Lecture 2

Relational Databases and Normalisation

Bikesh Raj Upreti
E-mail: b.upreti@uq.edu.au
Room: 530 Joyce Ackroyd (37) Building

The University Of Queensland AUSTRALIA

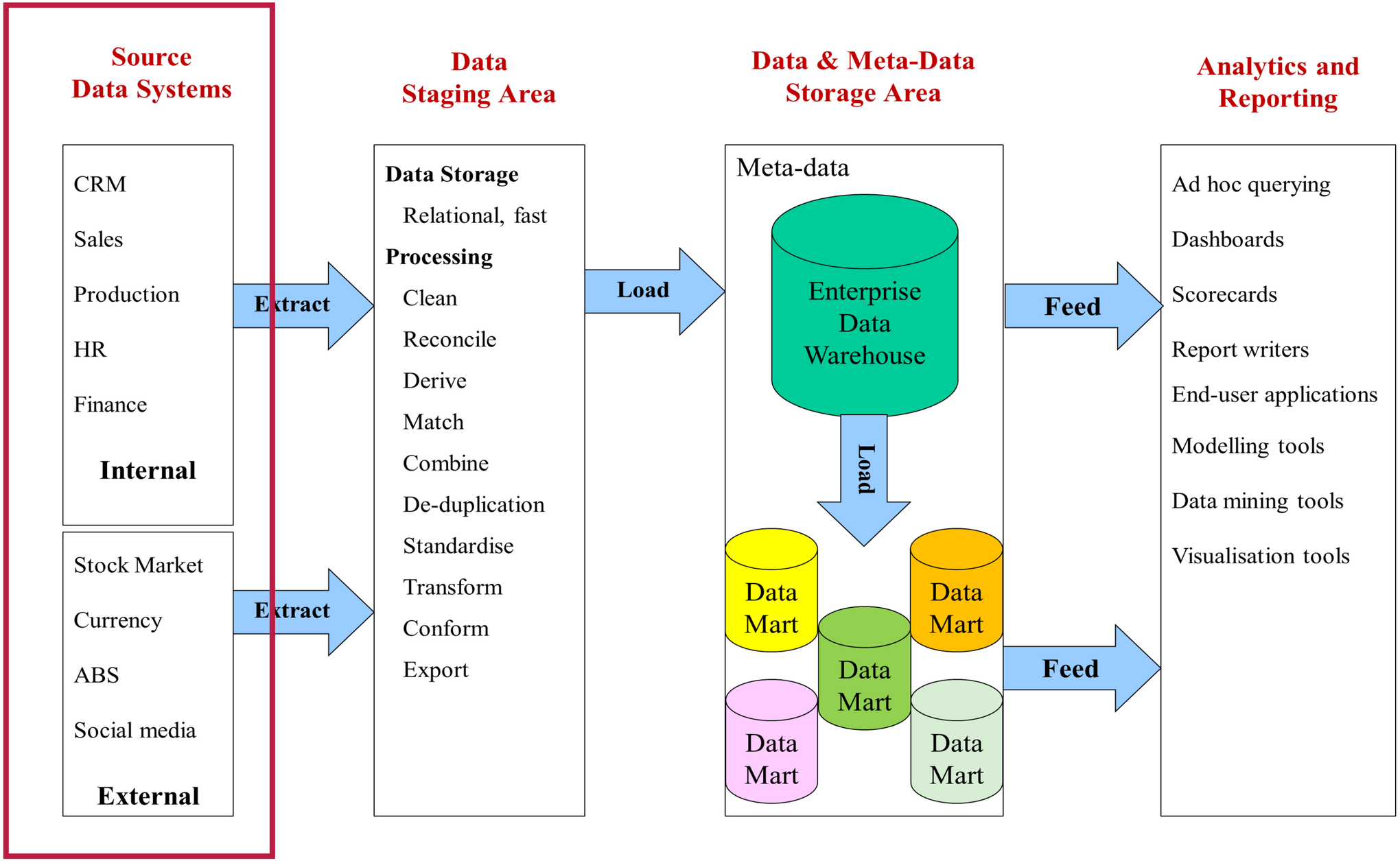
UQ BUSINESS SCHOOL

Recap: What is strategic information?

- Information that helps to make decisions on the formulation and execution of business strategies and objectives
- It is not!
  - Information for the daily business operations
    - Its not information to
      - Produce an invoice, Make a shipment, Settle a claim etc.
- Characteristics

Integrated	A single view of the firm. An Enterprise wide view
Data Integrity	Accurate and conform to business rules
Accessible	Easily accessible, intuitive access, responsive analysis
Credible	Trusted values – every business value has ≡ 1 value
Timely	Must be available with the correct timeliness for the data

