time an employee is assigned to a project emp name repeated
- Relations that contain redundant information may potentially suffer from several update anomalies
  – Types of update anomalies include:
    - **Insert Anomaly**
      –Insert a new employee only if they are assigned to a project
    - **Delete Anomaly**
      –Delete the last employee assigned to a project?
      –Delete the last employee of a particular job class?
    - **Modification (or update) Anomaly**
      –Update a job class hourly rate - need to update multiple rows

Data Normalisation

- Relations should be normalised in order to avoid anomalies which may occur when inserting, updating and deleting data.
- Normalisation is a *systematic series of steps* for progressively refining the data model.
- A formal approach to analysing relations based on their primary key (or candidate keys) and functional dependencies.
- Used:
  - as a design technique "bottom up design", and
  - as a way of validating table structures produced via "top down design" (ER modelling)

The Normalisation Process Goals

- Creating valid relations, i.e. each relation meets the properties of the relational model. In particular:
  - Entity integrity
  - Referential integrity
  - No many-to-many relationship
  - Each cell contains a single value (is atomic).
- In practical terms:
  - Each table represents a single subject
  - No data item will be unnecessarily stored in more than one table.
  - The relationship between tables can be established (pair of PK and FK is identified).
  - Each table is void of insert, update and delete anomalies.



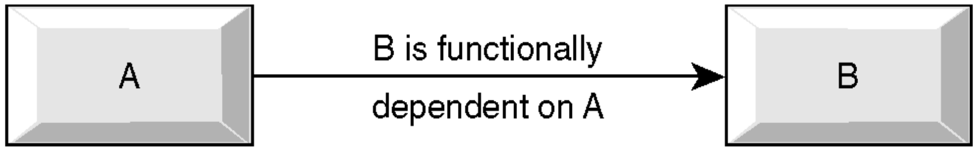
CUSTOMER ORDER

Order Number:	61384		**Order Date:** 12/3/2017
Customer Number:	1273		
Customer Name:	Computer Training Centre		
Customer Address:	123 Excellent St Monash, Vic, 3000		

PART NUMBER	DESCRIPTION	QTY ORDERED	LINE PRICE
M128	Bookcase	4	800
B381	TV Cabinet	2	600
R210	Round Table	3	1500

ORDER ( )

Functional Dependency Revisited



- An attribute B is FUNCTIONALLY DEPENDENT on another attribute A, if a value of A determines a single value of B at any one time.
  - A ➔ B
  - EMP# ➔ EMP_NAME
  - CUSTNUMB ➔ CUSTNAME
  - ORDER-NUMBER ➔ ORDER-DATE
    - ORDER-NUMBER - independent variable, also know as the DETERMINANT
    - ORDER-DATE - dependent variable
- TOTAL DEPENDENCY
  - attribute A determines B AND attribute B determines A
    - EMPLOYEE-NUMBER ➔ TAX-FILE-NUMBER
    - TAX-FILE-NUMBER ➔ EMPLOYEE-NUMBER

Functional Dependency

- For a ***composite*** PRIMARY KEY, it is possible to have FULL or PARTIAL dependency.
- FULL DEPENDENCY
  – occurs when an attribute is always dependent on all attributes in the composite PK
  – ORDER-NUMBER, PART-NUMBER ➔ QTY-ORDERED
- Lack of full dependency for multiple attribute key = PARTIAL DEPENDENCY
  – ORDER-NUMBER, PART-NUMBER
  ➔ QTY-ORDERED, PART-DESCRIPTION
  – here although qty-ordered is ***fully dependent*** on order-number and part-number, *only* part-number is required to determine part-description
  – part-description is said to be ***partially dependent*** on order-number and part-number

Functional Dependency

- TRANSITIVE DEPENDENCY
  - occurs when Y depends on X, and Z depends on Y - thus Z also depends on X ie. X ➔ Y ➔ Z
  - **and** Y is not a candidate key (or part of a candidate key)
  - ORDER-NUMB ➔ CUSTOMER-NUMB ➔ CUSTOMER-NAME
- Dependencies are depicted with the help of a **Dependency Diagram**.
- Normalisation converts a relation into relations of progressively smaller number of attributes and tuples until an optimum level of decomposition is reached - little or no data redundancy exists.
- The output from normalisation is a set of relations that meet all conditions set in the relational model principles.

