KE UNIT 3 DATA WAREHOUSING FOR BUSINESS ANALYTICS

DAY 1

Dr TIAN Jing
tianjing@nus.edu.sg

Lesson plan (part A)

Monday 16 July	**Tuesday 17 July**	**Wednesday 18 July**	**Thursday 19 July**	**Friday 20 July**
9.00 - 12.00	9.00 - 12.00	9.00 - 12.00	9.00 - 12.00	9.00 - 12.00
1b. Introduction to Data Modelling (I)	2a. Relational Database and SQL (I)	3a. Introduction to Data Warehousing	4a. Data Visualisation and Storytelling (II)	5a. Project Consultation and Presentation
Tian Jing	Tian Jing	Brandon	Brandon	Brandon / Tian Jing
13.30 - 17.00	13.30 - 17.00	13.30 - 17.00	13.30 - 17.00	13.30 - 17.00
1b. Introduction to Data Modelling (II)	2b. Relational Database and SQL (II)	3b. Data Visualisation and Storytelling (I)	4b. Data Visualisation and Storytelling (III)	5b. Project Consultation and Presentation
Tian Jing	Tian Jing	Brandon	Brandon	Brandon / Tian Jing

Assessment (Part A)

- Team: 4-5 students
- CA1 project presentation (5%)
  - 10 minutes per team
  - 5th day 20 July (Friday)
  - Concept and architecture design ONLY
- CA1 submission (15%)
  - Due date: 12 August (Sunday)
  - Refer to CA1 briefing document in IVLE
- Exam (part A): 30%

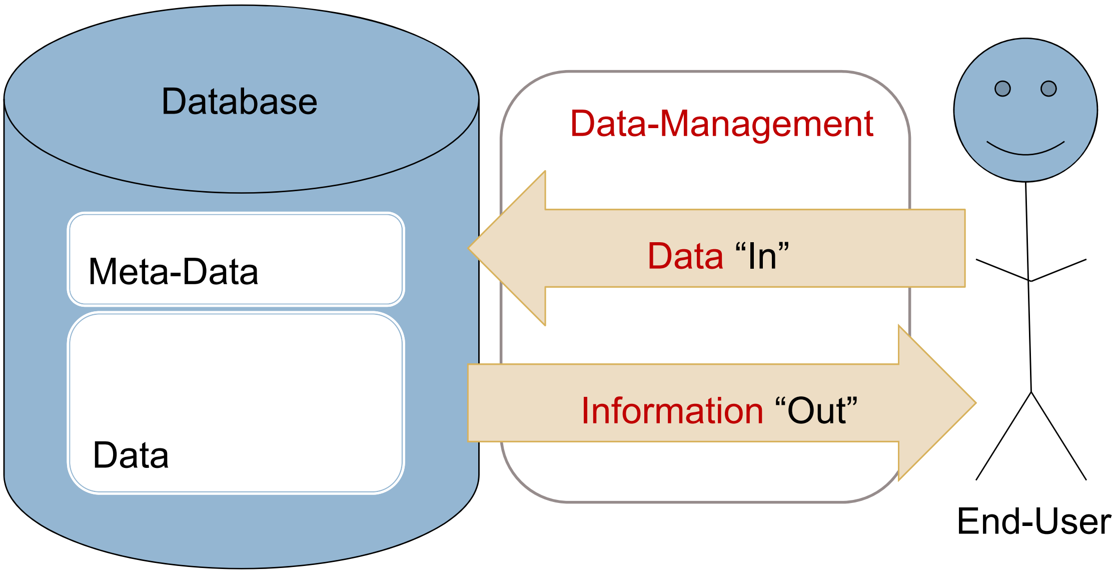
Topic

- Introduction to data modelling
- Relational data modelling
  - Entity relationship diagram (ERD).
  - Normalization, a technique that helps analysts validate the data models
  - Logical data modelling
  - Data management and query (Day 2)
- Dimensional data modelling (Day 3)
- Non-relational data modelling (part B)

Agenda

- Introduction to data modelling
- Data analysis
  - Entity relationship diagram (ERD)
  - Attribute analysis
- Data design
  - Normalization
  - Logical data model

Introduction



Introduction

Data	Data are raw unprocessed facts. By itself data has no meaning and no structure.
Information	Information is interpreted or processed data.
Data Management	•Create - adding new data •Read - retrieving information •Update - modifying existing data •Delete - removing data
Metadata	Data structure and category
Query	Asking questions of data in search of a specific answer.

Activity

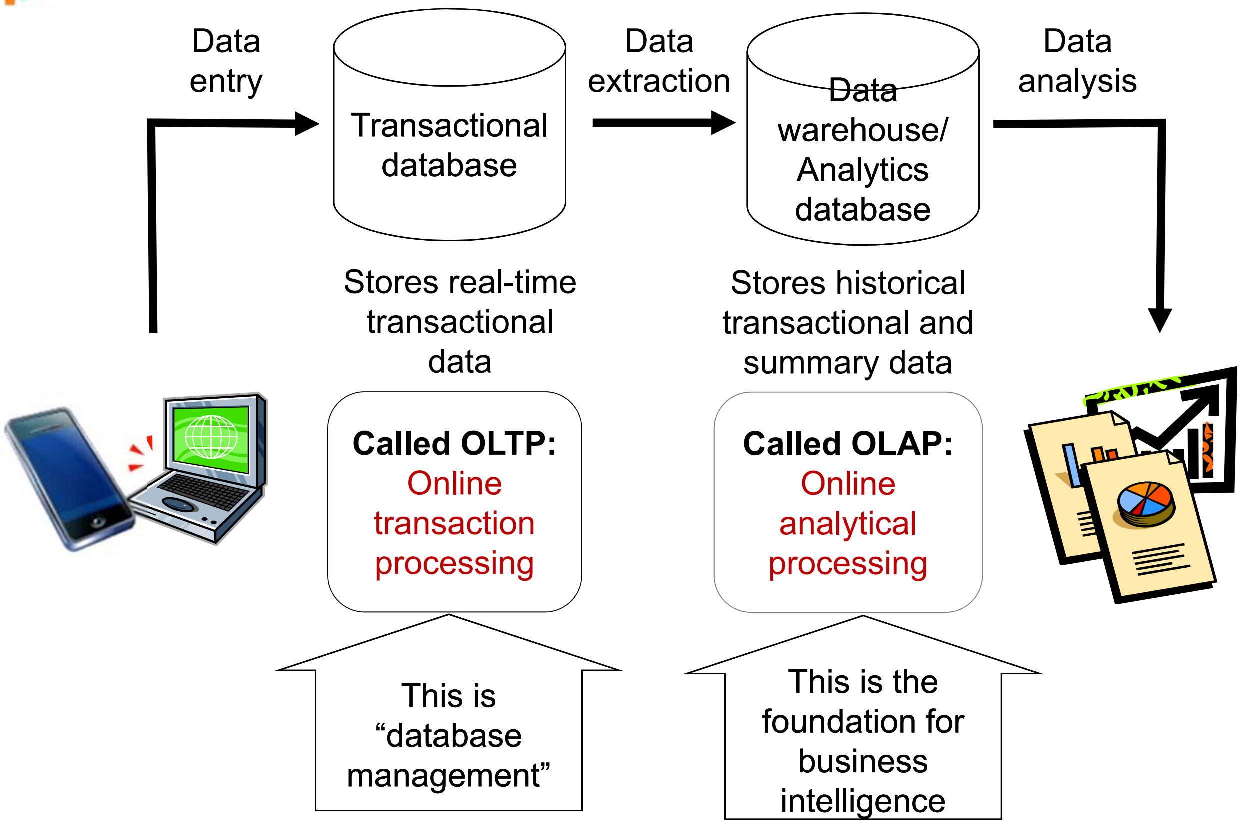
1. A Telephone book
2. Organizing the Phone Book in Alphabetical Order
3. Looking up ‘Michael Fudge’ yields the phone number 555-1234
4. How many ‘Fudges’ are there in the phone book?
5. Employee records (in a file cabinet)
6. Filing a new employee under “W” because their last name is “Williams”
7. The average employee salary is $40,000