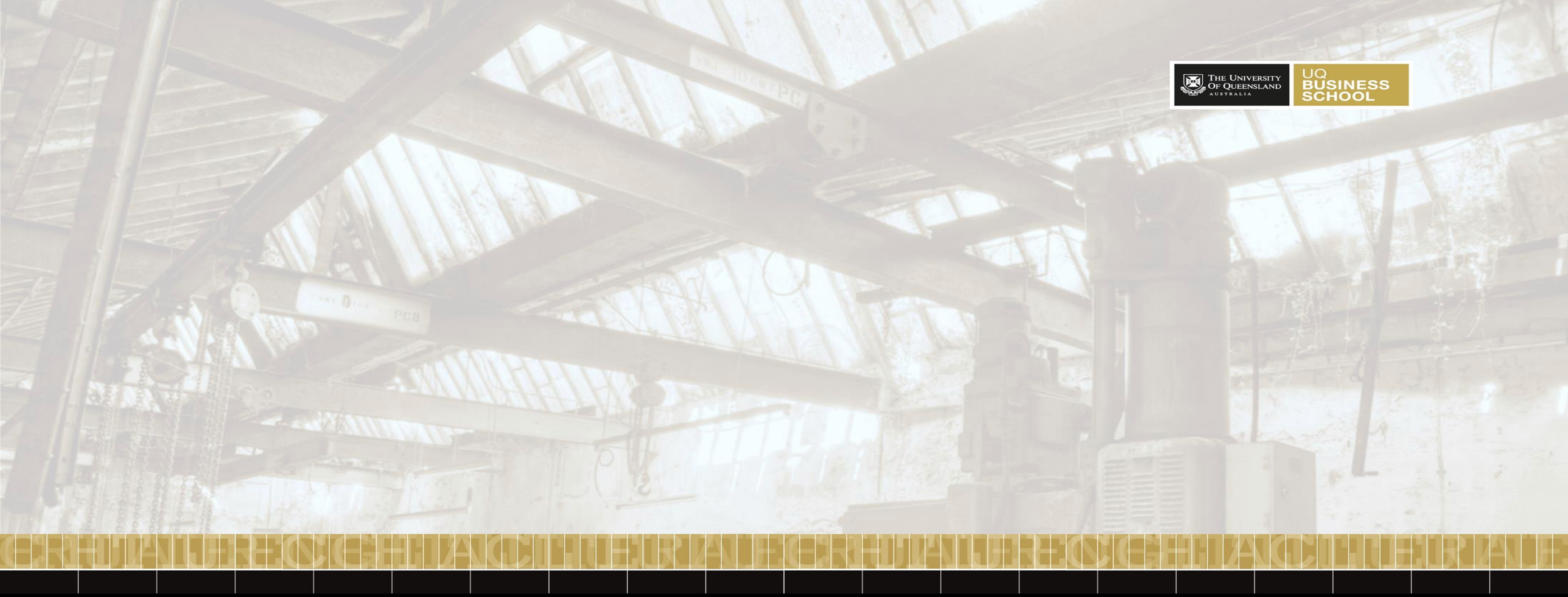
Recap: Business Analytics Framework



Learning Objectives

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

- Briefly explain the role of database management systems in transaction processing.
- Interpret and analyze a basic Entity Relationship Diagram (ERD) and validate its accuracy with respect to the description of a specific business process.
- Explain and identify database anomalies and motivations for normalization
- Write basic SQL queries



Data Modelling and Relational Databases

Organisational Memory

- Organisations need to remember things (or entities):
  - Customers, Employees, Products, Stores, Suppliers

- Question 1: What do universities need to remember?

- Question 2: Where do universities store the data?

A Table…

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
...	...	...	...	...

What does a Table Represent?

What does a Table Represent?

- A thing (or an entity)
- Columns represent attributes of an entity
- Rows represent instances of the entity
  - Records

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
...	...	...	...	...

More Entities (More Tables)…

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
...	...	...	...	...

Class Table

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

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
A126	ECO101	1-11	H2B
A126	ECO104	1-11	H2B
A126	ACC103	1-11	H2A
...	...	...	...

What is missing?

Relationships between Entities (or Tables)

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
...	...	..	...	...

Class Table

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

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
A126	ECO101	1-11	H2B
A126	ECO104	1-11	H2B
A126	ACC103	1-11	H2A
...	...	...	...

Organisational Memory- Continued

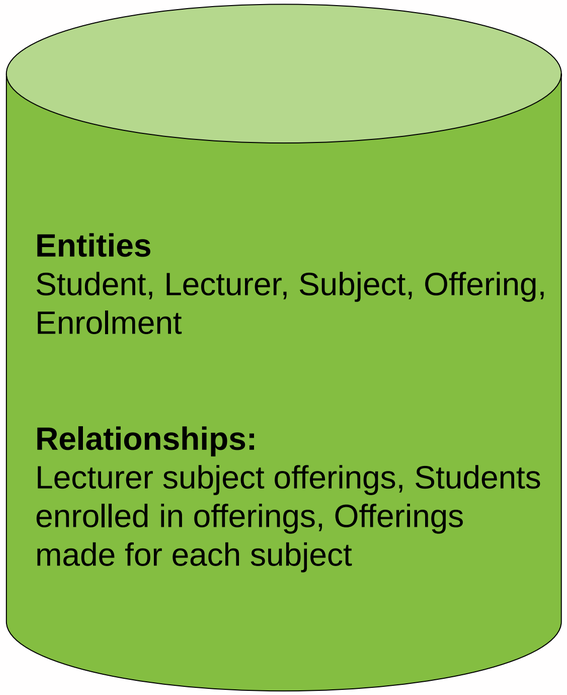
- Organisations also need to remember relationship between things (or entities):
  - What offerings are available for a subject in a given academic period?
  - Who is the lecturer for an offering of a subject?
  - What students are enrolled in an offering of a subject?

Relational Database

- Collection of tables and relationships between them

- A Data Base Management System (DBMS):
  - A software application with which you can create, store, organise and retrieve data from one or many databases
  - A Query Language (eg Structured Query Language – SQL or Sequel)
  - In this course, you will learn to write queries in PostgreSQL

University Database:
A Relational Database



But how do we specify the entities and relationships?

What is Data Modelling?

- Data modelling is a technique for determining
  - What data
  - What relationships
- Should be stored in a database
- We use Data Models to conceptually design our data needs
  - Allow us to describe data, relationships, data semantics and data constraints
  - Powerful tool for expressing information requirements (business rules and business processes)

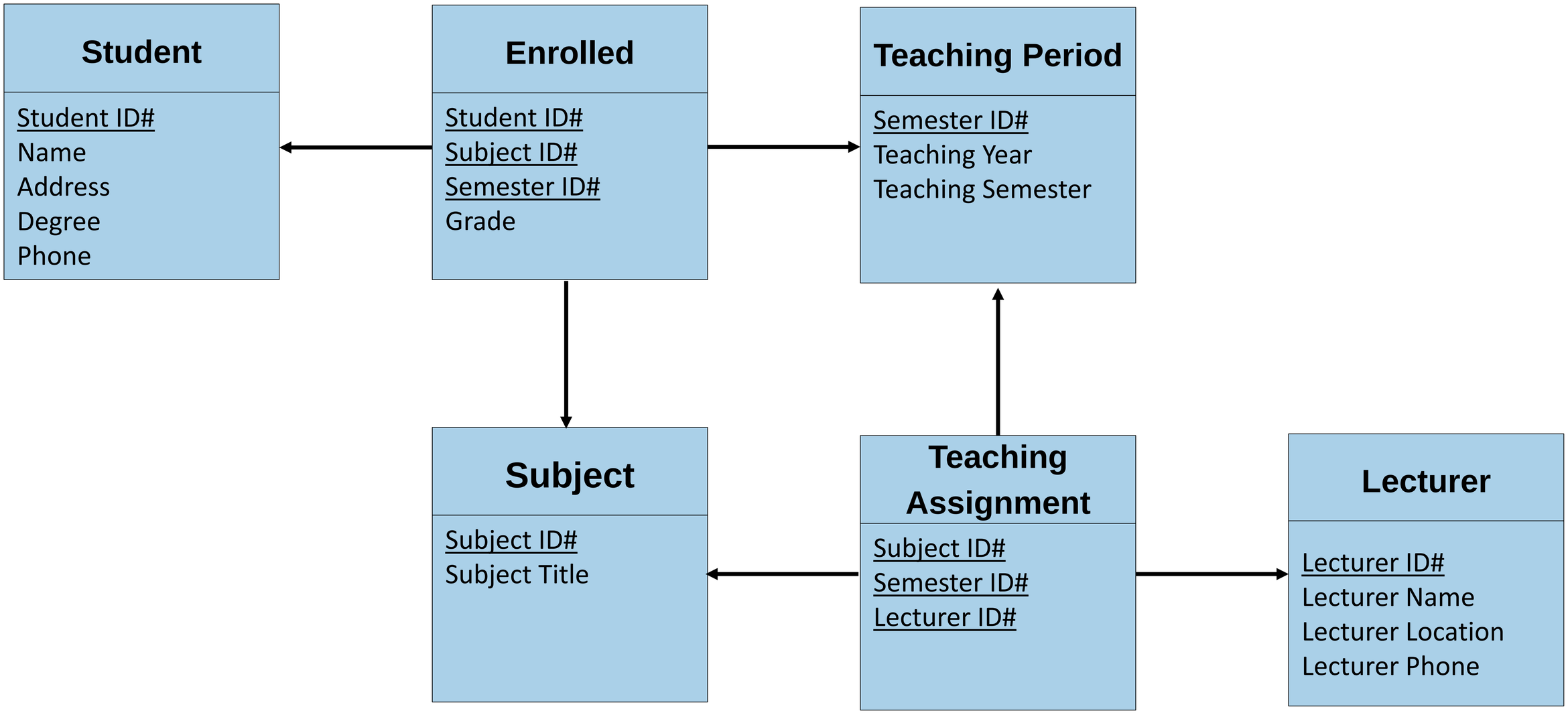
- In this subject we use
  - Entity Relationship Diagrams (for databases)
  - Star Schemas (for DW- next week)