First Normal Form

- FIRST NORMAL FORM (part of formal definition of a relation)
  - A RELATION IS IN FIRST NORMAL FORM (1NF) IF:
    - a unique primary key has been identified for each tuple/row.
    - it is a valid relation
      - Entity integrity (no part of PK is null)
      - Single value for each cell.
      - No repeating group.
    - all attributes are functionally dependent on all or part of the primary key

Unormalised Form (UNF)

- **Identify a “subject” that needs to be modelled**
  - For example from figure 6.1 possible “subjects” of interest:
    - PROJECT (we will call this representation 1)
    - EMPLOYEE_PROJECT_ASSIGNMENT (we will abbreviate this as ASSIGNMENT and we call this representation 2).
- Choose one subject of interest as a starting point and identify a primary key for this subject of interest.
  - For example for PROJECT, primary key would be project _number (or we will abbreviate it as proj_num).

UNF to 1NF transformation

- Identify the repeating group(s), if any, in the unnormalised relation.
  - For representation 1, a project will have more than one employee assigned to it, hence there is a repeating group.
  - We have one-to-many relationship from PROJECT to EMPLOYEE.

UNF to 1NF

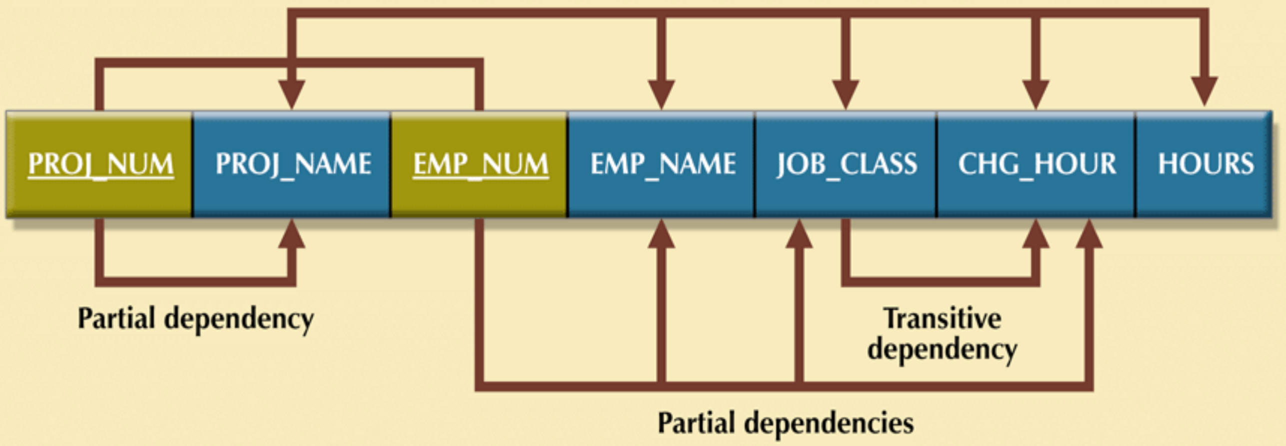
- Move from UNF to 1NF by:
  1. identify a unique identifier for the repeating group.
  2. *remove the repeating group* along with the PK of the main relation.
  3. The PK of the new relation resulting from the removal of repeating group will *normally* have a composite PK made up of the PK of the main relation and the unique identifier chosen in 1. above, but this ***must be checked***.

1NF to 2NF

- A RELATION IS IN 2NF IF -
  - all non key attributes are functionally dependent on the **entire** key (simplified definition)
    - i.e. no partial dependencies exist
  - all non key attributes are functionally dependent on **any candidate key** (general definition)
  - often, *but not always*, simplified vs general definition result in the same decomposition.