Data?	
Information?	
Data Management?	
Metadata?	
Query?	

Two types of data

Transactional	Analytical
• Captures data describing and event • An exchange between actors • Real-time	• Captures data to support analysis and reporting • An aggregated view of the business • Historical

Two types of data processing



Data entry

Transactional database

Data extraction

Data warehouse/ Analytics database

Data analysis

Stores real-time transactional data

Stores historical transactional and summary data

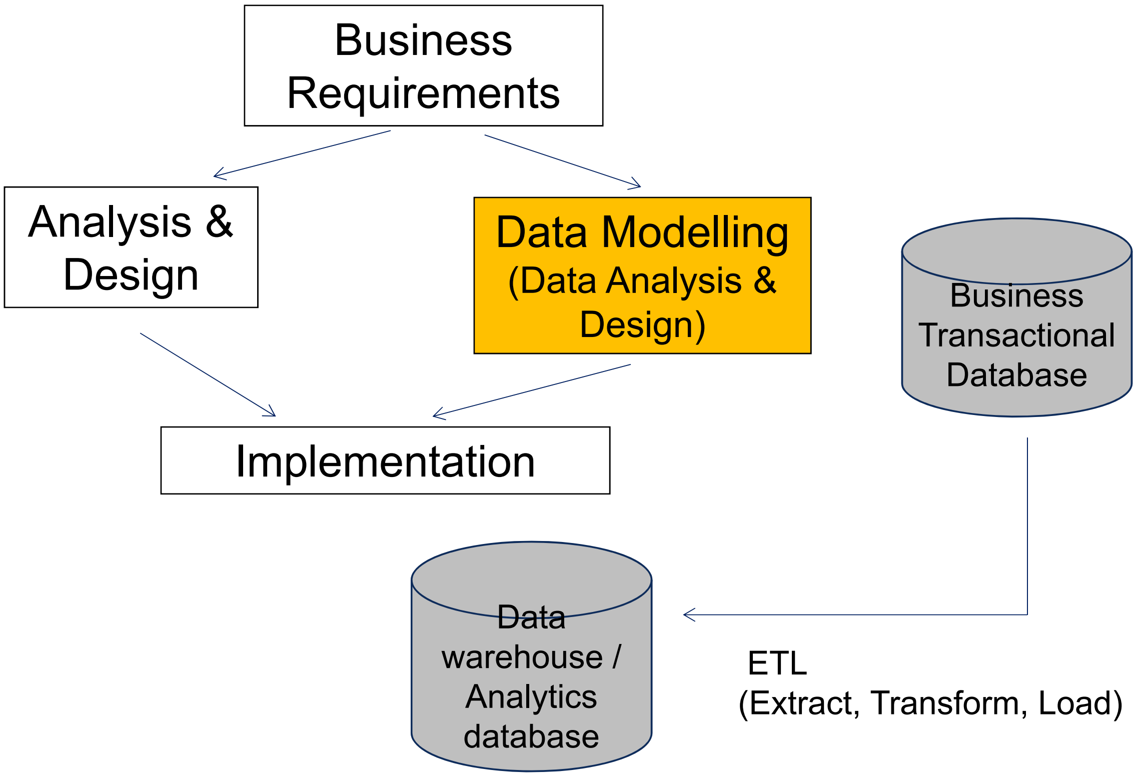
Called OLTP: Online transaction processing

Called OLAP: Online analytical processing

This is “database management”

This is the foundation for business intelligence

Application development workflow

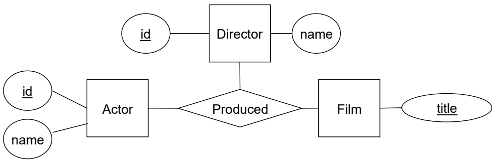

Business Requirements
Analysis & Design
Data Modelling (Data Analysis & Design)
Business Transactional Database
Implementation
Data warehouse / Analytics database
ETL (Extract, Transform, Load)

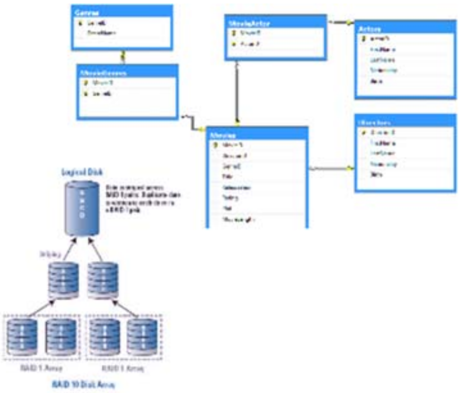
Data modelling

- What is a database?
  - A collection of files storing related data
- What is a database management system (DBMS)?
  - An application program that allows us to manage efficiently the collection of data files.
- What is data model?
  - Mathematical formalism (or conceptual way) for describing the data

