COMM1822

Term 2 2022

Introduction to Databases for Business Analytics

Week 5 Normalisation Part 2

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

WARNING

This material has been reproduced and communicated to you by or on behalf of the University of New South Wales in accordance with section 113P(1) of the Copyright Act 1968 (Act). The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice

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 work to these websites, and if another student downloads and presents it as their own either wholly or partially, **you might be found guilty of collusion — even years after graduation**.

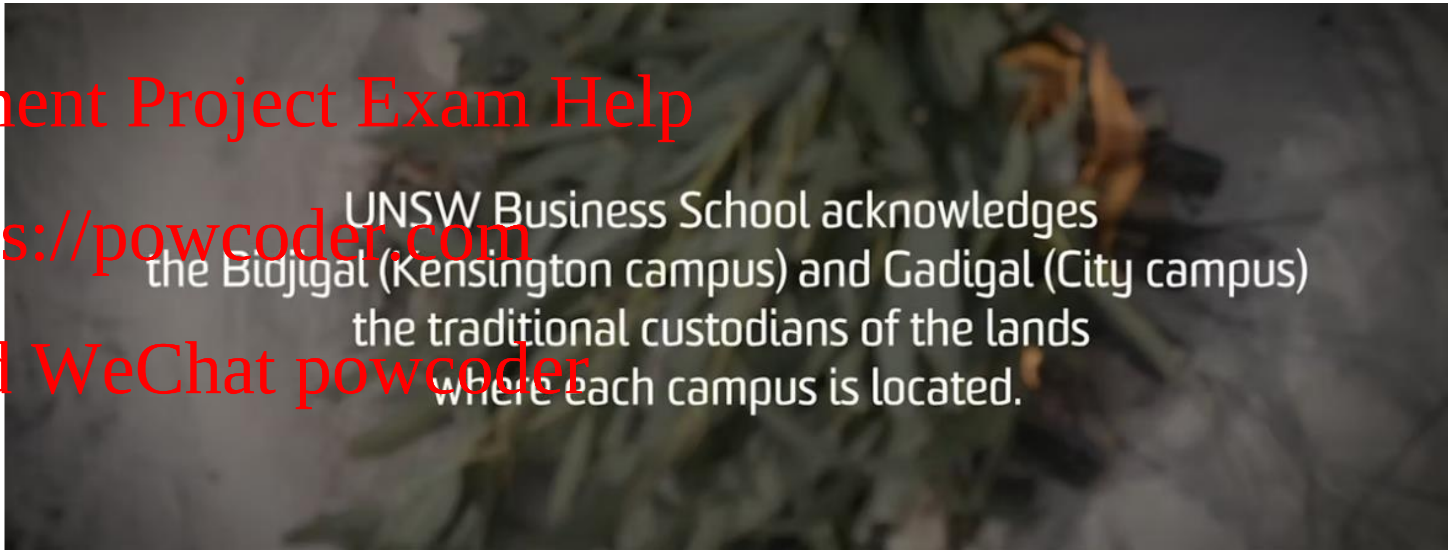
- These file-sharing websites may also accept purchase of course materials, **such as copies of lecture slides and tutorial handouts**. **By law, the copyright on course materials, developed by UNSW staff in the course of their employment, belongs to UNSW. It constitutes copyright infringement, if not 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 Strait Islander Elders, past and present and their 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 acknowledges the Bidjigal (Kensington campus) and Gadigal (City campus) the traditional custodians of the lands where each campus is located.

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

Agenda

- ❑ Review normalisation and its role in the database design process
- ❑ Identify and describe each of the normal forms: 1NF, 2NF, 3NF, and BCNF.
- ❑ Explain how normal forms can be transformed from lower normal forms to higher normal forms (and vice versa – denormalisation)
- ❑ Apply normalisation rules to evaluate and correct table structures
- ❑ Identify situations that require denormalisation to generate information efficiently

Review (Normal Forms)

Normal Form	Characteristic
First normal form (1NF)	PK identified and no repeating groups
Second normal form (2NF)	1NF and no partial dependencies
Third normal form (3NF)	2NF and no transitive dependencies
Boyce-Codd NF (BCNF)	Every determinant is a candidate key

- *Create a valid primary key and resolve multi-valued attributes.*
  - **First Normal Form (1NF)**
- *Draw partial functional dependency diagrams and resolve them.*
  - ***Second Normal Form (2NF)***
- *Draw transitive functional dependency diagrams and resolve them.*
  - **Third Normal Form (3NF)**
- *Resolve cases where non-key attributes determine primary key attributes*. (Special case of 3NF)
  - **Boyce-Codd Normal Form (BCNF)**

Review (Functional Dependencies)

- **Functional Dependencies**
  - Inclusion (or reflexive) rule
  - Augmentation rule
  - Transitivity rule, …
- **Partial dependency**: functional dependence in which the determinant is only part of the primary key
  - Assumption: one candidate key
  - Straight forward
  - Easy to identify
- **Transitive dependency**: attribute is dependent on another attribute 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 nonprime attributes

Inference Rules for Functional Dependencies (FDs) - Armstrong’s Axioms Primary Rules

A → B: Attribute B “functionally depends” on an attribute A; or

Attribute A determines attribute B; or

“If I know the value of A, then I know the value of B”.

1) **Inclusion (Reflexive) rule: if $Y \subseteq X$, then $X \rightarrow Y$** (⊆: the notation of subset)
“If zID ⊆ {zID, LastName}, then {zID, LastName} → zID” (“If zID is a part of the attribute set {zID, LastName}, then {zID, LastName} determines zID”)

2) **Augmentation rule: if $X \rightarrow Y$, then $\{W, X\} \rightarrow \{W, Y\}$**
“If zID → LastName, then {zID, FirstName} → {LastName, FirstName}”

3) **Transitivity rule: if $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$**
“If zID → MobileNumber and MobileNumber → LastName, then zID → LastName”.

Union
If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow \{Y, Z\}$.

Proof:

$X \rightarrow Y$ … (1) (Given)

$\{X, Z\} \rightarrow \{Y, Z\}$ … (2) (Augmentation of (1) & Z)

$X \rightarrow Z$ … (3) (Given)

$X \rightarrow \{X, Z\}$ … (4) (Augmentation of (3) & X)

$X \rightarrow \{Y, Z\}$ … (5) (Transitivity of (4) and (2))

Armstrong's Axioms Primary Rules

i. Inclusion (Reflexive) rule: If $Y \subseteq X$, then $X \rightarrow Y$.
ii. Augmentation rule: If $X \rightarrow Y$, then $\{W, X\} \rightarrow \{W, Y\}$.
iii. Transitivity rule: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

Pseudo-Transitivity
If $X \rightarrow Y$ and $\{Y, Z\} \rightarrow W$, then $\{X, Z\} \rightarrow W$.