Entity Relationship (ER) Diagram

- What is it?
  - A semantic, graphical data model.
  - Picture of the people, places, objects, things, events, or concepts, their characteristics and relationships, for an organisation or business area.

- Why do we care?
  - It visually expresses business rules.
  - A technology independent communication tool between business people and IT people.

ER Diagram- Example

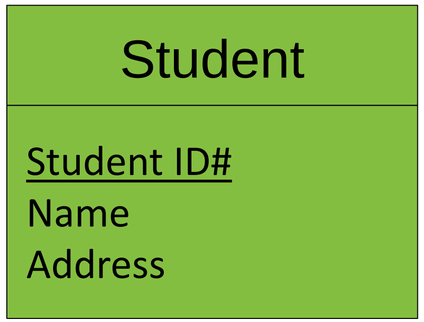


ER Diagram Building Blocks

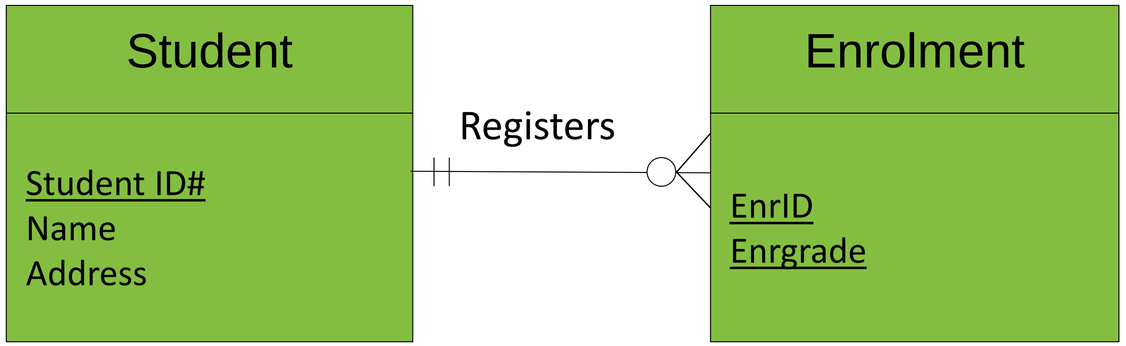
- Entities
  - Person, place, object, event, or concept (Nouns)
  - Things about which we wish to collect data
  - Entity type vs. Entity instance
    - (e.g., Lecturer vs. Michael Davern)
- Attributes
  - Characteristics descriptive of an entity.
- Relationships
  - Association between entities
  - (Verbs)
  - Directional

Notation

- Entities:



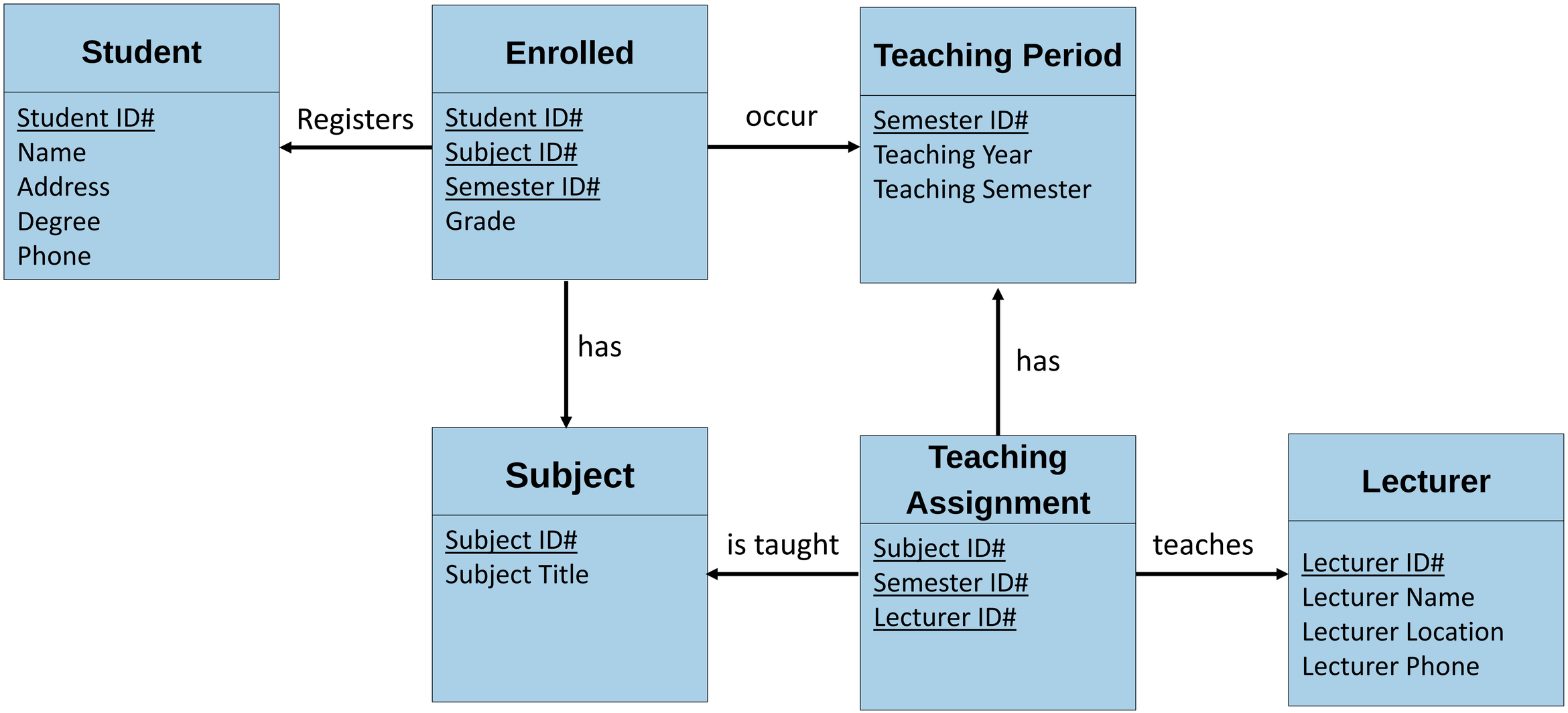
- Relationships:



Identifying Business Rules and Entities

- Business rules are statements that define or constrain aspects of the business
- They can impact the structure and behaviour of a database
- (amongst all the other organisation systems)
- We usually express business rules as terms or facts!
  - A customer sets up at least one account (fact)
  - A customer (term)
  - Account (term)
- Entities can often by identified by the “terms” (or nouns) and relationships as “verbs”

ER Diagram- Example

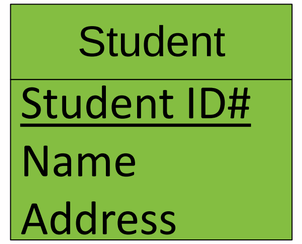


ER Process: How to Develop ER Models?

- Identify entity types
  - Person, Place, Object, Event, Concept …
  - Define attributes and primary key
- Identify relationships
  - Connect entity types that are related by a natural linkage or event occurrence
- Identify constraints
  - Include relationship cardinalities
- Iterate!

Identifying the Attributes

- Attributes: properties or characteristics of entities.
- Attributes that uniquely identify an entity instance (i.e., row, record) are “candidate” keys.
- Primary keys are the attributes we choose as the unique identifier we will use.
- How to choose:
  - should not change over time (age)
  - must have unique, non-null values
  - use as few attributes as possible
- Typically underlined.



“Key” Terminology

- A Primary Key (or key) is an attribute, or group of attributes, that uniquely identifies a row in a relation.
  - Every record must have a key.
  - Often numeric.
- A Foreign Key is a non-key attribute in one relation that appears as the primary key in another relation.
- A Composite Key is a key that consists of two or more attributes.

Integrity Rules: Safeguarding Keys

Database Management Systems enforce two integrity rules:

1. Entity Integrity- Makes entities traceable
   - Every table has column with unique values
   - No duplicates or blanks allowed for primary key

2. Referential Integrity- Ensures validity of relationships
   - The value entered in a foreign key attribute, must exist as a value in the corresponding relation's primary key
     - A student can't register for a course unless they already have a record in the student relation.
     - Can't remove a student from the student relation if they are currently registered for a course.

ER Review: Purpose

- To understand and describe data requirements (or assets)
- To communicate these requirements between various stakeholders
  - both IT people and business people
- To form a basis for database and IT system design

Quality of ER Models

1. Correct:
   - Conforms to ER syntax rules
2. Complete
   - Contains all required information
3. Simple
   - Contains minimum number of possible entities and relationships
4. Understandable
   - Concepts in model are easy to understand

Quality of ER Models- Continued

5. Flexible
   - Ease with which model copes with change
6. Integrity
   - Contains all required “business rules”
7. Integration
   - Consistency with other organisational data
8. Implementability
   - Ease with which model can be implemented

Normalization

What's wrong with the *organisation* of data in this table?

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
...	...	...	...	...	...	...	...	...	...	...	...

“Normalized” 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
...	...	...	...	...

Class Table

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

Teaching Assignment

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
...	...	...