Data modelling

- Requirement analysis
  - What information needs to be stored? How will it be used? What integrity constraints should be imposed?
- Conceptual data modelling
  - Define/describe/discuss the semantic modeling of data in the application (Entities Relations Diagrams)
- Logical data modelling
  - Enhance the Entities Relations Diagrams by optimizations and relationships
- Physical data modelling
  - Translate the database schema into a physical storage plan on available hardware (DBMS, SQL, day 2)





Data analysis and data design

- Data Analysis
  - Develop a data model for data required by the business requirements
  - A data model consists of
    - Entities Relations Diagram
    - Data Dictionary
- Data Design
  - Restructure the data model so that it is optimised or suitable for data accesses
    - Required before designing and implementing physical database
  - Logical data model (through normalization)

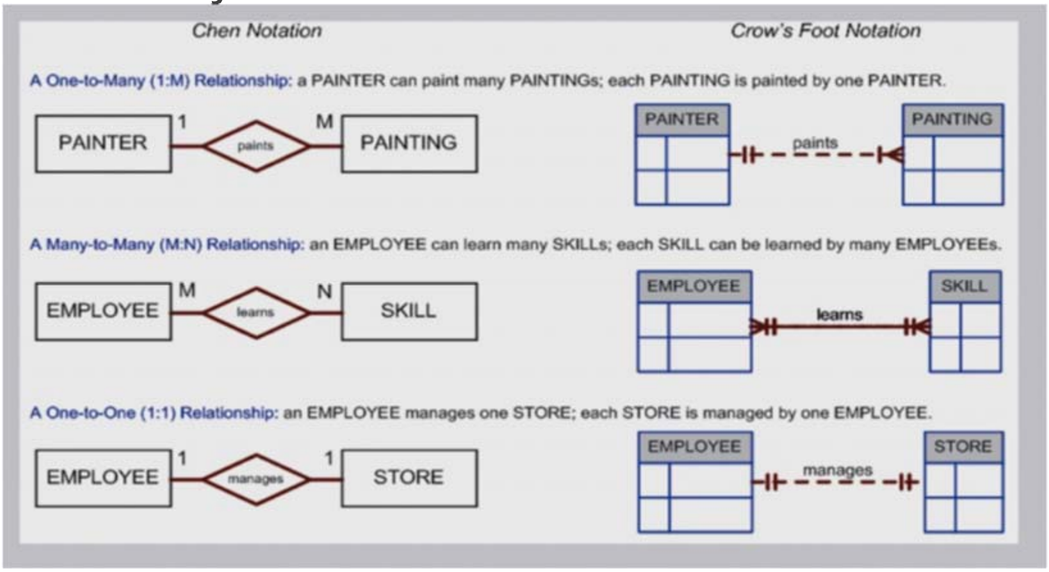
Agenda

- Introduction to data modelling
- Data analysis
  - Entity relationship diagram (ERD)
  - Attribute analysis
- Data design
  - Normalization
  - Logical data model

Entity relationship diagram (ERD)

- Entities
- Relationships
- Cardinality

We use Crow's Foot Notation in our class



A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

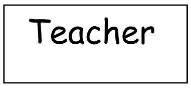
A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

Entities

- A distinguishable objects in the problem domain that we want to model.
- You need to distinguish:
  - Entity Type (or Entity)
  - Entity Occurrence

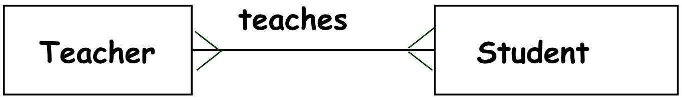
Example: In a ISS Course Registration System

- Teacher is an Entity Type
- Brandon and I are the Entity Occurrences

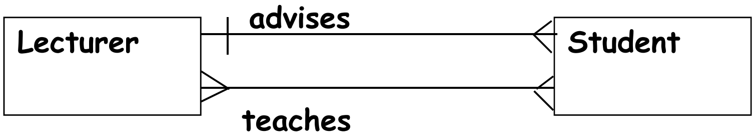


Relationships

- Depends on the business rules
- Every Relationship is bi-directional
  - a teacher teaches one or more students
  - a student is taught by one or more teacher



- There may be more than one important relationships
  - a teacher counsels zero, one or more students
  - a teacher teaches one or more students
- Every Relationship is described in terms of a verb



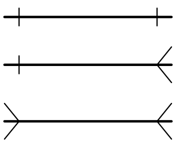
Cardinality of relationships

- Cardinality of relationships: How many
  - The number of occurrences of one entity type that relate to the occurrences of another entity type

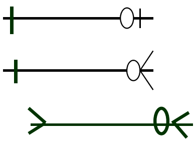
one- to- one

one- to- many

many-to-many

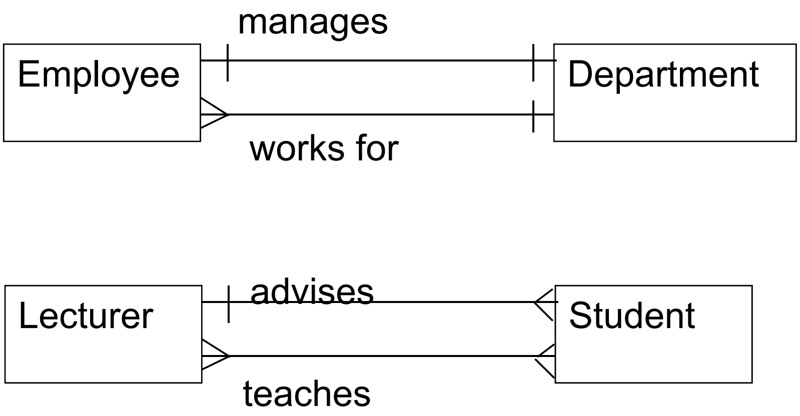


- Optional Relationships
  - one to zero or one
  - one to zero or many
  - many to zero or many



Example (1)

- An education institute management system



manages

Employee

Department

works for

advises

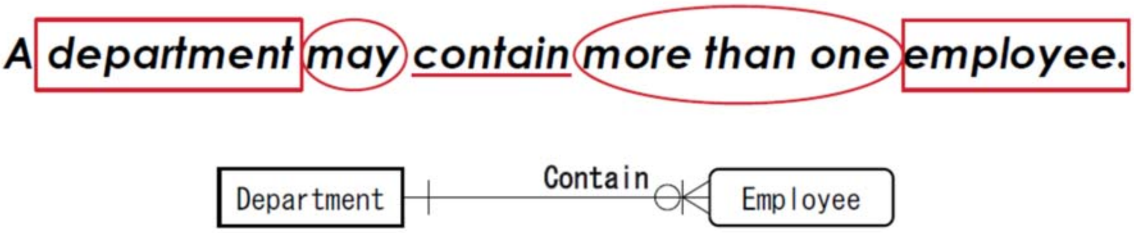
Lecturer

Student

teaches

Example (2)

- Data modelling is similar to diagramming a sentence
  - Place boxes around the ‘nouns’, or entities.
  - Underline the ‘verbs’.
  - Circle the ‘how many’ qualifier.
  - Look for optionality words such as ‘may/must’.