Representation 2: Dependency Diagram (1NF)

FIGURE 6.3 First normal form (1NF) dependency diagram



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

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

TRANSITIVE DEPENDENCY:
(JOB CLASS ➡ CHG_HOUR)

MONASH University

2NF Conversion Results (Representations 1 & 2)

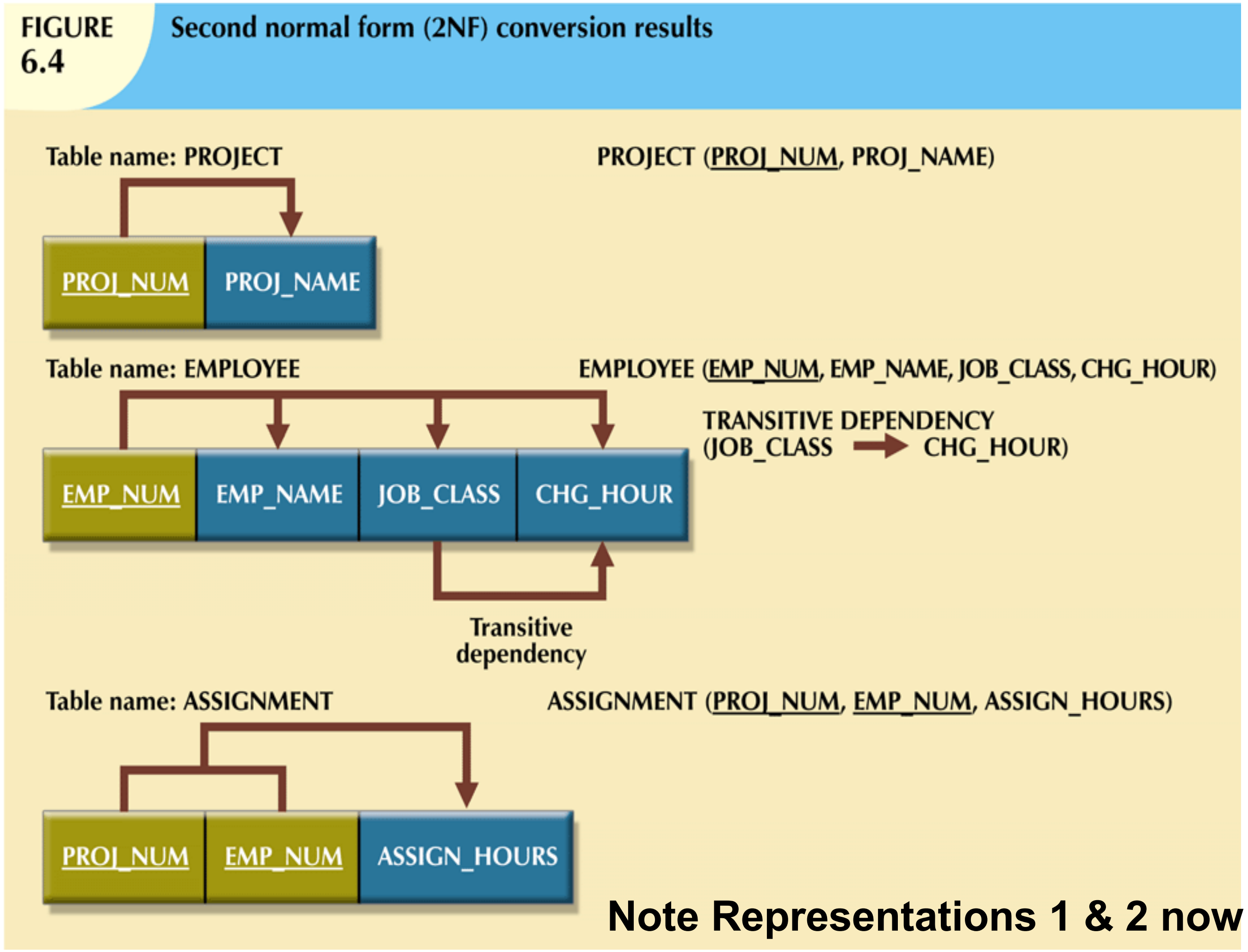


FIGURE 6.4 Second normal form (2NF) conversion results

Note Representations 1 & 2 now equivalent

2NF to 3NF

- A RELATION IS IN 3NF IF -
  - all transitive dependencies have been removed - check for ***non key attribute dependent on another non key attribute***
- Move from 2NF to 3NF by removing transitive dependencies