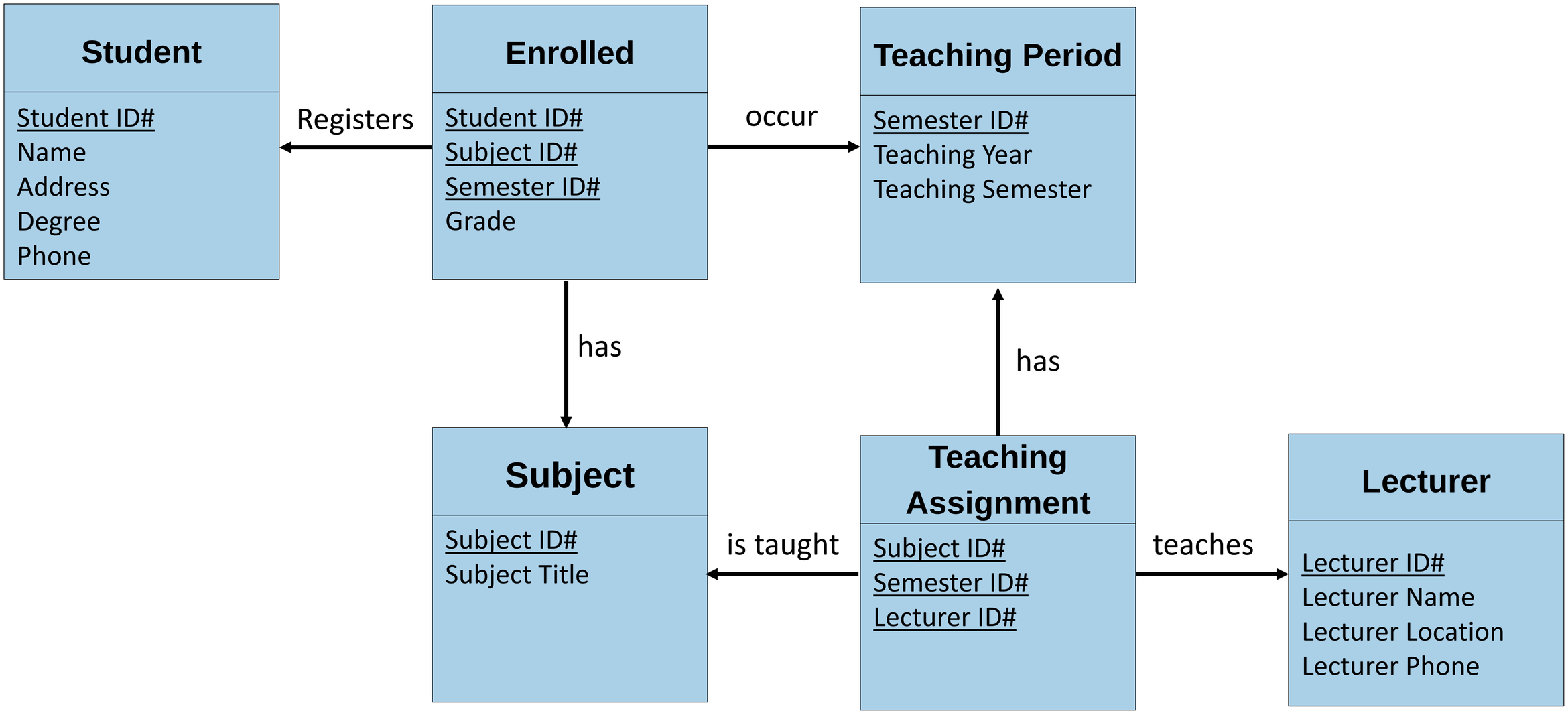
Lecturer Table

Lecturer Name	Lecturer Location	Lecturer Phone
Collier	T240D	8344-5716
	T240E	8344-5309

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
A126	ECO101	1-11	H2B
A126	ECO104	1-11	H2B
A126	ACC103	1-11	H2A
...	...	...	...

ER Diagram- Example



Anomalies in Unnormalised Data

- Consider the following unnormalised table (relation) :

Student-ID	Course-ID	Fee
130	C200	75
200	C300	100
250	C200	75
425	C400	150
500	C300	100
575	C500	50
…	…	…

- ***Insertion Anomaly****:* A new course cannot be added until at least one student has enrolled (which comes first student or course?)
- ***Deletion Anomaly****:* If student 425 withdraws, we lose all record of course C400 and its fee!
- ***Update Anomaly****:* If the fee for course C200 changes, we have to change it in multiple records (rows), else the data will be inconsistent.

