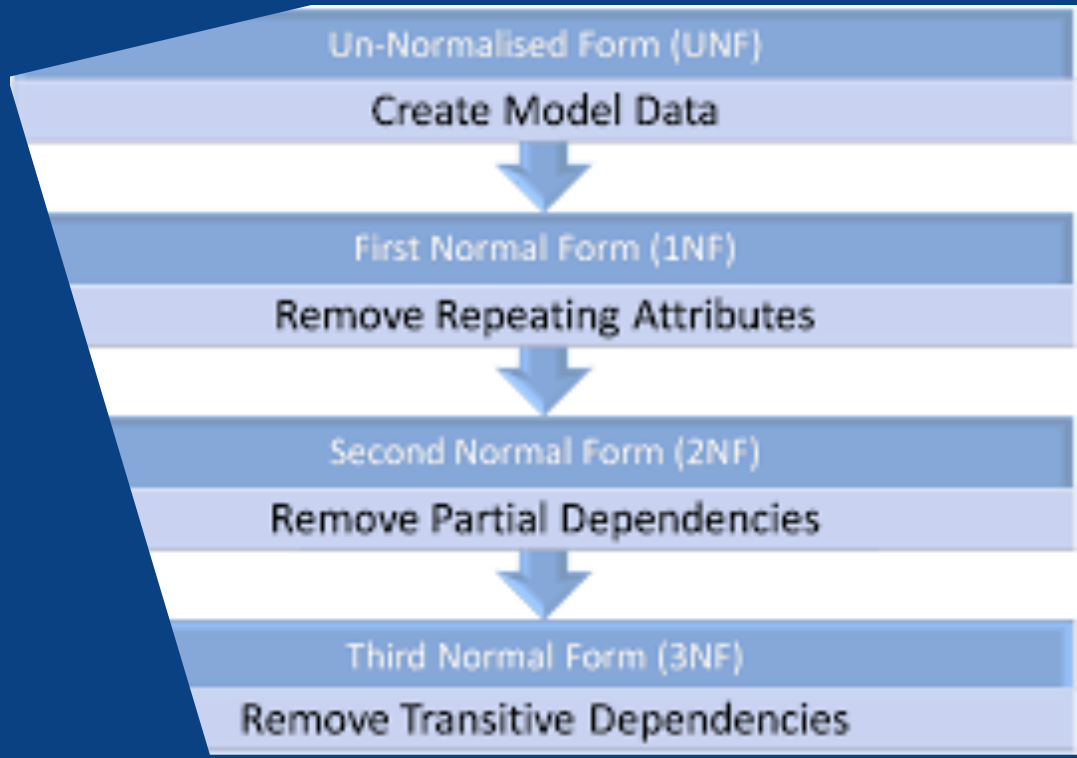
THE UNIVERSITY OF MELBOURNE

Normalisation

Database Systems & Information Modelling
INFO90002



Week 4 – Three normal forms

Dr Tanya Linden
Dr Simon D'Alfonso

Learning Objectives

By the end of this lecture, you should be able to:

- Define normalisation
- Explain and identify database anomalies
- Define and identify functional dependencies
- Normalise relations to:
  - 1st Normal Form (1NF)
  - 2nd Normal Form (2NF)
  - 3rd Normal Form (3NF)

Motivation for normalisation

What happens if we don’t normalise?

What's wrong with the *organisation* of data in this table? duplication is a problem