Relations in 3NF

FIGURE 6.5 Third normal form (3NF) conversion results

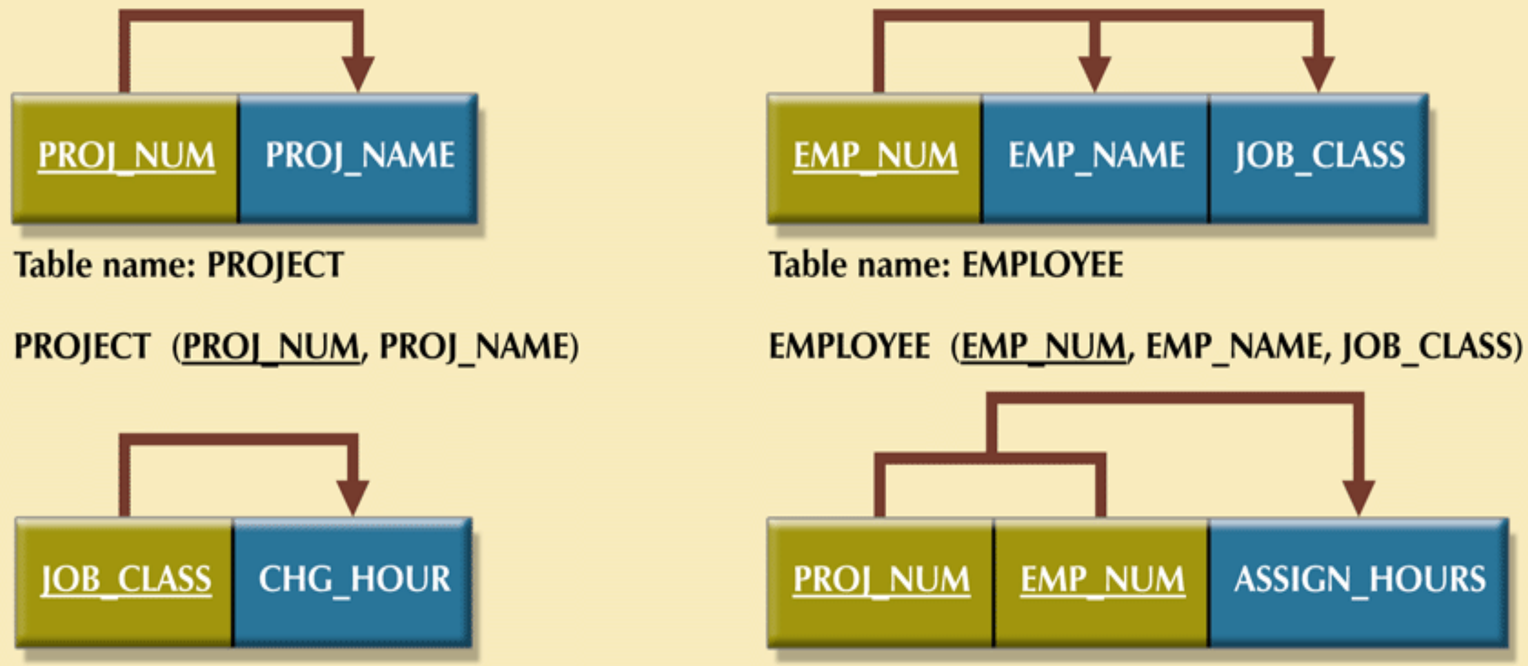


Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)