Your turn: ER Exercise

Bill To

John
Synex Inc
128 AA Juanita Ave
Glendora
CA 91740 US

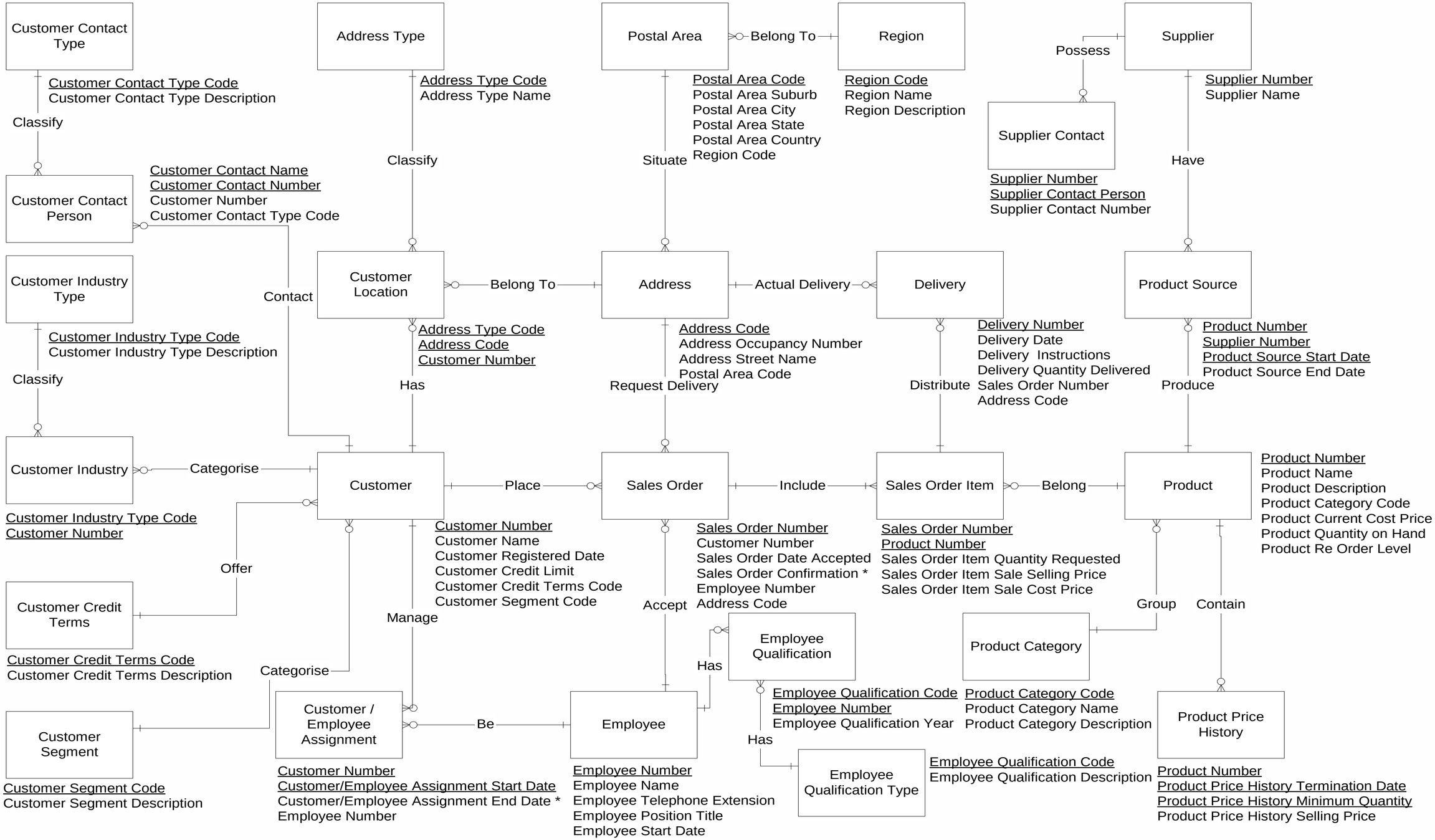
Ship To

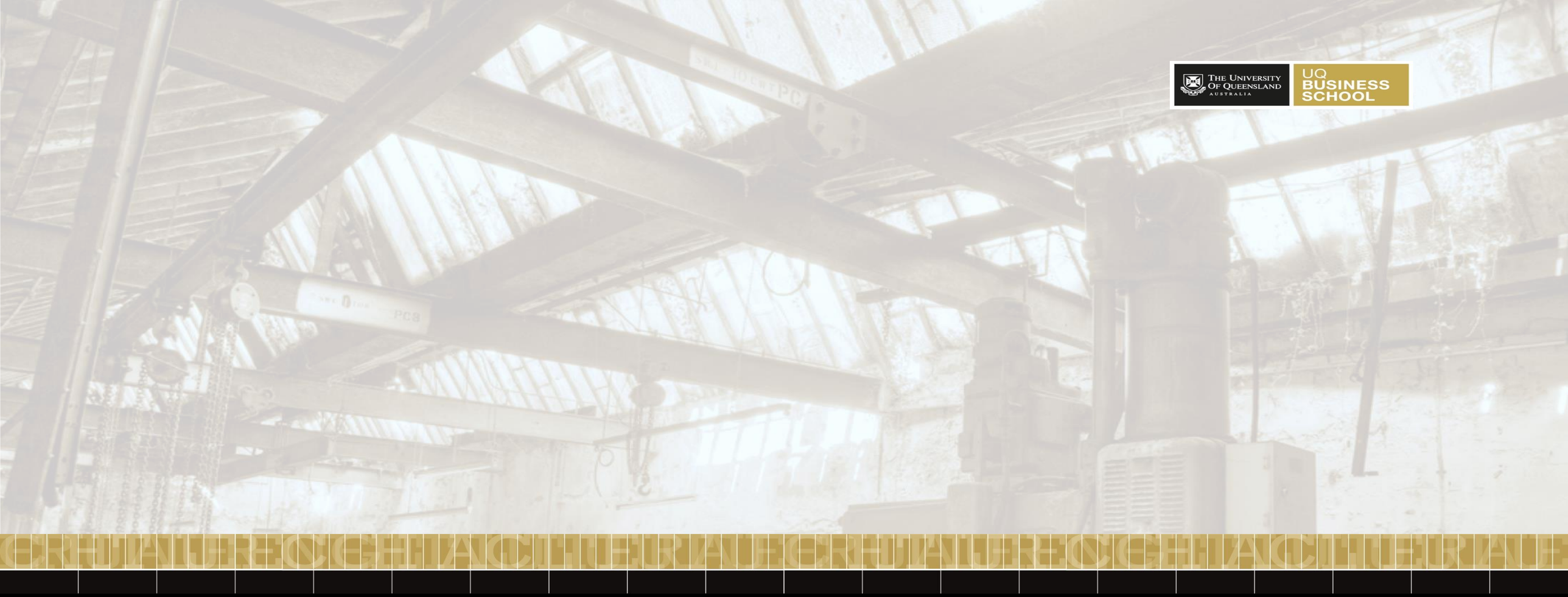
John
Synex Inc
128 AA Juanita Ave
Glendora
CA 91740 US

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.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

Hint: Take the ER model development process?


Customer Contact Type
Customer Contact Type Code
Customer Contact Type Description
Classify
Customer Contact Person
Customer Contact Name
Customer Contact Number
Customer Number
Customer Contact Type Code
Customer Industry Type
Customer Industry Type Code
Customer Industry Type Description
Classify
Customer Industry
Categorise
Customer Industry Type Code
Customer Number
Contact
Address Type
Address Type Code
Address Type Name
Classify
Customer Location
Belong To
Address Type Code
Address Code
Customer Number
Has
Customer
Place
Customer Number
Customer Name
Customer Registered Date
Customer Credit Limit
Customer Credit Terms Code
Customer Segment Code
Manage
Offer
Customer Credit Terms
Customer Credit Terms Code
Customer Credit Terms Description
Categorise
Customer Segment
Customer Segment Code
Customer Segment Description
Customer / Employee Assignment
Be
Customer Number
Customer/Employee Assignment Start Date
Customer/Employee Assignment End Date *
Employee Number
Postal Area
Belong To
Postal Area Code
Postal Area Suburb
Postal Area City
Postal Area State
Postal Area Country
Region Code
Situate
Address
Address Code
Address Occupancy Number
Address Street Name
Postal Area Code
Request Delivery
Sales Order
Sales Order Number
Customer Number
Sales Order Date Accepted
Sales Order Confirmation *
Employee Number
Address Code
Accept
Include
Region
Region Code
Region Name
Region Description
Actual Delivery
Delivery
Delivery Number
Delivery Date
Delivery Instructions
Delivery Quantity Delivered
Sales Order Number
Address Code
Distribute
Sales Order Item
Sales Order Number
Product Number
Sales Order Item Quantity Requested
Sales Order Item Sale Selling Price
Sales Order Item Sale Cost Price
Belong
Possess
Supplier
Supplier Number
Supplier Name
Supplier Contact
Supplier Number
Supplier Contact Person
Supplier Contact Number
Have
Product Source
Product Number
Supplier Number
Product Source Start Date
Product Source End Date
Produce
Product
Product Number
Product Name
Product Description
Product Category Code
Product Current Cost Price
Product Quantity on Hand
Product Re Order Level
Group
Contain
Product Category
Product Category Code
Product Category Name
Product Category Description
Product Price History
Product Number
Product Price History Termination Date
Product Price History Minimum Quantity
Product Price History Selling Price
Employee
Employee Number
Employee Name
Employee Telephone Extension
Employee Position Title
Employee Start Date
Has
Employee Qualification
Employee Qualification Code
Employee Number
Employee Qualification Year
Has
Employee Qualification Type
Employee Qualification Code
Employee Qualification Description



Querying a Database (Introduction to SQL)

What is SQL?

- ***S***tructured ***Q***uery ***L***anguage
- A high-level declarative programming language for
- accessing databases
- Highly optimised for manipulating data
- Multiple similar standards - every vendor is slightly different

Four Primary Operations of SQL:

Operation	SQL Command
Create	INSERT
Read	SELECT
Update	UPDATE
Delete	DELETE

SQL Overview

- `CREATE TABLE <name> ( <field> <domain>, … )`

- ```
  INSERT INTO <name> (<field names>)
       VALUES (<field values>)
  ```

- ```
  DELETE FROM <name>
         WHERE <condition>
  ```

- ```
  UPDATE <name>
     SET <field name> = <value>
   WHERE <condition>
  ```

- ```
  SELECT <fields>
    FROM <name>
   WHERE <condition>
  ```

SELECT statement

SELECT [ALL | DISTINCT] ***select_expr*** **[,** ***select_expr*** **...]**

List the columns (and expressions) that are returned from the query

[FROM ***table_references***

Indicate the table(s) or view(s) from where the data is obtained

[WHERE ***where_condition*****]**

Indicate the conditions on whether a particular row will be in the result

GROUP BY {*col_name* **|** ***expr*** **} [ASC | DESC], ...]**

Indicate categorisation of results

HAVING ***where_condition*****]**

Indicate the conditions under which a particular category (group) is included in the result