Proof:

$X \rightarrow Y$ … (1) (Given)

$\{X, Z\} \rightarrow \{Y, Z\}$ … (2) (Augmentation of (1) & Z)

$\{Y, Z\} \rightarrow W$ … (3) (Given)

$\{X, Z\} \rightarrow W$ … (4) (Transitivity of (2) and (3))

Armstrong's Axioms Primary Rules

i. Inclusion (Reflexive) rule: If $Y \subseteq X$, then $X \rightarrow Y$.
ii. Augmentation rule: If $X \rightarrow Y$, then $\{W, X\} \rightarrow \{W, Y\}$.
iii. Transitivity rule: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

Review – Denormalisation

❑ Structural point of view of normal forms

- Higher normal forms are better than lower normal forms

❑ Denormalisation: produces a lower normal form

- Results in increased performance and greater data redundancy

Demonstration of Normalisation (Exercises)

Source: weightwatchers.com

Demonstration (Exercise 0)

We are supposed to create **1NF, 2NF and 3NF** as well as to create an **ER diagram** from this table. To do this, we need to draw **functional, partial and transitive dependency diagrams.**

Attribute	Value
Emp#	1003
Emp. Name	Simpson
Education	BE, MSc, PhD
Dep. Code	SISTM
Dep. Name	Information Systems
Dep. Mgmt	Mr. Black
Job Class	SL-4
Title	Senior Lecturer
Dependents	Marge (wife), Bart (son), Lisa (daughter)
DOB	1/1/1960
Hire Date	10/4/1990
Training	Level-1, Level-2
Base Salary	$85,000

Handling Multi-Valued Attributes

Problem 1: the table has several **multi-valued attributes** and some attributes are not atomic...

Attribute	Value
Emp#	1003
Emp. Name	Simpson
Education	**BE, MSc, PhD**
Dep. Code	SISTM
Dep. Name	Information Systems
Dep. Mgmt	Mr. Black
Job Class	SL-4
Title	Senior Lecturer
Dependents	**Marge (wife), Bart (son), Lisa (daughter)**
DOB	1/1/1960
Hire Date	10/4/1990
Training	**Level-1, Level-2**
Base Salary	$85,000

Handling Multi-Valued Attributes

Multi-valued

R (**Emp#**, Emp. Name, Education, Dep. Code, Dep. Name, Dep. Mgmt., Job Class, Title, Dependents, DOB, Hire Date, Training, Base Salary)

Multi-valued

Multi-valued

Dependency diagrams cannot handle multi-valued attributes.

Handling Multi-Valued Attributes

We **split the multivalued attributes apart**, using our ER/conceptual modelling knowledge. We **replace with appropriate single-value attributes.**

- **Delete** Education ➔ **Add** Education ID (Edu#), Education Description, Graduate Date
- **Delete** Dependents ➔ **Add** Dependent ID (Depd#), Dependent Name
- **Delete** Training ➔ **Add** Training ID (Train#), Training Description, Certificate Date

❑ For the case of Job Class, we associate with other variables, we can change to:

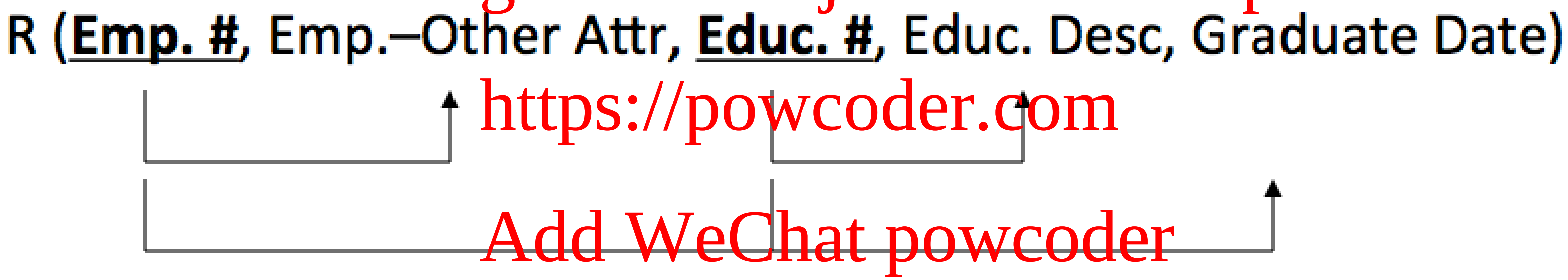
- **Delete** Job Class ➔ **Add** Job ID (Job#), Title, Base Salary

R (**Emp#,** Employee Name, DOB, Hire Date, **Edu#**, Education Desc., Graduate Date, **Dept. Code**, Dept. Name, Dept. Mgmt., **Job#**, Title, Base Salary, **Depd#**, Dependent Name, **Train#,** Training Desc. Certification Date)

Bottom-Up Approach

Start with existing data structure/tables > then try to derive the 3NF from there.

Identify the candidate keys – from there you can identify the PKs



R (**Emp. #**, Emp.–Other Attr, **Educ. #**, Educ. Desc, Graduate Date)

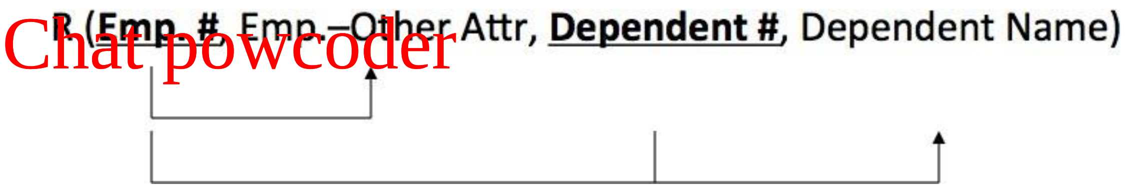
(Hint: this is what we do here)

- ❑ You can see Emp# and Educ# could be candidate keys. Other Employee attributes associate with Emp#.
- ❑ Likewise, a few attributes are associated with Educ#.

Normalisation: Weak Entity

There seems there is a **“weak entity”** in the table.