Student ID#	Student Name	Campus Address	Degree	Phone	Subject ID	Subject Title	Lecturer Name	Lecturer Office	Lecturer Phone	Sem.	Grade
A121	Joy Egbert	166 Grattan Street	B.Com.	555-7771	ACC101	Accounting	Davern	T240C	8344-1846	1-11	H1
A121	Joy Egbert	166 Grattan Street	B.Com.	555-7771	ECO101	Economics	Smyth	T240F	8344-1868	1-11	H2B
A121	Joy Egbert	166 Grattan Street	B.Com.	555-7771	ECO104	Quant. M.	Collier	T240D	8344-5716	1-11	H2B
A121	Joy Egbert	166 Grattan Street	B.Com.	555-7771	FIN101	Finance.	James	T240D	8344-5275	1-11	H2A
A121	Joy Egbert	166 Grattan Street	B.Com.	555-7771	ACC103	Processes	Wise	T240E	8344-5309	1-11	H3
A123	Larry Mueller	302 Royal Parade	B.Com.	555-1235	ACC101	Accounting	Davern	T240C	8344-1846	1-11	H1
A123	Larry Mueller	302 Royal Parade	B.Com.	555-1235	ECO101	Economics	Smyth	T240F	8344-1868	1-11	H2B
A123	Larry Mueller	302 Royal Parade	B.Com.	555-1235	ECO104	Quant. M.	Collier	T240D	8344-5716	1-11	H2A
A123	Larry Mueller	302 Royal Parade	B.Com.	555-1235	FIN101	Finance.	James	T240D	8344-5275	1-11	H3
A124	Mike Guon	224 Swanston St.	B.Eco.	555-2214	ACC101	Accounting	Davern	T240C	8344-1846	1-11	H2A
A124	Mike Guon	224 Swanston St.	B.Eco.	555-2214	ECO101	Economics	Smyth	T240F	8344-1868	1-11	H2A
A124	Mike Guon	224 Swanston St.	B.Eco.	555-2214	ECO104	Quant. M.	Collier	T240D	8344-5716	1-11	H2B
A124	Mike Guon	224 Swanston St.	B.Eco.	555-2214	ACC103	Processes	Wise	T240E	8344-5309	1-11	H2B
A126	Jackie Judson	85 Barry Street	B.Eco.	555-1245	ACC101	Accounting	Davern	T240C	8344-1846	1-11	H1
A126	Jackie Judson	85 Barry Street	B.Eco.	555-1245	ECO101	Economics	Smyth	T240F	8344-1868	1-11	H2B
A126	Jackie Judson	85 Barry Street	B.Eco.	555-1245	ECO104	Quant. M.	Collier	T240D	8344-5716	1-11	H2B
A126	Jackie Judson	85 Barry Street	B.Eco.	555-1245	ACC103	Processes	Wise	T240E	8344-5309	1-11	H2A
...	...	...	...	...	...	...	...	...	...	...	...

“Normalised” Data

Student Table

Student ID#	Student Name	Campus Address	Degree	Phone
A121	Joy Egbert	166 Grattan Street	B.Com.	555-7771
A123	Larry Mueller	302 Royal Parade	B.Com.	555-1235
A124	Mike Guon	224 Swanston St.	B.Eco.	555-2214
A126	Jackie Judson	85 Barry Street	B.Eco.	555-1245
...	...	...	...	...

Subject Table

Subject ID	Subject Title
ACC101	Accounting
ECO101	Economics
ECO104	Quant. M.
FIN101	Finance.
ACC103	Processes
...	...

Subject Coordinator

Subject ID	Sem.	Lecturer Name
ACC101	1-11	Davern
ECO101	1-11	Smyth
ECO104	1-11	Collier
FIN101	1-11	James
ACC103	1-11	Wise
...	...	...

Enrolled Table

Student ID#	Subject ID	Sem.	Grade
A121	ACC101	1-11	H1
A121	ECO101	1-11	H2B
A121	ECO104	1-11	H2B
A121	FIN101	1-11	H2A
A121	ACC103	1-11	H3
A123	ACC101	1-11	H1
A123	ECO101	1-11	H2B
A123	ECO104	1-11	H2A
A123	FIN101	1-11	H3
A124	ACC101	1-11	H2A
A124	ECO101	1-11	H2A
A124	ECO104	1-11	H2B
A124	ACC103	1-11	H2B
A126	ACC101	1-11	H1
...	...	...	...

Lecturer Table

Lecturer Name	Lecturer Location	Lecturer Phone
Collier	T240D	8344-5716
Wise	T240E	8344-5309
Smyth	T240F	8344-1868
Wilkin	T240D	8344-2223
Davern	T240C	8344-1846

Anomalies in Denormalised Data

Consider the following denormalised table (relation) :

StudentID	StLastN	SubjectCode	SubjectTitle	FinalMark
123234	Lee	ISYS90001	Foundations of IS	73
123234	Lee	ISYS90012	Enterprise Systems	81
122378	Sung	ISYS90001	Foundations of IS	67
123784	Cook	ISYS90003	Ethics in IS	92
123784	Cook	ISYS90012	Enterprise Systems	
…	…	…	…	

1) ***Insertion Anomaly****:* A new subject cannot be added until at least one student has enrolled (which comes first: student or subject?)