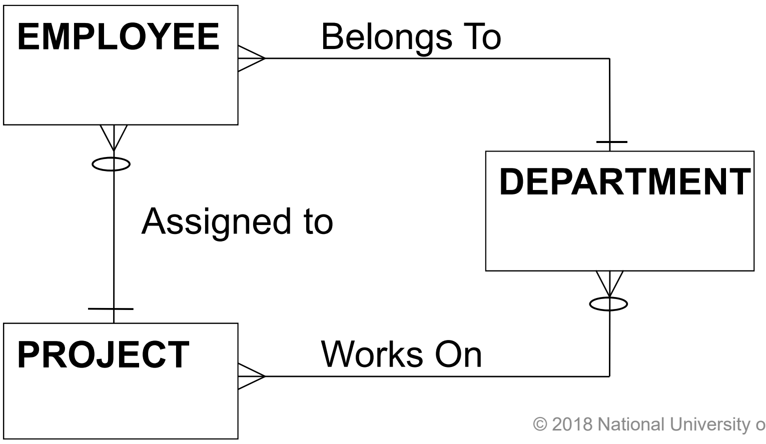


A department may contain more than one employee.

Example (3)

A project management system with objective to keep track of

- all the projects undertaken by the company
- assignments of projects to departments
  - a project can be assigned to one or more departments
  - each department can take on one or more projects
  - one or more employee of a department may work on a project
  - a employee can only take on one project at any one time



Agenda

- Introduction to data modelling
- Data analysis
  - Entity relationship diagram (ERD)
  - Attribute analysis
- Data design
  - Normalization
  - Logical data model

Attribute analysis

- Meta Data
  - Data about data
- Data Dictionary
  - a repository to store all information, description about the data
  - Contains metadata

Data dictionary

ATTRIBUTE NAME	TYPE	LENGTH	DEFINITIONS AND BUSINESS RULES
EMP-M	A	25	Employee Name (full name, start with the surname)
Emp-JOB-T	A	25	Employee Job title (Programmer, Analyst, Project Manager, Department Manager)
EMP-JOB-DESC	A	60	Simple short description
PROJ-M	A	10	A unique short name given to the project
MTH-SAL-A	N	6.2	Monthly salary (999999.99)
EMP-PROJ-START-D	D	8	Start date of the project (DDMMYYYY)
EMP-PROJ-END-D	D	8	End date of the project (DDMMYYYY)

Note: A – Alphanumeric, N – Numeric, D - date

Example

- Recall for the project management system, possible attributes of each entity are
  - Employee
    - employee name,job title, job description, project name, employee skill type, employee skill type description, employee monthly salary etc
  - Department
    - department name, department manager number, department manager name, department employee size, project name, project-department budget allocated, department employee number, department employee name
  - Project
    - project name, project description, project budget allocated, project start date, project end date

Example

- Attributes of the Project Management system

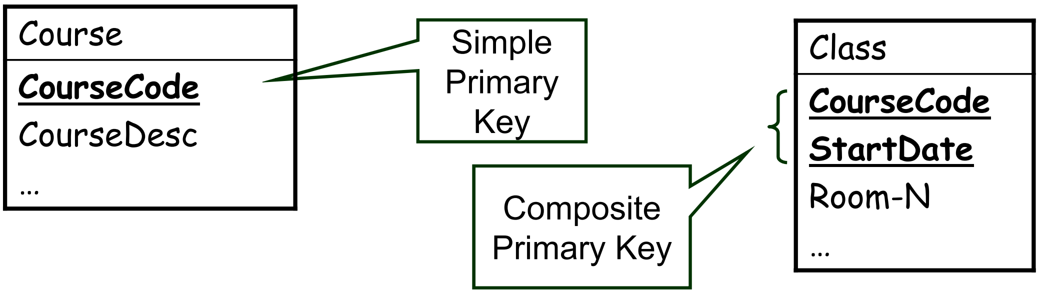
ENTITY	ATTRIBUTE NAME	TYPE	LEN	DEFINITION AND BUSINESS RULES
Employee	EMP-N	N	6	Employee number. Unique for each employee.
	EMP-M	A	25	Employee name in the form of the last name and two initials.
	EMP-JOB-T	A	25	Employee job Title (analyst, programmer, project manager, department manager)
	EMP-JOB-DESC	A	60	Short description of job title
	SAL-CHNG-D	N	8	The effective date of the employee's salary.. Employee salary history is kept for 3 years.
	MTH-SAL-A	N	6.2	Monthly salary (in the form of 999999.99).
	PROJ-M	A	10	Name of project currently assigned to employee Each project has a unique name
	EMP-PROJ-START-D	D	8	Start date of the employee on the project
	EMP-PROJ-END-D	D	8	End date of the employee on the project
	SKILL-TYPE-C	A	6	Skill type code
	SKILL-TYPE-DESC	A	20	Description of the skill (usually an employee has more than one skill

Example

ENTITY	ATTRIBUTE NAME	TYPE	LEN	DEFINITION AND BUSINESS RULES
Department	DEPT-M	A	10	A unique name for each department
	DEPT-MGR-N	N	6	Employee number of the manager in charge of department
	DEPT-MGR-M	A	25	Name of the manager in charge of department
	DEPT-EMP-SIZE-Q	N	3	Number of employee in the department
	DEPT-PROJ-M	A	10	Project (name) currently assigned to department
	DEPT-PROJ-BUDGT-A	N	6.2	Allocated budget(money) for the project for that department
	DEPT-EMP-N	N	6	Employee Numbers of all employees working in the department
	DEPT-EMP-M	A	25	Employee names of all employees working in the department
Project	PROJ-M	A	10	Unique name assigned to project. Project currently work on by company.
	PROJ-DESC	A	100	Short description of the project
	PROJ-BUDGT-A	N	6.2	Total budget (money) allocated to the entire project
	PROJ-START-D	D	8	Project start date (DDMMYYYY)
	PROJ-END-D	D	8	Project end date (DDMMYYY). The project must end by this date.