Attribute	Value
Emp#	1003
Emp. Name	Simpson
Education	BE, MSc, PhD
Dep. Code	SISTM
Dep. Name	Information Systems
Dep. Mgmt	Mr. Black
Job Class	SL-4
Title	Senior Lecturer
Dependents	**Marge (wife), Bart (son), Lisa (daughter)**
DOB	1/1/1960
Hire Date	10/4/1990
Training	Level-1, Level-2
Base Salary	$85,000



R (**Emp #**, Emp –Other Attr, **Dependent #**, Dependent Name)

1NF

Original R:

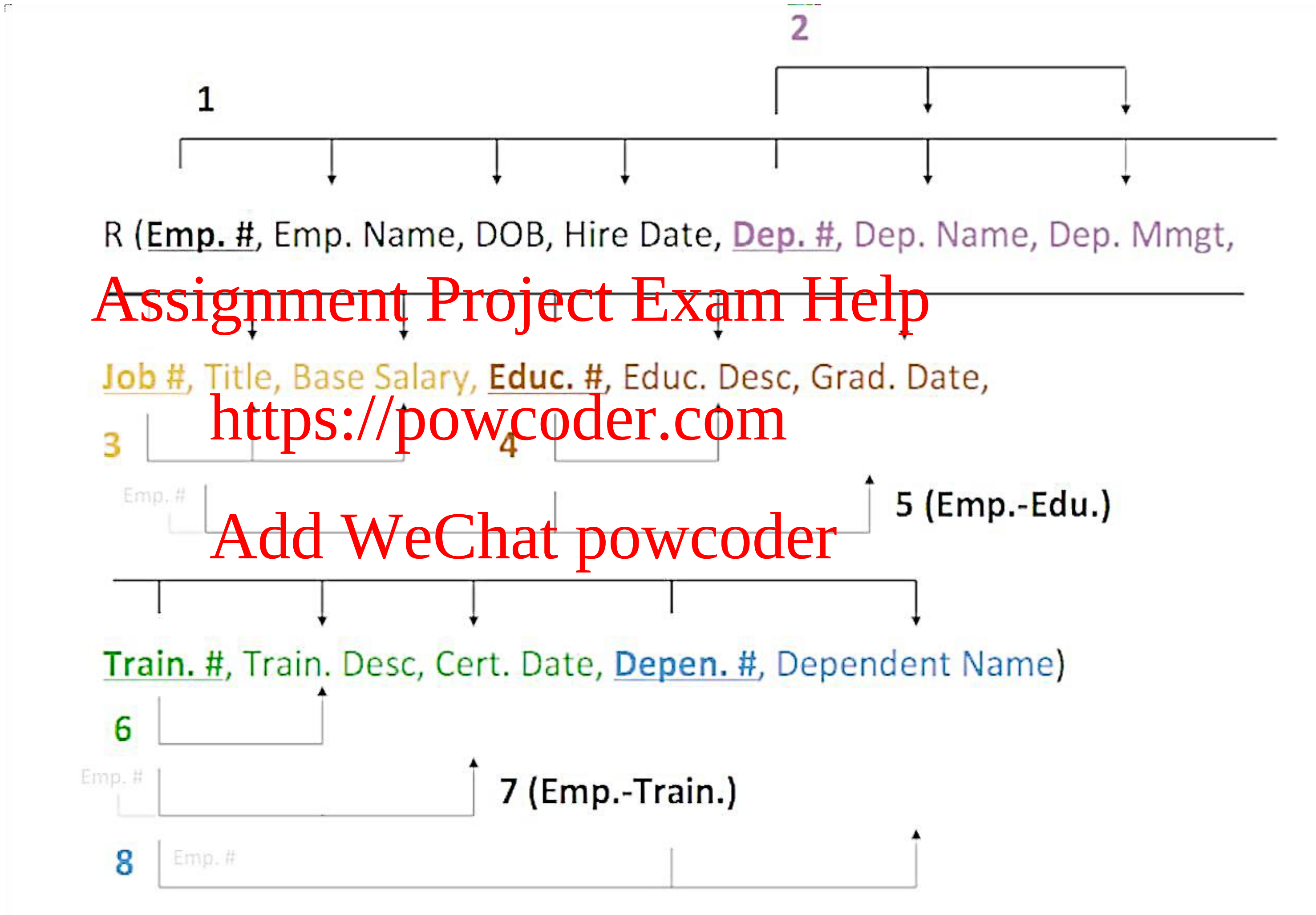
R (Emp#, Emp. Name, Education, Dep. Code, Dep. Name, Dep. Mgmt., Job Class, Title, Dependents, DOB, Hire Date, Training, Base Salary)

Modified R:

After all the changes, now the updated R is 1NF because: **no multivalued attributes** + **valid primary key**.

R (**Emp#,** Emp. Name, DOB, Hire Date, **Edu#**, Education Desc., Graduate Date, **Dept#**, Dept Name, Dept Mgmt., **Job#**, Title, Base Salary, **Depd#**, Depd. Name, **Train#,** Training Desc., Certification Date)

From 1NF to 2NF Via Dependency Diagrams



2

1

R (Emp. #, Emp. Name, DOB, Hire Date, Dep. #, Dep. Name, Dep. Mmgt,

Job #, Title, Base Salary, Educ. #, Educ. Desc, Grad. Date,

3

4

Emp. #

5 (Emp.-Edu.)

Train. #, Train. Desc, Cert. Date, Depen. #, Dependent Name)

6

Emp. #

7 (Emp.-Train.)