2) ***Deletion Anomaly****:* If student Cook withdraws from ISYS90003, we lose all record of the subject ISYS90003!

3) ***Update Anomaly****:* If the Subject title for ISYS90001 changes (e.g. to Information Systems Concepts), we have to change it in multiple records (rows), otherwise the data will be inconsistent.

Normalisation

A technique used to remove undesired redundancy from databases (Break one large table into several smaller tables).

How do we normalise?

Invoice example

Bill To

John
Synex Inc
128 AA Juanita Ave
Glendora
CA 91740 US

Ship To

John
Synex Inc
128 AA Juanita Ave
Glendora
CA 91740 US

Date	14-Aug-2009	Order No		Sales Person	Charles Wooten
Shipping Date	13-Aug-2009	Shipping Terms		Terms	COD

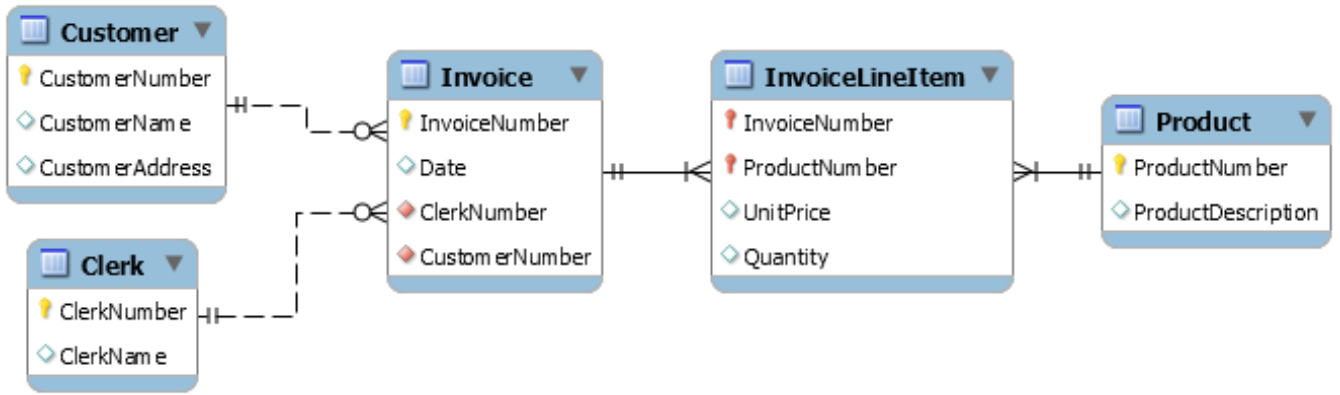
ID	SKU / Description	Unit Price (USD)	Qty	Amount (USD)
PS.V860.005	AMD Athlon X2DC-7450, 2.4GHz/1GB/160GB/SMP-DVD/VB	580.00	6.00	3,480.00
PS.V880.037	PDC-E5300 - 2.6GHz/1GB/320GB/SMP-DVD/FDD/VB	645.00	4.00	2,580.00
LC.V890.002	LG 18.5" WLCD	230.00	10.00	2,300.00
HP.Q754.071	HP LaserJet 5200	1,103.00	1.00	1,103.00