Identifiers/Keys

- Primary key (PK): unique identifier of a record
  - A simple key is a key consisting of a single attribute
  - A composite key is a key consisting of more than one attribute
  - No two rows must contain the same value for their primary keys
  - None of the component attributes of the identifier may have null values.



Example: Identify key attributes

ENTITY	ATTRIBUTE NAME	TYPE	LEN	DEFINITION AND BUSINESS RULES
Employee	EMP-N	N	6	Employee number. Unique for each employee.
	EMP-M	A	25	Employee name in the form of the last name and two initials.
	EMP-JOB-T	A	25	Employee job Title (analyst, programmer, project manager, department manager)
	EMP-JOB-DESC	A	60	Short description of job title
	SAL-CHNG-D	D	8	Date an employee’s salary was changed, in the form of ‘DDMMYYYY`. Employee salary history is kept for 3 years.
	MTH-SAL-A	N	6.2	Monthly salary after change on a given date (in the form of 999999.99).
	PROJ-M	A	10	Name of project assigned to employee Each project has a unique name
	EMP-PROJ-START-D	D	8	Start date of employee assignment to the project, DDMMYYYY (assignment period of employee to the project varies between employees)
	EMP-PROJ-END-D	D	8	End date of employee assignment to the project, DDMMYYYY (assignment period of employee to the project varies between employees)
	SKILL-TYPE-C	A	6	Skill type code
	SKILL-TYPE-DESC	A	20	Description of the skill

Example: Identify key attributes

ENTITY	ATTRIBUTE NAME	TYPE	LEN	DEFINITION AND BUSINESS RULES
Department	DEPT-M	A	10	A unique name for each department
	DEPT-MGR-N	N	6	Employee number of the manager in charge of department
	DEPT-MGR-M	A	25	Name of the manager in charge of department
	DEPT-EMP-SIZE-Q	N	3	Number of employee in the department
	DEPT-PROJ-M	A	10	Project (name) assigned to department
	DEPT-PROJ-BUDGT-A	N	6.2	Allocated budget(money) for the project for that department
	DEPT-EMP-N	N	6	Employee Numbers of all employees working in the department
	DEPT-EMP-M	A	25	Employee names of all employees working in the department
Project	PROJ-M	A	10	Unique name assigned to project
	PROJ-DESC	A	100	Short description of the project
	PROJ-BUDGT-A	N	6.2	Total budget (money) allocated to the entire project
	PROJ-START-D	D	8	Project start date (DDMMYYYY)
	PROJ-END-D	D	8	Project end date (DDMMYYY). The project must end by this date.

Summary: Design ERD

- Identify the entities
  - If you begin the data model using a use case, look at the major inputs and outputs of the use case.
  - If the process models are available, look at the data stores, external entities, and data flows.
- Add the appropriate attributes to each entity
  - One or more of the attributes will become the entity’s identifier.
- Draw relationships among entities
  - Each relationship is labeled, and cardinality and modality are assigned.

Agenda

- Introduction to data modelling
- Data analysis
  - Entity relationship diagram (ERD)
  - Attribute analysis
- Data design
  - Normalization
  - Logical data model

What is normalization?

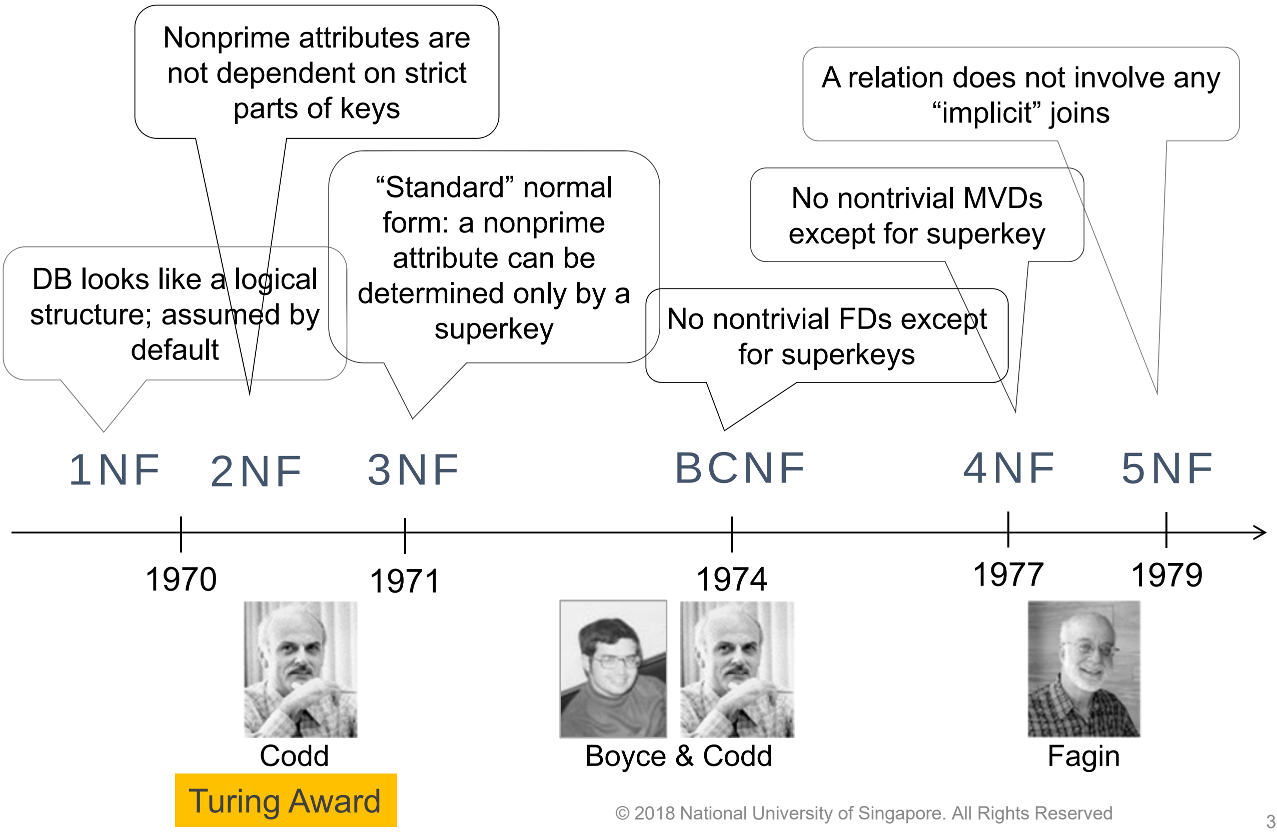
- A technique to organize **“efficiently”** organize data in a database
- **“Efficiently”:**
  - Eliminating redundant data
    - Not storing the same data in more than one table
  - Ensuring that functional dependencies make sense