8

Emp. #

Using Dependency Diagrams

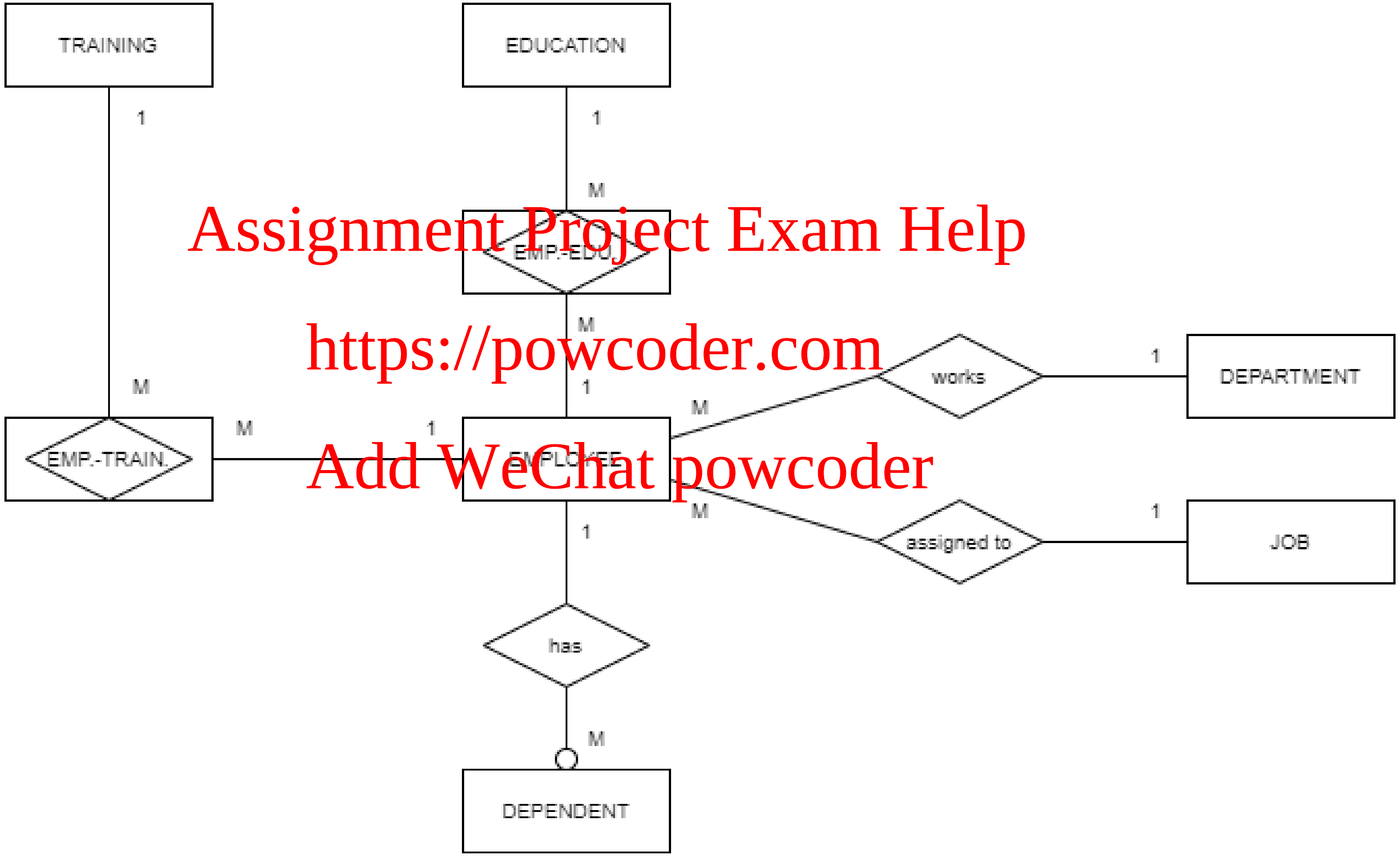
1) Emp# + Dep.# + Job# + Edu# + Train# + Depd# > primary key functional dependency > **OK/no action**
2) Dep.# is the key for Dep.Name and Dep.Mgmt (partial func. dependency) > **new relation Department required**
3) Job# is the key for Title, Base Salary (partial func. dependency) > **new relation Job required**
4) Edu# is the key for Educ.Desc. (partial func. dependency) > **new relation Education required**
5) Emp# AND Edu# are the keys for Grad.Date (partial func. dependency) > **new relation Emp.-Edu. required (Composite/Bridge entity)**
6) Train# is the key for Train. Desc. (partial func. dependency) > **new relation Training required**
7) Emp# AND Train# are the keys for Cert.Date (partial func. dependency) > **new relation Emp.-Train. required (Composite/Bridge entity)**
8) Emp# AND Depd# are the key for Depn.Name (partial func. dependency) > **new relation Dependent required (weak entity)**

2NF / 3NF

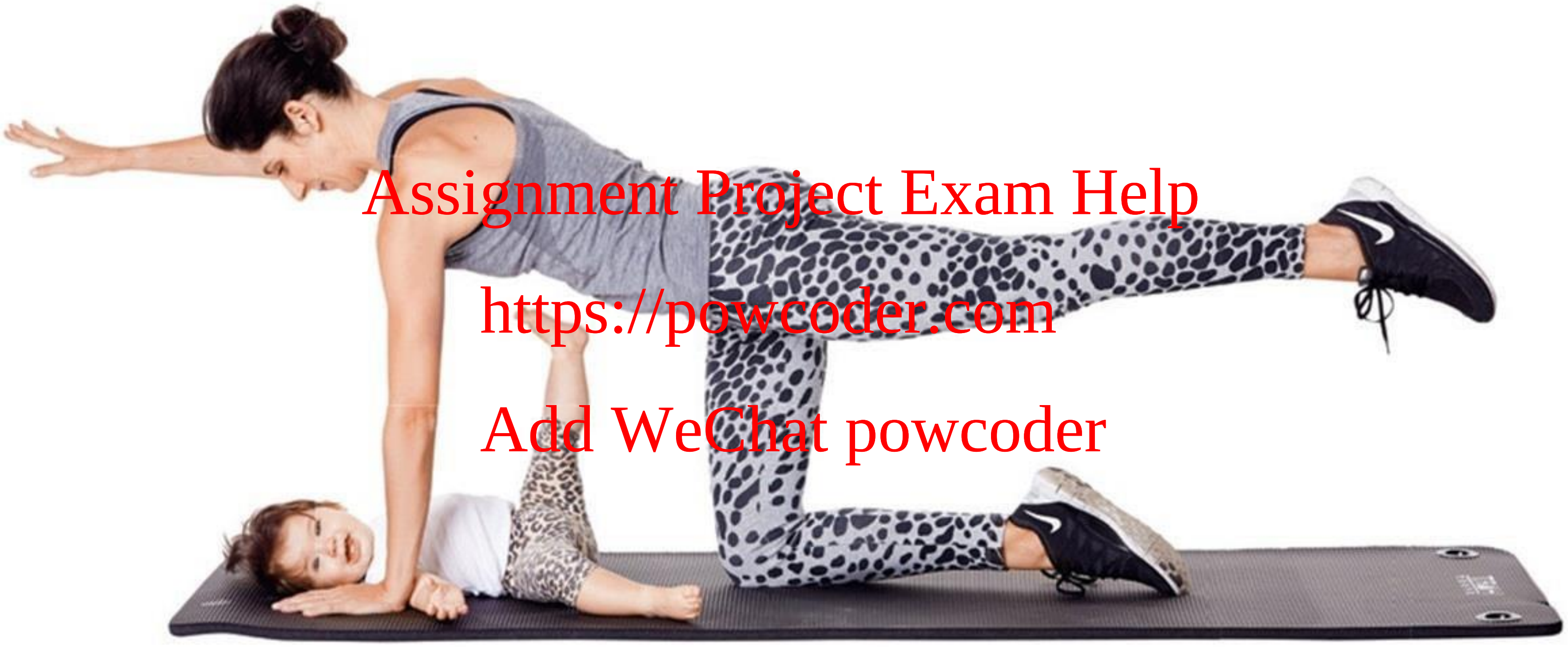
1. Employee (**Emp#**, Emp. Name, DOB, Hire Date, Dept. Code, Job#)
2. Department (**Dept.#**, Dept Name, Dept Mgmt)
3. Job (**Job #**, Title, Base Salary)
4. Education (**Edu#**, Edu. Desc.)
5. Emp.-Edu. (**Emp#, Edu#**, Grad. Date)
6. Training (**Train#**, Train. Desc.)
7. Emp.-Train. (**Emp#**, **Train#**, Cert. Date)
8. Dependent (**Emp#, Depd#**, Dependent Name)