Normalised Relations and ER Diagram

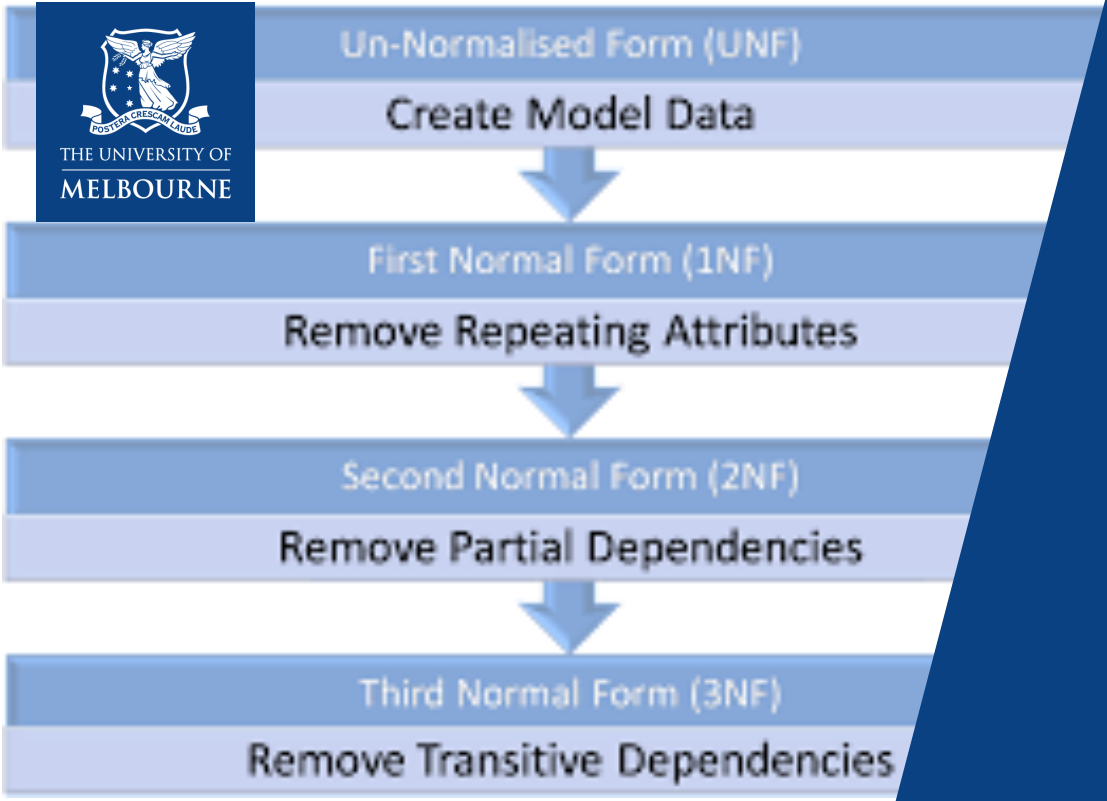
We can name the relations now

Customer (CustomerNumber, CustomerName, CustomerAddress)
Clerk (ClerkNumber, ClerkName)
Product (ProductNumber, ProductDescription)
Invoice (InvoiceNumber, Date, *CustomerNumber*, *ClerkNumber*)
InvoiceLineItem (*InvoiceNumber*, *ProductNumber*, UniPrice, Quantity)

KEY:
PK = Underline
FK = Italic
PFK Underline + Italic



Our goals is to get this table



Normalisation and Associated Concepts

Functional Dependency

- A functional dependency concerns values of attributes in a relation
- The attribute Y is **fully functionally dependent** on the attribute X if each value of X determines one and only value of Y
  Notation: X → Y
  Also known as Y = f(X)
- X determines Y (If I know X then I also know Y)

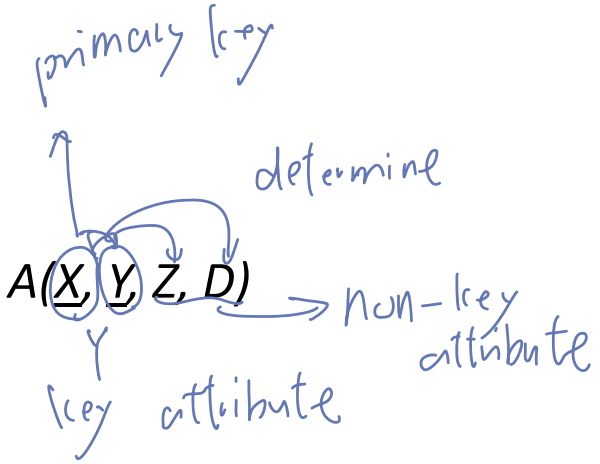
X functionally define Y

Y functionally dependent X

StudentID → Student-name

StudentID → Student-email

Functional Dependency:



- Determinants (*X,Y → Z*)
  – the attribute(s) on the left hand side of the arrow

- Key and Non-Key attributes
  – each attribute is either part of the primary key or it is not

- Partial functional dependency *(Y → Z)*
  – a functional dependency of one or more non-key attributes upon part (but not all) of the primary key, e.g. (StudentID, SubjectCode) -> StudentName

- Transitive dependency *(Z → D)*
  – a functional dependency between 2 (or more) non-key attributes, e.g. SubjCode|SubjectName|LecturerID|LecturerName results in LecturerID->LecturerName

Armstrong's Axioms

Functional dependencies can be identified using Armstrong's Axioms

1. **Reflexivity:** $B \subseteq A \Longrightarrow A \rightarrow B$

   B is a subset of A, means A functionally determines B

   Example: Student_ID, name -> name

2. **Augmentation:** $A \rightarrow B \Longrightarrow AC \rightarrow BC$

   Example: Student_ID -> st_email => Student_ID, surname ->st_email, surname

3. **Transitivity:** $A \rightarrow B \text{ and } B \rightarrow C \Longrightarrow A \rightarrow C$

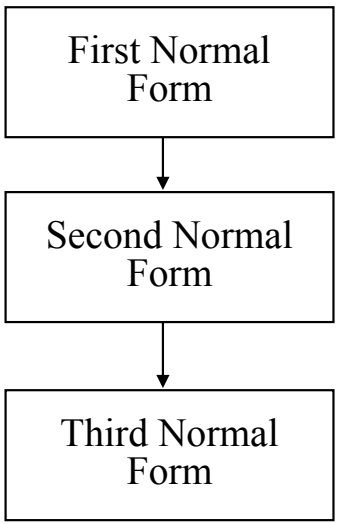
   If A functionally determines B and B functionally determines C, then A functionally determines C

   Example: ID -> birthdate, and birthdate -> age then ID ->age

D → E

Steps in Normalisation

Normal form - the state of a relation resulting from applying rules about the functional dependency of some attributes upon others



no multivalued attribute and repeating groups

Any multivalued attributes and repeating groups have been removed.
There is a SINGLE value in each cell of the table.

remove partial dependency

Any *partial* functional dependencies have been removed.
(i.e. all non-key attributes are identified by the WHOLE primary key)

remove transitive dependencies

Any *transitive* dependencies have been removed.
(i.e. all non-key attributes are identified by the primary key ONLY, they are not identified by another non-key attribute)

Invoice example

Bill To	Ship To
John **Synex Inc** 128 AA Juanita Ave Glendora CA 91740 US	**John** **Synex Inc** 128 AA Juanita Ave Glendora CA 91740 US

Date	14-Aug-2009	Order No		Sales Person	Charles Wooten
Shipping Date	13-Aug-2009	Shipping Terms		Terms	COD

ID	SKU / Description	Unit Price (USD)	Qty	Amount (USD)
PS.V880.005	AMD Athlon X2DC-7450, 2.4GHz/1GB/160GB/SMP-DVD/VB	580.00	6.00	3,480.00
PS.V880.037	PDC-E5300 - 2.6GHz/1GB/320GB/SMP-DVD/FDD/VB	645.00	4.00	2,580.00
LC.V890.002	LG 18.5" WLCD	230.00	10.00	2,300.00
HP.Q754.071	HP LaserJet 5200	1,103.00	1.00	1,103.00

Invoice example – Spreadsheet

attribute name

multivalued attribute

This is not a relational model

Invoice Number	Date	Customer Name	Customer Address	Sales Person	Terms	Product ID	Product Name	Unit Price	Quantity	Amount	Sub Total
INV0012	14-Aug-09	John / Synex	128 Juanita Ave...	Charles Wooten	COD	PSV880.006	AMD Athlon X2DC	580	6	3480	9463
						PSV880.037	PDC E5300	645	4	2580	
						LC.V890.002	LG 8.5" LCD	230	10	2300	
						HPQ754.071	HP LaserJet 5200	1103	1	1103	
INV0013	15-Aug-09	Ann / ThisCo	123 Smith Street...	Mary Smith	COD	HP Q754.071	HP LaserJet 5200	1103	2	2206	3356
						LCV890.002	LG 8.5" LCD	230	5	1150	

we don't need these two, we can know us Quantity

Repeating groups

Normalisation process

Write table in relational notation. List all the attributes that must be recorded, as one big relation.