Without normalization

student_id	name	address	subject
401	Adam	133 Our Lane	Biology
402	Alex	123 Here Lane	Math
403	Stuart	123 My Lane	Math
404	Adam	123 Their Lane	Physics

- **Update Anomaly :** To update address of a student who occurs twice or more than twice in a table, we will have to update **address** column in all the rows, else data will become inconsistent.
- **Insertion Anomaly :** Suppose for a new admission, we have a Student id(S_id), name and address of a student but if student has not opted for any subjects yet then we have to insert **NULL** there, leading to Insertion Anamoly.
- **Deletion Anomaly :** If (student_id) 401 has only one subject and temporarily he drops it, when we delete that row, entire student record will be deleted along with it.

History of normal forms



Nonprime attributes are not dependent on strict parts of keys

A relation does not involve any "implicit" joins

"Standard" normal form: a nonprime attribute can be determined only by a superkey

No nontrivial MVDs except for superkey

DB looks like a logical structure; assumed by default

No nontrivial FDs except for superkeys

1NF 2NF 3NF BCNF 4NF 5NF

1970 1971 1974 1977 1979

Codd

Turing Award

Boyce & Codd

Fagin

Business case: Order processing

Id: A1091

Purchase Order

Customer ID: *S009*
Customer Name: *Lynn Wang*
Date: *21/7/2011*

ItemCode | Description | Qty

S1001 Pencil 100
S1003 Eraser
200
S1005 Ruler
250

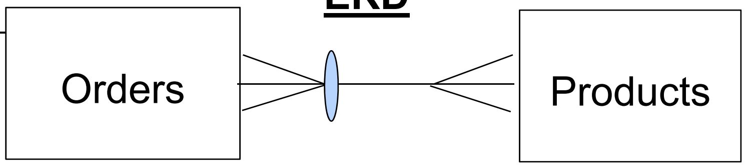
Total Number of Items: 3

Product Code : S1001
Product Name: Pencil
Unit Price: $0.20

Business Rules:

- An order contain 1 to many products
- A product may be appear in multiple order
- A product may not be ordered if it is not popular!

ERD

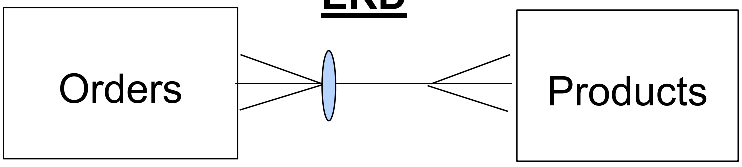


Initial design of ERD

Orders
OrderID
CustomerID
CustomerName
OrderDate
ProductID
ProductName
Qty
ProductsTotal

Products
ProductID
ProductName
UnitPrice

ERD



Overview of Normalization

0NF-> 1NF	- **Multivalue Attributes/Repeating Groups?** - Move multivalue/repeating group to a new entity -determine the key of the new relation
1NF ->2NF	- Are there **attributes dependent on a partial key** (of **composite key**)? •Move attribute(s) to a new entity -determine the key of the new entity
2NF ->3NF	- Any **non-key attribute dependent on any other non-key attribute**? - Move attribute(s) to a new entity -determine the key of the new entity
	Optimization – combining entities with same primary key / remove derivable attribute

Question

Question: Check all tables with multivalue repeating attributes?

OrderID [PK]	CustomerID	Customer Name	Order Date	ProductID	Product Name	Qty	Products Total
A1091	S009	Lynn Wang	21/7/2011	S1001 S1003 S1005	Pencil Eraser Ruler	100 200 250	3
A1092	S010	Suzan Tan	12/1/2011	S1001 S1004	Pencil Pen	10 50	2
A1093	S010	Suzan Tan	21/8/2011	S1001 S1005	Pencil Ruler	80 90	2

Multivalue attribute

Multivalue attribute

Multivalue attribute

Repeating group
(group of related multivalue attributes)

ONF → 1NF: Action required

- If there are multivalue attributes and/or repeating groups
  - Place the each attribute/group into a separate new table
  - Copy the primary key from the original table to the new tables
- Examine the new table and determine which additional attribute(s) are needed to uniquely identify a single row of the new table. The primary key from the original table usually is insufficient to be the primary key in the new table
- Give a names to the new table

Example

Orders

OrderID [PK]	CustomerID	CustomerName	Order Date	ProductsTotal
A1091	S009	Lynn Wang	21/7/2011	3
A1092	S010	Suzan Tan	12/1/2011	2
A1093	S010	Suzan Tan	21/8/2011	2

OrderDetails

Order ID [PK]	ProductID [PK]	Product Name	Qty
A1091	S1001	Pencil	100
A1091	S1003	Eraser	200
A1091	S1005	Ruler	250
A1092	S1001	Pencil	10
A1092	S1004	Pen	50
A1093	S1001	Pencil	80
A1093	S1005	Ruler	90

0NF → 1NF

Orders
OrderID
CustomerID
CustomerName
OrderDate
ProductID
ProductName
Qty
ProductsTotal

Products
ProductID
ProductName
UnitPrice

0NF (top) → 1NF (bottom)
We have split table(s) with multi-value attributes or repeating group