No transitive dependencies → 3NF

ER Diagram


TRAINING
EDUCATION
1
1
M
EMP.-EDU.
M
works
1
DEPARTMENT
M
1
M
EMP.-TRAIN.
M
1
EMPLOYEE
M
1
assigned to
1
JOB
has
M
DEPARTMENT
DEPENDENT

Normalisation Exercises



Source: weightwatchers.com

Exercise 1

Consider the following **relational schema** R (A, B, C, D, E, F, G, H) and the following **functional dependencies**:

- A, B → C, D, E, F, G, H
- A → C, D, G, H
- B → E
- G → H

□ Create functional dependency, partial dependency and transitive dependency diagrams.

□ Normalise to 1NF, 2NF and 3NF.

□ Draw the ER diagram from the 3NF (Optional)

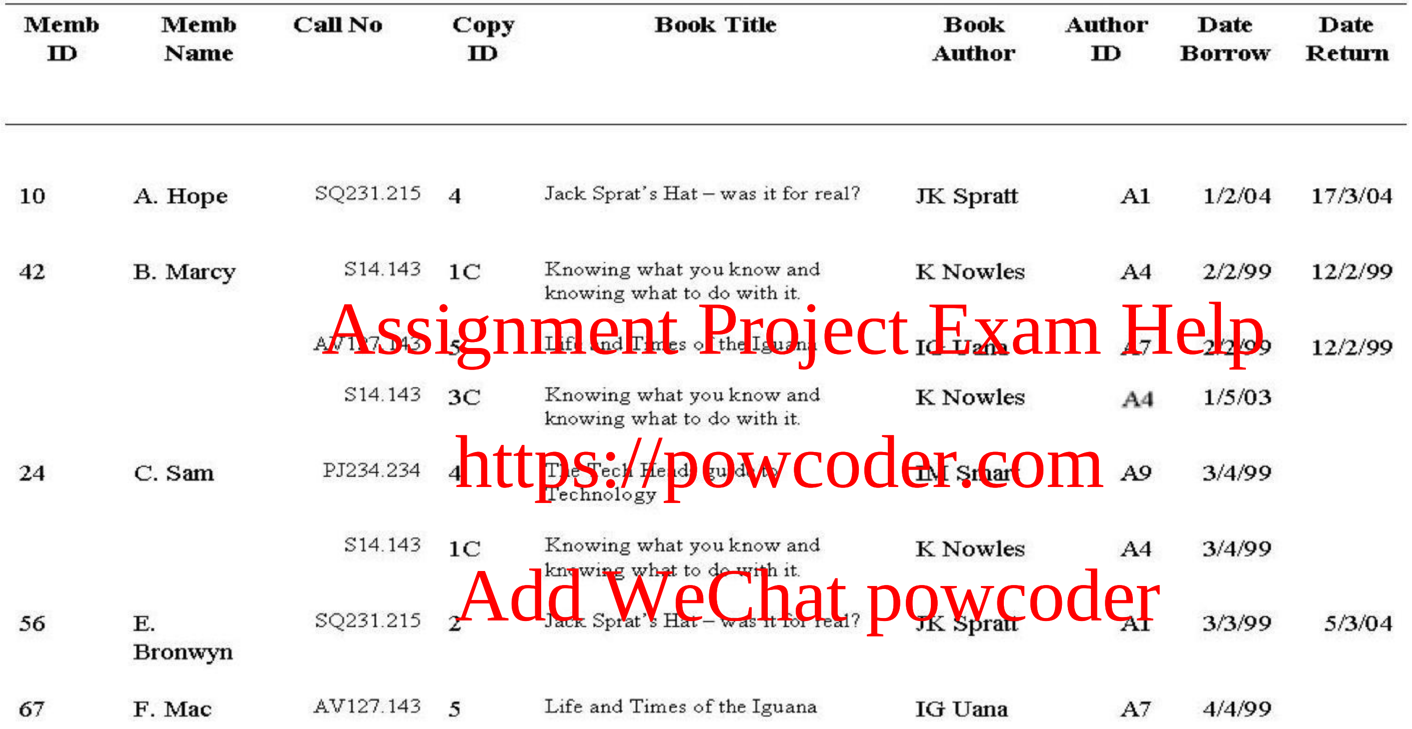
Exercise 2

For the following relations:

- ❑ Indicate the normal form (1NF, 2NF or 3NF) for that relation.
- ❑ Decompose into the 3NF (if not already in 3NF)
- ❑ Note: Functional dependencies (FDs) – other than those implied by the primary keys (PKs) – are shown.

1) CLASS (Course_No, Section_No)
2) CLASS (Course_No, Section_No, Room)
3) CLASS (Course_No, Section_No, Room, Capacity),
   with FD: Room ➔ Capacity
4) CLASS (Course_No, Section_No, Course_Name, Room, Capacity),
   with FDs: Course_No ➔ Course_Name; Room ➔ Capacity

Exercise 3



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 Head's guide to Technology	JM Stuart	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	

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.