Don’t include anything that you can derive (e.g., Age, TotalDue)

Use parentheses “()” to indicate repeating groups

Underline the primary key

Invoice(InvoiceNo, Date, CustomerNo, CustomerName, CustomerAddress, ClerkNo, ClerkName (ProductNo, ProductDesc, UnitPrice, Qty))

↳ parentheses: repeating group.

First normal form (1NF)

1. Remove repeating groups product to quantity is repeating group.
2. Identify PK for each resulting relation

Our example:

Invoice(InvoiceNo, Date, CustomerNo, CustomerName, CustomerAddress, ClerkNo, ClerkName, (ProductNo, ProductDesc, UnitPrice, Qty))

Which part is a repeating group?

Getting to 1NF in our example

1NF

* 1 to many

so add the FK to the repeat group and identify PK

Repeating group is (ProductNo, ProductDesc, UnitPrice, Qty)

The repeating fields will be removed from the original relation:

Invoice(InvoiceNo, Date, CustomerNo, CustomerName, CustomerAddress, ClerkNo, ClerkName) PK

For the removed repeating group, give the new relation a name, link it to the "leftover" relation with FK and identify PK, e.g.

It one to many

InvoiceLineItem(*InvoiceNo*, ProductNo, ProductDesc, UnitPrice, Qty)

PK

FK

one invoice many lineItem

What will be the PK of the new relation InvoiceLineItem?

Second normal form (2NF)

This step is relevant only for relations whose primary key is a COMPOSITE key (i.e., PK consisting of more than one attribute)

If a relation has a single-attribute PK, it's already in 2NF

Our result as 1NF

Invoice(InvoiceNo, Date, CustomerNo, CustomerName, CustomerAddress, ClerkNo, ClerkName) - already 2NF because there is single-attribute PK

InvoiceLineItem(*InvoiceNo*, ProductNo, ProductDesc, UnitPrice, Qty)

remove fo seperate relation

two primary key not second normal form

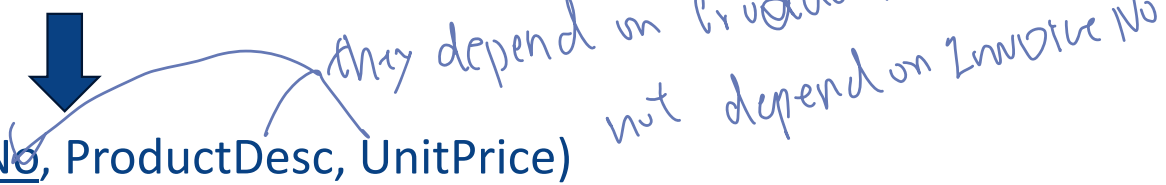
From 1NF to 2NF

Remove partial functional dependencies, where an attribute is dependent on only part of the primary key (i.e., not dependent on all the columns in a composite primary key)

Create a separate relation, containing the functionally dependent data, and the part of the key on which it depends.

Our example: Which attributes depend on part of PK (partial dependence)?

InvoiceLineItem(*InvoiceNo*, ProductNo, ProductDesc, UnitPrice, Qty)



Product(ProductNo, ProductDesc, UnitPrice)

InvoiceLineItem(*InvoiceNo*, *ProductNo*, Qty)

2NF

Our database design in 2NF

Invoice(InvoiceNo, Date, CustomerNo, CustomerName, CustomerAddress, ClerkNo, ClerkName)

Product(ProductNo, ProductDesc, UnitPrice)

InvoiceLineItem(*InvoiceNo*, *ProductNo*, Qty)

Third normal form (3NF)

1. Remove *transitive dependencies* , which are functional dependencies where one **non-key attribute** is *functionally dependent* on another **non-key attribute**. Thus, its value is only indirectly determined by the primary key.
2. Create a separate relation containing the determinant attribute and the fields that are functionally dependent on it.
3. Keep a copy of the key (FK) in the original relation. and create a new PK in a new relationship

Our example

The following two relations are in 3NF already since there are no transitive dependencies

Product(ProductNo, ProductDesc, UnitPrice)

InvoiceLineItem(*InvoiceNo*, *ProductNo*, Qty)

From 2NF to 3NF

Invoice(InvoiceNo, Date, CustomerNo, CustomerName, CustomerAddress, ClerkNo, ClerkName) – contains transitive dependencies

Extracting relations:

Clerk(ClerkNo, ClerkName)
Customer(CustomerNo, CustomerName, CustomerAddress)

What will be left in Invoice?