Orders
OrderID
CustomerID
CustomerName
OrderDate
ProductsTotal

OrderDetails
OrderID
ProductID
ProductName
Qty

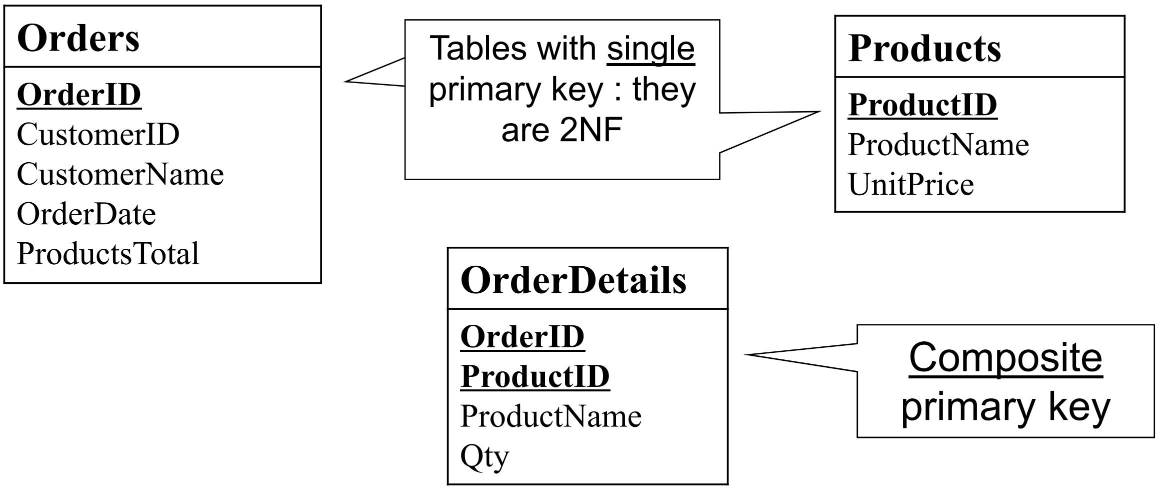
Products
ProductID
ProductName
UnitPrice

Next step (1NF → 2NF)

0NF-> 1NF	• **Multivalue Attributes/Repeating Groups?** • Move multivalue/repeating group to a new entity -determine the key of the new relation
1NF ->2NF	• Are there **attributes dependent on a partial key** (of **composite key**)? •Move attribute(s) to a new entity -determine the key of the new entity
2NF ->3NF	• Any **non-key attribute dependent on any other non-key attribute**? • Move attribute(s) to a new entity -determine the key of the new entity
	Optimization – combining entities with same primary key / remove derivable attribute

Question

Question: Check all tables with composite primary key: any attribute depends on part of whole-key?



Orders

OrderID
CustomerID
CustomerName
OrderDate
ProductsTotal

Tables with single primary key : they are 2NF

Products

ProductID
ProductName
UnitPrice

OrderDetails

OrderID
ProductID
ProductName
Qty

Composite primary key

Example

Table with composite primary key

OrderDetails

OrderID [PK]	ProductID [PK]	ProductName	Qty
A1091	S1001	Pencil	100
A1091	S1003	Eraser	200
A1091	S1005	Ruler	250
A1092	S1001	Pencil	10
A1092	S1004	Pen	50
A1093	S1001	Pencil	80
A1093	S1005	Ruler	90

ProductName depends on ProductID (part of composite key) and not OrderID

1NF→ 2NF: Action required

- Table with a single primary key is already 2NF
- Table with a compose primary key
  - Check each attribute against the whole key, move attribute(s) and the part of the key on which it depends to form a new table
  - Name the new tables(s)
  - Decide on the primary key of the new table

Example

OrderDetails

OrderID [PK]	ProductID [PK]	Product Name	Qty
A1091	S1001	Pencil	100
A1091	S1003	Eraser	200
A1091	S1005	Ruler	250
A1092	S1001	Pencil	10
A1092	S1004	Pen	50
A1093	S1001	Pencil	80
A1093	S1005	Ruler	90

Split table

OrderDetails

OrderID [PK]	ProductID [PK]	Qty
A1091	S1001	100
A1091	S1003	200
A1091	S1005	250
A1092	S1001	10
A1092	S1004	50
A1093	S1001	80
A1093	S1005	90

OrderDetailsProduct

ProductID [PK]	ProductName
S1001	Pencil
S1003	Eraser
S1005	Ruler

1NF → 2NF

Orders
OrderID
CustomerID
CustomerName
OrderDate
ProductsTotal

OrderDetails
OrderID
ProductID
ProductName
Qty

Products
ProductID
ProductName
UnitPrice

1NF (top) → 2NF (bottom)
We have split table(s) with attributes dependent on a partial key (of composite key)

Orders
OrderID
CustomerID
CustomerNam
e OrderDate
ProductsTotal

OrderDetails
OrderID
ProductID
Qty

OrderDetailsProduct
ProductID
ProductName

Products
ProductID
ProductName
UnitPrice

Next step (2NF → 3NF)

0NF-> 1NF	• **Multivalue Attributes/Repeating Groups?** • Move multivalue/repeating group to a new entity -determine the key of the new relation
1NF ->2NF	• Are there **attributes dependent on a partial key** (of **composite key**)? •Move attribute(s) to a new entity -determine the key of the new entity
2NF ->3NF	• Any **non-key attribute dependent on any other non-key attribute?** • Move attribute(s) to a new entity -determine the key of the new entity
	Optimization - combining entities with same primary key / remove derivable attribute

Question

- Check all tables: any attribute depends on non-key attribute

Orders

OrderID [PK]	CustomerID	CustomerName	OrderDate	ProductsTotal
A1091	S009	Lynn Wang	21/7/2011	3
A1092	S010	Suzan Tan	12/1/2011	2
A1093	S010	Suzan Tan	21/8/2011	2

CustomerName depends on CustomerID (non-key attribute)

2NF → 3NF: Action required

- Examine each attribute
- If an attribute(s) does not depend on the whole key, or it depends on another non-key attribute, remove the attribute(s) and use attribute on which it depends on to form a new relation. i.e. create new table comprising the attribute(s) and the non-key attribute upon which it depends
- Determine the key(s) for the new table(s)
- Name the new table(s)

Example

Orders

OrderID[PK]	CustomerID	CustomerName	OrderDate	ProductsTotal
A1091	S009	Lynn Wang	21/7/2011	3
A1092	S010	Suzan Tan	12/1/2011	2
A1093	S010	Suzan Tan	21/8/2011	2

split table

Orders

Order ID [PK]	Customer ID	Order Date	Products Total
A1091	S009	21/7/2011	3
A1092	S010	12/1/2011	2
A1093	S010	21/8/2011	2

Customers

CustomerID	Customer Name
S009	Lynn Wang
S010	Suzan Tan

2NF → 3NF

Orders
OrderID CustomerID CustomerName OrderDate ProductsTotal

OrderDetails
OrderID **ProductID** Qty

OrderDetailsProduct
ProductID ProductName

Products
ProductID ProductName UnitPrice

2NF (top) → 3NF (bottom)
We have split table(s) with any attribute depends on non-key attribute