Solution to Exercise 3

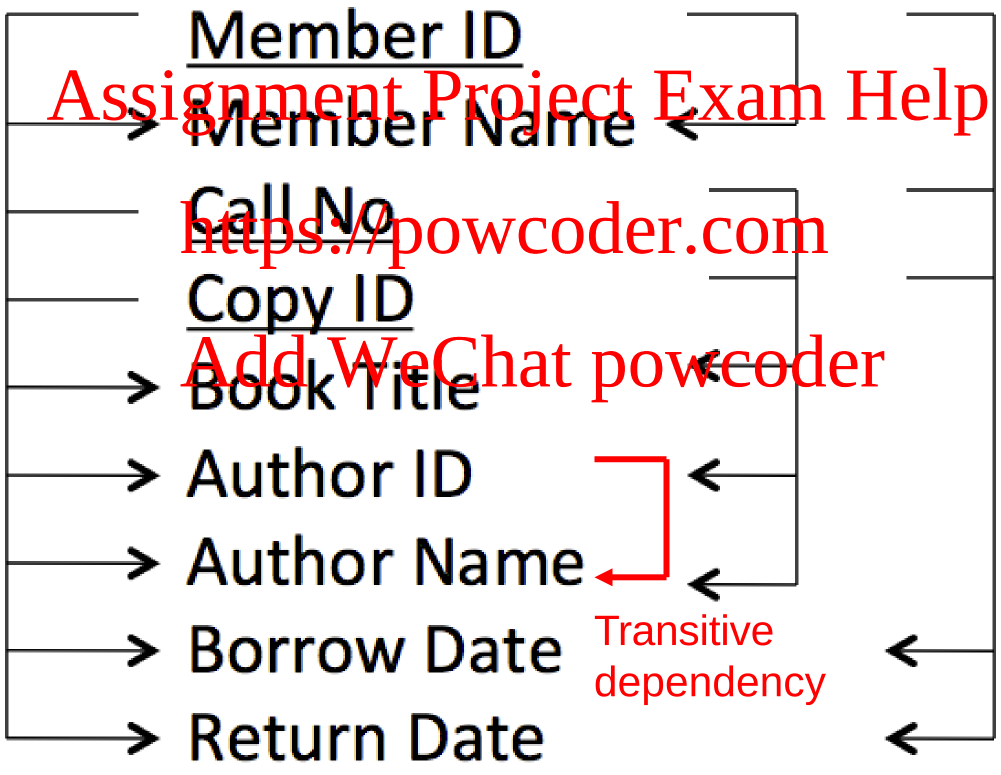
1. Potential problems with the table are:

- Not in 1NF, hence cannot be used in relational DBMS.
- PK not completely defined, could let to identical tuples.
- Order of rows matters (cannot be sorted in different order).
- Has redundant data.
- Invites inconsistencies/anomalies…

Solution to Exercise 3

2. PK(s) and dependencies diagrams

1NF

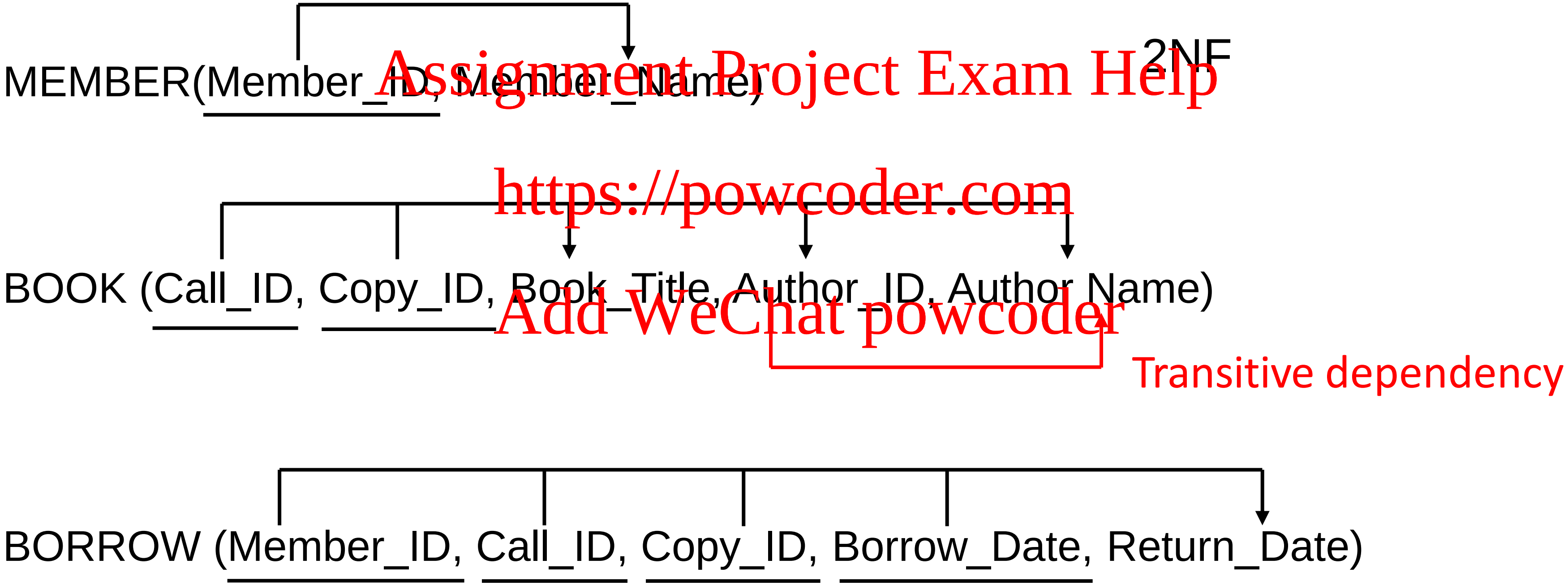


Functional dependency diagram

Partial dependency

Solution to Exercise 3

3. Normalise it to the 3NF (Step 1)



MEMBER(Member_ID, Member_Name)

2NF

BOOK (Call_ID, Copy_ID, Book_Title, Author_ID, Author_Name)

Transitive dependency

BORROW (Member_ID, Call_ID, Copy_ID, Borrow_Date, Return_Date)

Note: Borrow Date should be modelled as part of PK to handle multiple borrowing for the same book by the same member.