ORDER BY {*col_name* **|** ***expr*** **|** ***position*****} [ASC | DESC], ...]**

Sort the result based on the criteria

[LIMIT {[*offset*****,]** ***row_count*** **|** ***row_count*** **OFFSET** ***offset*****}]**

Limit which rows are returned by their return order (ie 5 rows, 5 rows from row 2)**]**

An SQL Primer : SELECT

In order to get data from the database (from a table) we send a SQL command **SELECT** to the database

The simplest query takes the form:

select column(s)

from table(s)

where condition(s)

SELECT Example 1

Customer	
PK	**CustomerID**
	CustFirstName CustMiddleName **CustLastName** BusinessName **CustType**

SQL →

```
SELECT CustLastName FROM Customer;
```

RESULT →

CustLastName
Smith
Jones
Smithies
Smythe
Smart
Lam
Unila
Jones
Samson

The names of the attributes that we want data from in the table

The TABLE (name) we want to query

SELECT Example 2

Customer	
PK	**CustomerID**
	CustFirstName CustMiddleName **CustLastName** BusinessName **CustType**

SQL →

```
SELECT * FROM Customer;
```

RESULT →

Edit: Export: Autosize:

CustomerID	CustFirstName	CustMiddleName	CustLastName	BusinessName	CustType
1	Peter	NULL	Smith	NULL	Personal
2	James	NULL	Jones	JJ Enterprises	Company
3	Akin	NULL	Smithies	Bay Wart	Company
4	Julie	Anne	Smythe	Konks	Company
5	Jen	NULL	Smart	BRU	Company
6	Lim	NULL	Lam	NULL	Personal
7	Kim	NULL	Unila	Saps	Company
8	James	Jay	Jones	JJ's	Company
9	Keith	NULL	Samson	NULL	Personal
NULL	NULL	NULL	NULL	NULL	NULL

SELECT Example 3

Customer	
PK	**CustomerID**
	CustFirstName CustMiddleName **CustLastName** BusinessName **CustType**

SQL →

```
SELECT CustLastName, CustFirstName
    FROM Customer;
```

RESULT →

Export: Autosize:

CustLastName	CustFirstName
Smith	Peter
Jones	James
Smithies	Akin
Smythe	Julie
Smart	Jen
Lam	Lim
Unila	Kim
Jones	James
Samson	Keith

An SQL Primer: SELECT WHERE

The result of each SELECT statement so far has included every row in the table (for the specified attribute). WHERE clause filters unwanted rows from the result.

Customer	
PK	**CustomerID**
	CustFirstName CustMiddleName **CustLastName** BusinessName **CustType**

SQL →

```
SELECT CustLastName FROM Customer
    WHERE CustLastName = "Smith";
```

RESULT →

CustLastName
Smith

An SQL Primer: GROUP BY

Aggregating data by particular attribute

Customer	
PK	**CustomerID**
	CustFirstName CustMiddleName **CustLastName** BusinessName **CustType**

SQL →

```
SELECT CustType, Count(CustomerID)
    FROM Customer
    GROUP BY CustType;
```

RESULT →

Export: Autosize:

CustType	Count(CustomerID)
Personal	3
Company	6

Logic: Count (Customer ID) will return the number of customers,
Group BY CustType will group the result based on CustType

An SQL Primer: GROUP BY and ORDER BY

Customer	
PK	**CustomerID**
	CustFirstName CustMiddleName **CustLastName** BusinessName **CustType**

SQL →

```
SELECT CustLastName, CustType
    FROM Customer
    ORDER BY CustLastName;
```

RESULT →

Export

CustLastName	CustType
Jones	Company
Jones	Company
Lam	Personal
Samson	Personal
Smart	Company
Smith	Personal
Smithies	Company
Smythe	Company
Unila	Company

```
SELECT CustLastName, CustType
    FROM Customer
    ORDER BY CustLastName DESC;
```

Export Au

CustLastName	CustType
Unila	Company
Smythe	Company
Smithies	Company
Smith	Personal
Smart	Company
Samson	Personal
Lam	Personal
Jones	Company
Jones	Company

An SQL Primer: LIMIT rows

Customer	
PK	**CustomerID**
	CustFirstName CustMiddleName **CustLastName** BusinessName **CustType**

SQL →

```
SELECT CustLastName, CustType
    FROM Customer
    ORDER BY CustLastName
    LIMIT 5;
```

RESULT →

Export

CustLastName	CustType
Jones	Company
Jones	Company
Lam	Personal
Samson	Personal
Smart	Company

CustLastName	CustType
Jones	Company
Jones	Company
Lam	Personal
Samson	Personal
Smart	Company
Smith	Personal
Smithies	Company
Smythe	Company
Unila	Company

A SQL Primer : JOINS

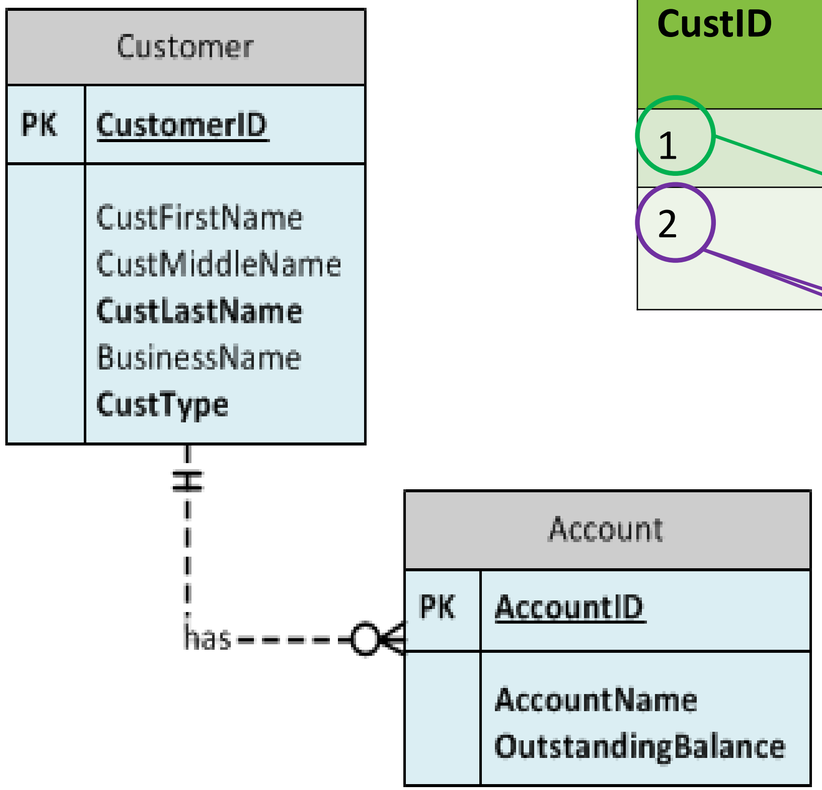
A JOIN statement create a 'virtual' table which displays the fields from both tables under a condition.