Invoice(InvoiceNo, *CustomerNo*, *ClerkNo*, Date)

Clerk(ClerkNo, ClerkName)
Customer(CustomerNo, CustomerName, CustomerAddress)

Product(ProductNo, ProductDesc, UnitPrice)

InvoiceLineItem(*InvoiceNo*, *ProductNo*, Qty)

final solution

Our example 3NF

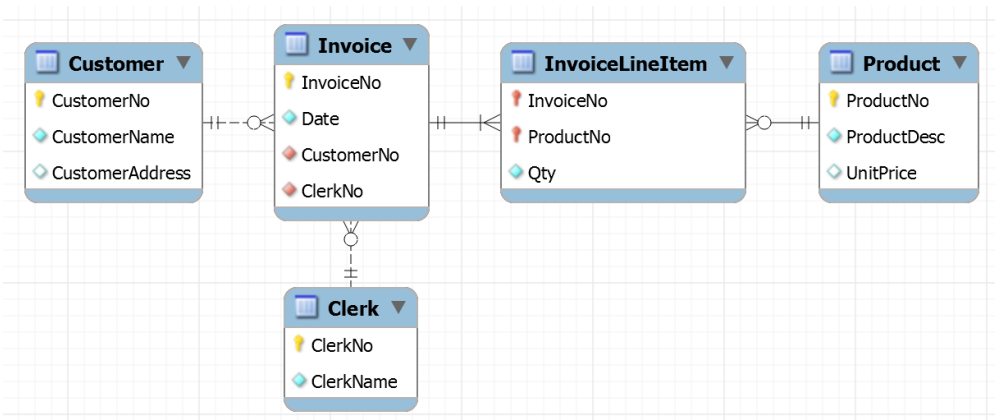
Clerk(ClerkNo, ClerkName)

Customer(CustomerNo, CustomerName, CustomerAddress)

Product(ProductNo, ProductDesc, UnitPrice)

Invoice(InvoiceNo, *CustomerNo*, *ClerkNo*, Date)

InvoiceLineItem(*InvoiceNo*, *ProductNo*, Qty)



Beyond 3NF

It is generally sufficient to stop normalizing at 3NF

However, Boyce-Codd Normal Form, 4NF, 5NF and even 6NF have been defined

http://en.wikipedia.org/wiki/Database_normalization#Normal_forms

These NFs are not examinable

Case study 1 – check your skills

Student ID	Student Name	Student DOB	Subject ID	Subject Name	Subject Faculty	Teacher ID	Teacher Name	When Taken	Result
1	Tina Zheng	2000-12-11	INFO90002	Database Systems & Information Modelling	FEIT	4	Tanya Linden	S1-2021	76
			COMP90059	Introduction to Programming	FEIT	5	Simon D'Alfonso	S1-2021	75
			ISYS10001	Foundations of Information Systems	FEIT	8	Libo Liu	S1-2021	80
			PHIL10002	Philosophy: The Big Questions	Arts	11	Russell Sage	S1-2021	77
2	Bob Brown	1999-01-01	INFO90002	Database Systems & Information Modelling	FEIT	4	Tanya Linden	S2-2021	80
			COMP90059	Introduction to Programming	FEIT	5	Simon D'Alfonso	S2-2021	72
			ISYS10001	Foundations of Information Systems	FEIT	8	Libo Liu	S2-2021	77
			PHIL30007	The Philosophy of Philosophy	Arts	11	Russell Sage	S2-2021	81

studentID, studentName, studentDoB, (subjectID, subjectName, subjectFaculty, TeacherID, teacherName, When, Result)