Solution to Exercise 3

3. Normalise it to the 3NF (Step 2)

MEMBER(Member_ID, Member_Name)

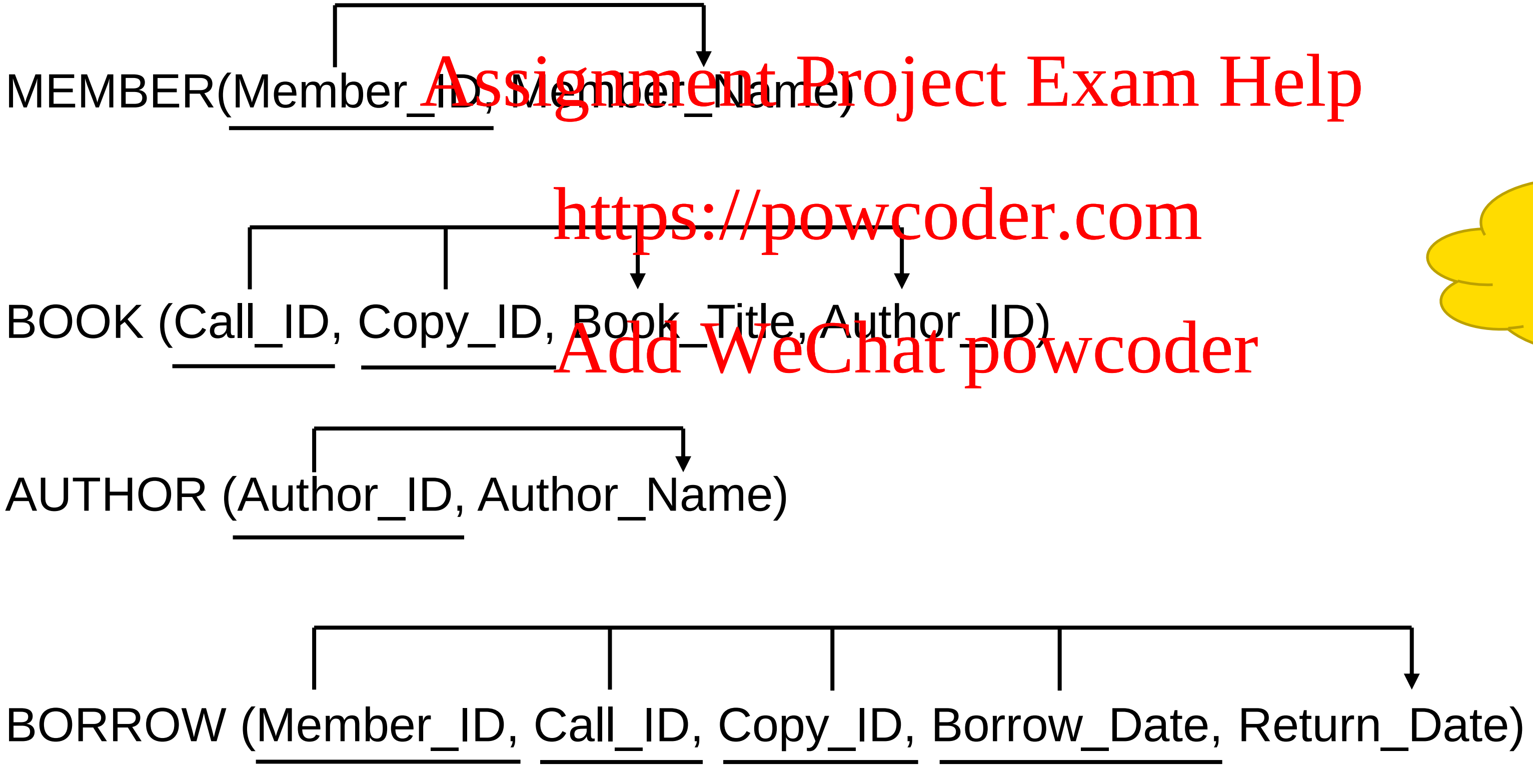
BOOK (Call_ID, Copy_ID, Book_Title, Author_ID)

AUTHOR (Author_ID, Author_Name)

BORROW (Member_ID, Call_ID, Copy_ID, Borrow_Date, Return_Date)

Solution to Exercise 3

3. Normalise it to the 3NF, and show the dependency diagram for each table



MEMBER(Member_ID, Member_Name)

BOOK (Call_ID, Copy_ID, Book_Title, Author_ID)

AUTHOR (Author_ID, Author_Name)

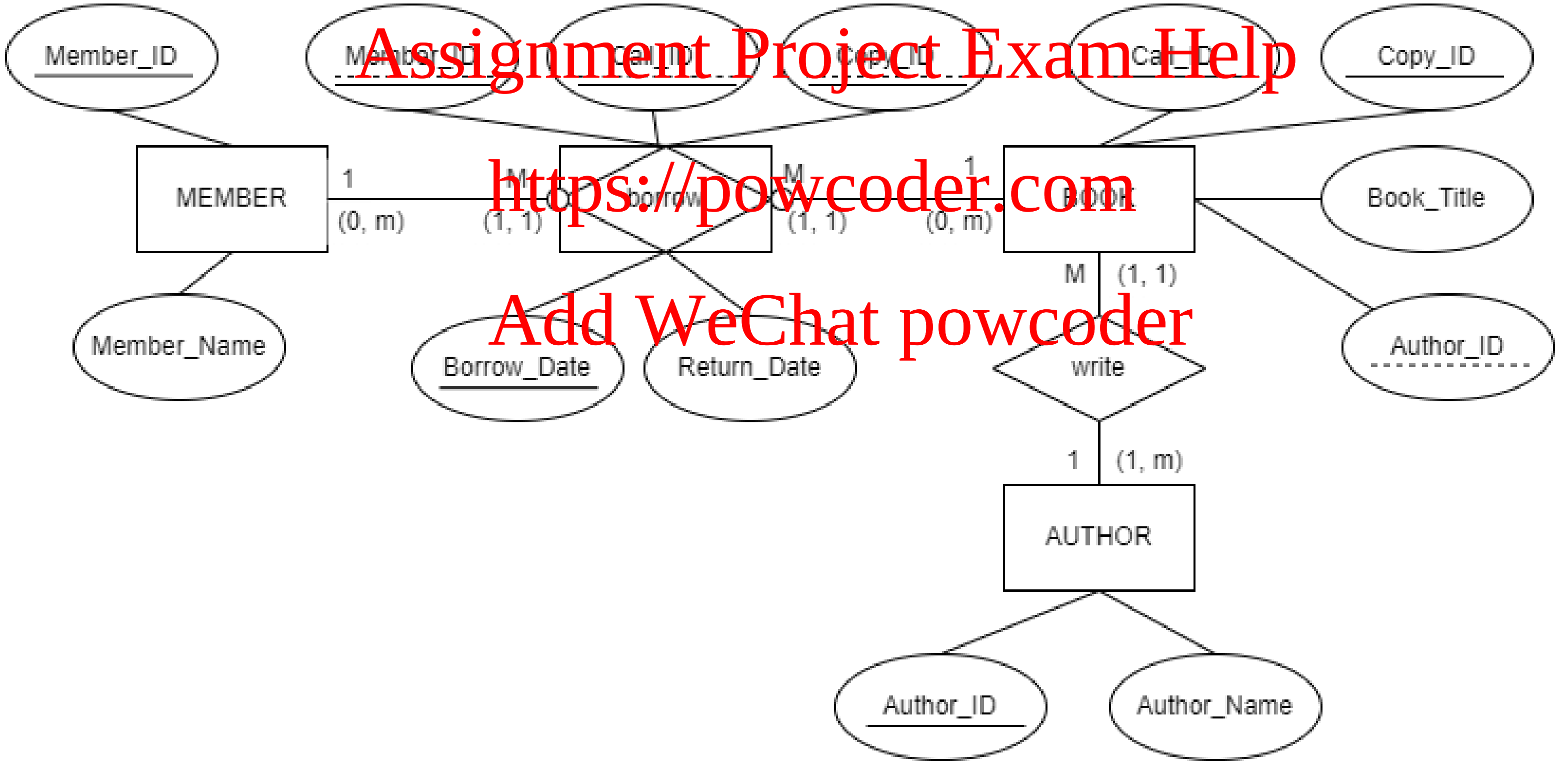
BORROW (Member_ID, Call_ID, Copy_ID, Borrow_Date, Return_Date)