Orders
OrderID CustomerID OrderDate ProductsTotal

Customers
CustomerID CustomerName

OrderDetails
OrderID **ProductID** Qty

OrderDetailsProduct
ProductID ProductName

Products
ProductID ProductName UnitPrice

Optimization

Orders
OrderID
CustomerID
OrderDate
ProductsTotal

Customers
CustomerID
CustomerName

OrderDetails
OrderID
ProductID
Qty

OrderDetailsProduct
ProductID
ProductName

Products
ProductID
ProductName
UnitPrice

Optimization: Combine tables with same primary key

Orders
OrderID
CustomerID
OrderDate
ProductsTotal

Customers
CustomerID
CustomerName

OrderDetails
OrderID
ProductID
Qty

~~OrderDetailsProduct~~
~~**ProductID**~~
~~ProductName~~

Products
ProductID
ProductName
UnitPrice

Merged
(same primary key)

Optimization

Orders

OrderID [PK]	CustomerID	OrderDate	Products Total
A1091	S009	21/7/2011	3
A1092	S010	12/1/2011	2
A1093	S010	21/8/2011	2

Derivable attribute

OrderDetails

OrderID [PK]	ProductID [PK]	Qty
A1091	S1001	100
A1091	S1003	200
A1091	S1005	250
A1092	S1001	10
A1092	S1004	50
A1093	S1001	80
A1093	S1005	90

Optimization

Orders
OrderID
CustomerID
OrderDate
ProductsTotal

Customers
CustomerID
CustomerName

OrderDetails
OrderID
ProductID
Qty

Products
ProductID
ProductName
UnitPrice

Optimization: Remove derivable attributes

Orders
OrderID
CustomerID
OrderDate
~~ProductsTotal~~

Removed (derivable Attribute)

Customers
CustomerID
CustomerName

OrderDetails
OrderID
ProductID
Qty

Products
ProductID
ProductName
UnitPrice

Note: Optimization can be done at the end of each normalization step, or after 3NF.

Summary

Orders
OrderID
CustomerID
CustomerName
OrderDate
ProductID
ProductName
Qty
ProductsTotal

Products
ProductID
ProductName
UnitPrice

0NF (top) → 3NF optimized (bottom)

Orders
OrderID
CustomerID
OrderDate

Customers
CustomerID
CustomerName

OrderDetails
OrderID
ProductID
Qty

Products
ProductID
ProductName
UnitPrice

How “far” should we Normalize?

- For relational databases:
  - 1NF is required, at minimum for practical RDBMS implementations.
  - The majority of the time data models are normalized to 3NF.
  - Sometimes certain tables are left in 1NF or 2NF, for performance or practical reasons.
  - Higher normal forms BCNF, 4NF are rare.
- In General, the Higher the NF of your data model:
  - The more complicated the internal data model
  - The more “programming” required to reproduce the external data model.
  - But, the lesser the chance for data anomalies!!
- It’s a total trade-off: Database complexity vs. data anomalies.

Agenda

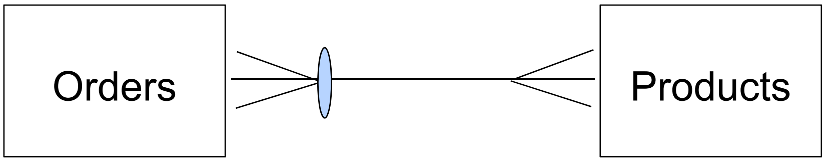
- Introduction to data modelling
- Data analysis
  - Entity relationship diagram (ERD)
  - Attribute analysis
- Data design
  - Normalization
  - Logical data model

Logical data model

- Logical data model is a more detailed representation of data
  - Additional Entities (as a result) of normalization
  - Additional Relationships (linking new and existing entities)

Example

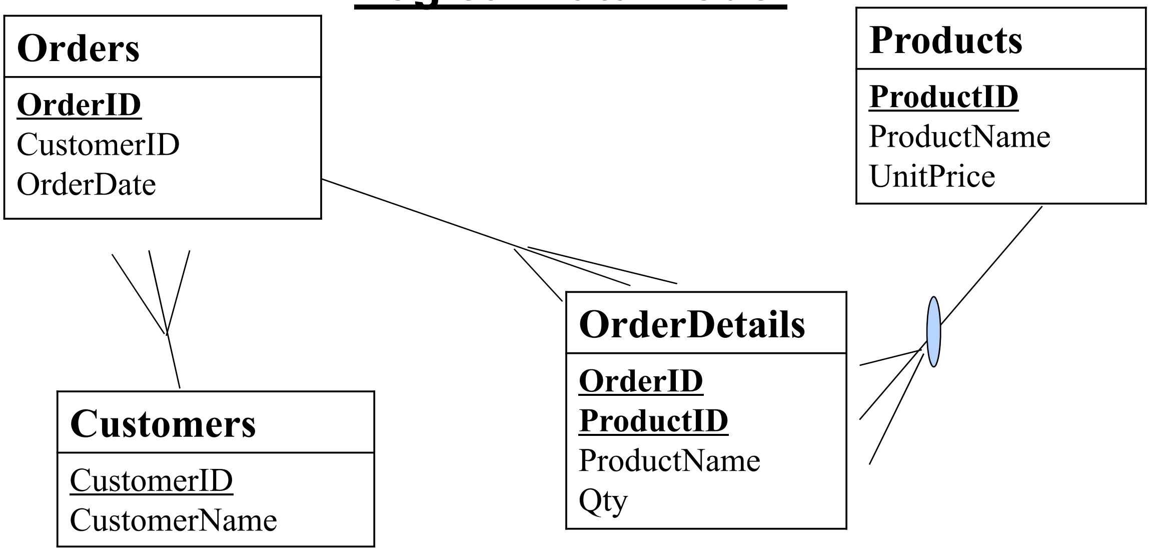
ERD (Before Normalisation)



Orders

Products

Logical Data Model



Orders

OrderID
CustomerID
OrderDate

Products

ProductID
ProductName
UnitPrice

OrderDetails

OrderID
ProductID
ProductName
Qty

Customers

CustomerID
CustomerName

Summary

- Design entity relation diagram
- Perform normalization on a conceptual data model
- Design logical data model

Thank you!

Dr TIAN Jing
Email: tianjing@nus.edu.sg