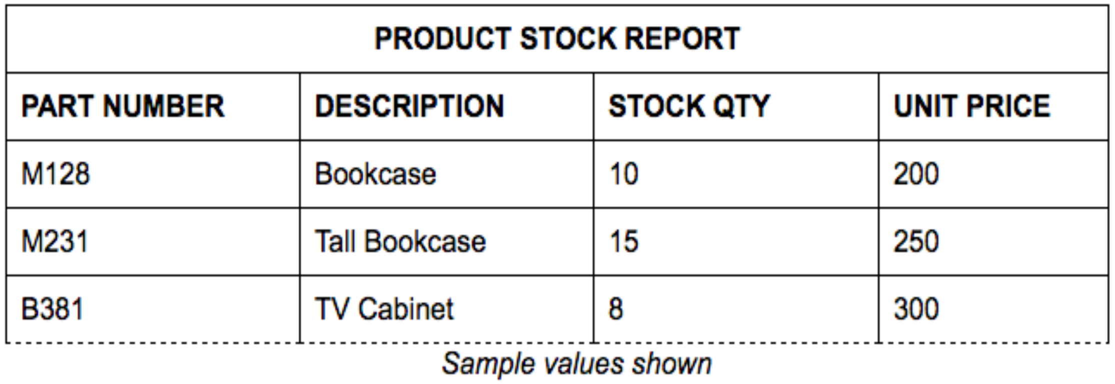
Table name: JOB

JOB (JOB_CLASS, CHG_HOUR)

Table name: ASSIGNMENT

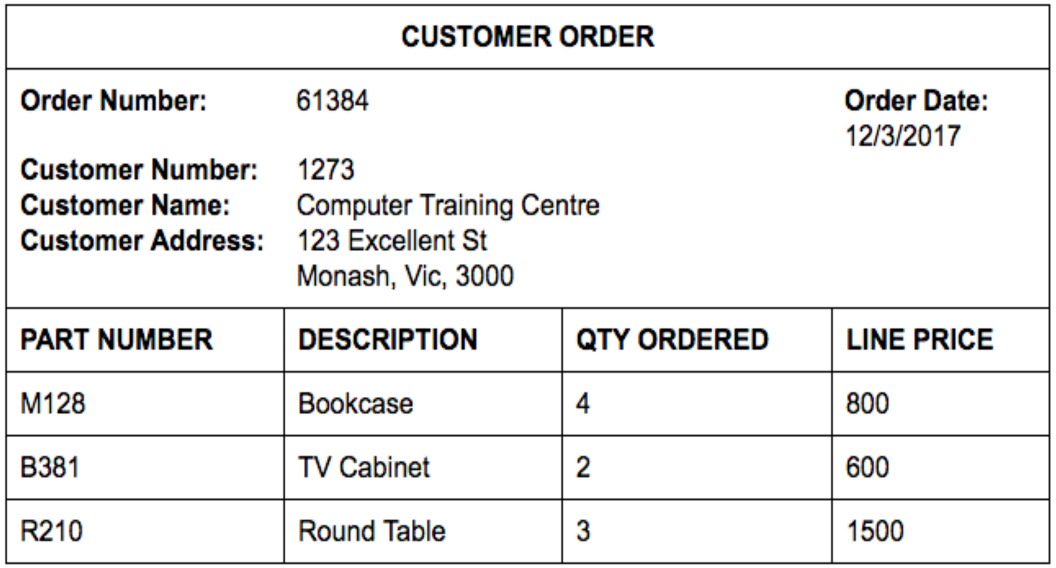
ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)

NORMALISE MULTIPLE FORMS



PRODUCT STOCK REPORT			
PART NUMBER	**DESCRIPTION**	**STOCK QTY**	**UNIT PRICE**
M128	Bookcase	10	200
M231	Tall Bookcase	15	250
B381	TV Cabinet	8	300

Sample values shown



CUSTOMER ORDER

Order Number: 61384 **Order Date:** 12/3/2017

Customer Number: 1273
Customer Name: Computer Training Centre
Customer Address: 123 Excellent St
Monash, Vic, 3000

PART NUMBER	DESCRIPTION	QTY ORDERED	LINE PRICE
M128	Bookcase	4	800
B381	TV Cabinet	2	600
R210	Round Table	3	1500

NORMALISE MULTIPLE FORMS PROCESS

- Take each form one by one
  - Represent as UNF
  - Normalise through to 3NF, show ***all*** stages (1NF, 2NF, 3NF), include dependency diagrams
- Collect all 3NF relations from all forms as one list
- Combine relations within this list with the same PK, since these represent the same object class

Summary

- Things to remember
  - Primary Key selection in moving from UNF to 1NF is important, it will determine the starting point (choose your subject of interest).
  - Functional dependency
  - Process of removing attributes in relations based on the concept of 1NF, 2NF and 3NF.
    - UNF to 1NF define PK & remove repeating group.
    - 1NF to 2NF remove partial dependency.
    - 2NF to 3NF remove transitive dependency.