Required in your Assessment

Solution to Exercise 3

Required in your Assessment

4. Draw the ERM (based on the 3NF):



Exercise 4

Joe is the manager of a **dinner club** would like to create a database to email event invitations to the club's members, to plan the meals, to keep track of who attends the dinners etc. He explains the following business rules:

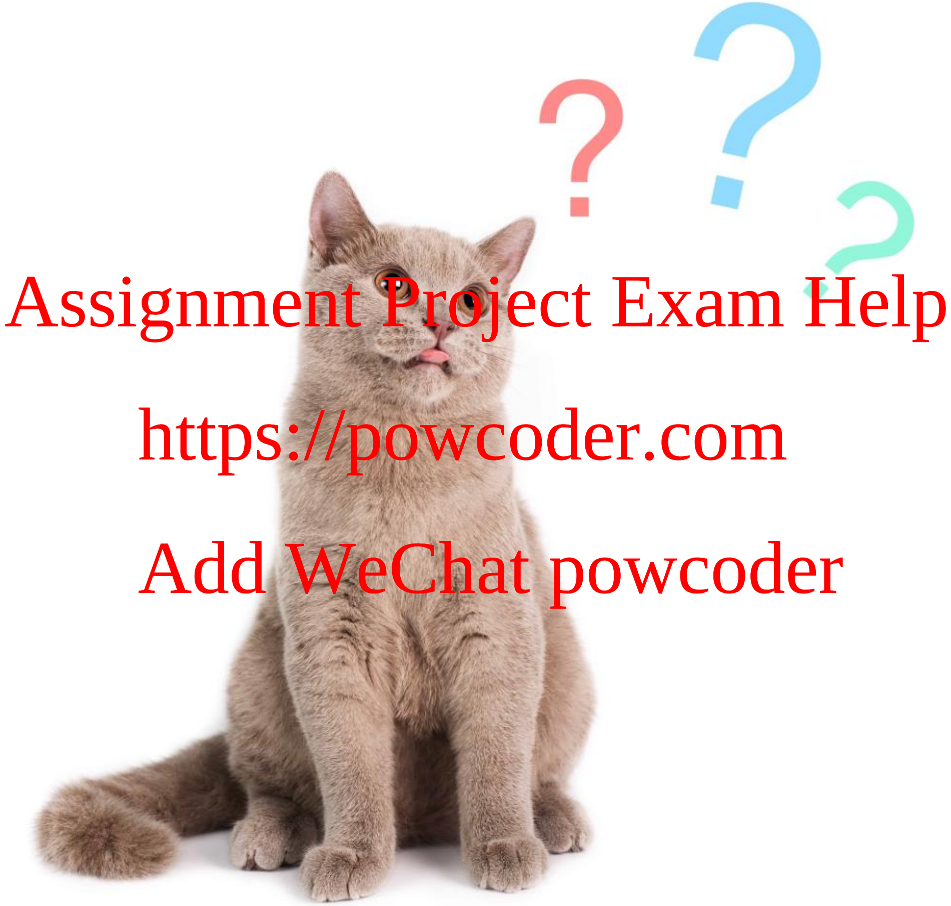
- Each dinner is joined by many members and each member may attend many dinners.
- A member receives many invitations and each invitation is emailed to many members.
- A dinner is based on a single entrée, but an entrée may be used as the basis for many dinners. For example, a dinner may be composed of a fish entrée, mushroom risotto and panna cotta. Or, a dinner may be composed of a fish entrée, wagyu beef and tiramisu. The same goes for the other dishes…

Because the manager is not a database expert, his first attempts at a "database" have resulted in the following, not very functional structure (on the right). Can you help Joe?

1. Draw functional, partial and transitive dependency diagrams.
2. Create the 1NF, the 2NF and the 3NF.
3. Draw the ER diagram from the 3NF.

Attribute Name	**Sample Value**
MEMBER_NUM	214
MEMBER_NAME	Alice B. Van der Voort
MEMBER_ADDR	325 Meadow Park
MEMBER_CITY	Murkywaters
MEMBER_ZIPCODE	12345
INVITE_NUM	8
INVITE_DATE	1/8/12
ACCEPT_DATE	9/8/12
DINNER_DATE	23/8/12
DINNER_ATTEND	Y
DINNER_CODE	5
DINNER_DESCRIPTION	Sea Delight
ENTRÉE_CODE	3
ENTRÉE_DESCRIPTION	Stuffed Crab
DESSERT_CODE	8
DESSERT_DESCRIPTION	Chocolate Mousse

Questions



Source: petcare.com.au