The join condition tells the database how it should match the records from one table to the other.

- What fields should it use
- What should happen if records are found in one table and not the other

More than One Entity

- We looked at Customer
  - A customer can have a number of Accounts
  - The tables get linked through a foreign key



CustID	CustomerFirstName	CustMiddleName	CustLastName	BusinessName	CustType
1	Peter		Smith		Personal
2	James		Jones	JJ Enterprises	Company

AccountID	AccountName	OutstandingBalance	CustID
01	Peter Smith	245.25	1
05	JJ Ent.	552.39	2
06	JJ Ent. Mgr	10.25	2

SQL Joins – Natural JOIN

- Natural Join: Join the tables with foreign keys where the primary key and foreign key have the same name

SQL →

```
SELECT * FROM Customer NATURAL JOIN Account;
```

RESULT →

Export: Autosize:

CustomerID	CustFirstName	CustMiddleName	CustLastName	BusinessName	CustType	AccountID	AccountName	OutstandingBalance
1	Peter	NULL	Smith	NULL	Personal	1	Peter Smith	245.25
2	James	NULL	Jones	JJ Enterprises	Company	2	JJ ENt.	552.39
2	James	NULL	Jones	JJ Enterprises	Company	3	JJ ENt. Mgr	10.25

SQL Joins – Inner JOIN

- Inner Join the tables with foreign keys!

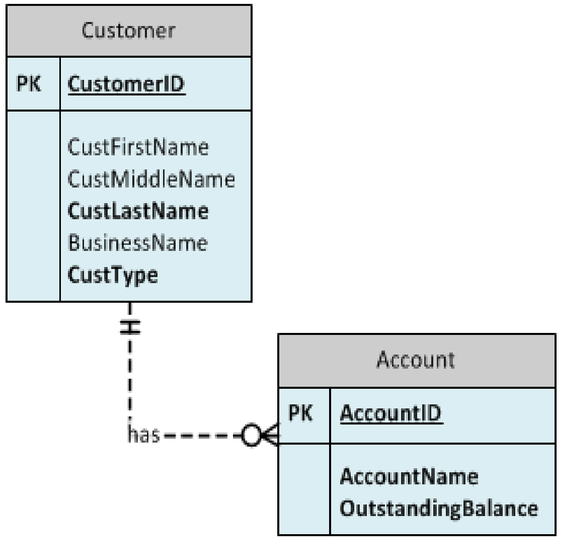
SQL →

```
SELECT * FROM Customer INNER JOIN Account
    ON Customer.CustomerID = Account.CustomerID;
```

RESULT →

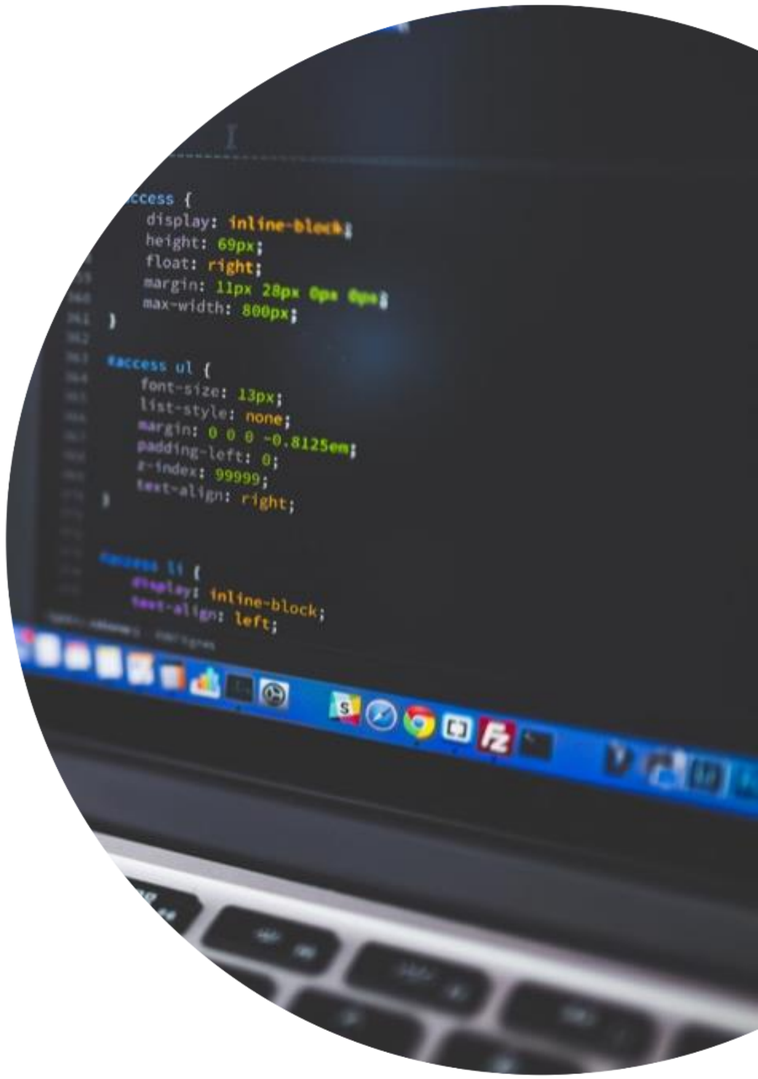
Export: Autosize:

CustomerID	CustFirstName	CustMiddleName	CustLastName	BusinessName	CustType	AccountID	AccountName	OutstandingBalance	CustomerID
1	Peter	NULL	Smith	NULL	Personal	1	Peter Smith	245.25	1
2	James	NULL	Jones	JJ Enterprises	Company	2	JJ ENt.	552.39	2
2	James	NULL	Jones	JJ Enterprises	Company	3	JJ ENt. Mgr	10.25	2



Summary of what we learned

- Role of databases in transaction processing
- ER Models and Normalisation
- SQL for querying a database



What is Examinable:

- This lecture was the backbone for learning about business analytics
- You will not be asked to develop an ER model
- You need to analyse and interpret ER models including entities, attributes, different types of keys and relationships
- You need to explain SQL SELECT, Aggregate functions and JOIN Statements

Next Seminar

Next Seminar

- Dimensional Modelling