1NF

studentID (PK), studentName, studentDoB

studentID (PFK), subjectID (PK), subjectName, subjectFaculty, TeacherID, teacherName, WhenTaken (PK), Result

2NF

studentID (PK), studentName, studentDoB

subjectID (PK), subjectName, subjectFaculty

studentID (PFK), subjectID (PFK), teacherID, teacherName, WhenTaken, Result

3NF

studentID (PK), studentName, studentDoB

subjectID (PK), subjectName, subjectFaculty

studentID (PFK), subjectID (PFK), teacherID (FK), WhenTaken, Result

teacherID (PK), teacherName,

enrollment (studentID, studentName, studentDOB, (subjectID, subjectName, subjectfaculty, TeacherID, TeacherName, WhenTaken, Result))

1NF

PK

student (studentID, studentName, studentDOB)

enrollment (studentID, subjectID, subjectName, subjectfaculty, TeacherID, TeacherName, WhenTaken, Result)

FK PK

2 NF

PK

student (studentID, studentName, studentDOB)

enrollment (studentID, subjectID, TeacherID, TeacherName, WhenTaken, Result)

Subject (subjectID, subjectName, subjectfaculty)

3NF

PK

student (studentID, studentName, studentDOB)

Subject (subjectID, subjectName, subjectfaculty)

enrollment (studentID, subjectID, TeacherID, WhenTaken, Result)

Teacher (TeacherID, TeacherName)

Enrollment (studentID, StudentName, student DoB. (subjectID, subjectName, subjectFaculty, TeacherID. TeacherName. whenTaken . Result))

1NF

student (studenID . studentName, StudentDoB)

enrollment (subjectID, subjectName, subjectfaculty, TeacherID, TeacherName. whenTaken . Result

studentID) (FK)

primary key

2NF

student (studenID . studentName, StudentDoB)

subject (subjectID, subjectName. subjectfaculty)

subjectCoordinator (subjectID, TeacherID, TeacherName, whenTaken)

enrollment (subjectID , , whenTaken , Result, studentID)

3NF

student (studenID . studentName, StudentDoB)

subject (subjectID. subjectName. subjectfaculty)

Teacher (TeacherID. TeacherName)

subjectCoordinator (subjectID, TeacherID, , whenTaken)

enrollment (subjectID , , whenTaken , Result, studentID)

Case study 2 – check your skills

Consider a simple database for a real estate agency. The agency stores agent details and associated details of property sales.

Agent (AgentID, AgentName, Mobile (PropertyID, PropertyAddress, OwnerID, OwnerName, OwnerMobile, SaleDate, saleAmount))

Normalise this database by converting this unnormalized relation to 1NF, then 2NF, then 3NF

Agent (AgentID, AgentName, Mobile (PropertyID, PropertyAddress, OwnerID, OwnerName, OwnerMobile, SaleDate, saleAmount))

1NF: remove repeating group

this two are unique.

PropertySale (PropertyID, PropertyAddress, OwnerID, OwnerName OwnerMobile, SaleDate, saleAmount, AgentID)

Agent (AgentID, AgentName, Mobile)

2NF:

Agent (AgentID, AgentName, Mobile)

Property (PropertyID, PropertyAddress, OwnerID, ownerName, OwnerMobile)

PropertySale (PropertyID, SaleDate, saleAmount, AgentID)
PFK. FK

3NF:

PropertySale (PropertyID, SaleDate, saleAmount, AgentID)
(weak entity) PFK. FK

Agent (AgentID, AgentName, Mobile)

Property (PropertyID. PropertyAddress. OwnerID)
FK

Owner (OwnerID . OwnerName . OwnerMobile)

1 NF

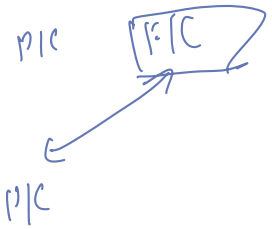
PK

(PK)

PK

FK (PK)

2 NF



What's examinable

- **Normalisation Process (1NF -> 2NF -> 3NF)**
- **Functional dependencies**
- **Armstrong's Axioms**

THE UNIVERSITY OF MELBOURNE

Thank you

Subtitle

Identifier first line

Second line