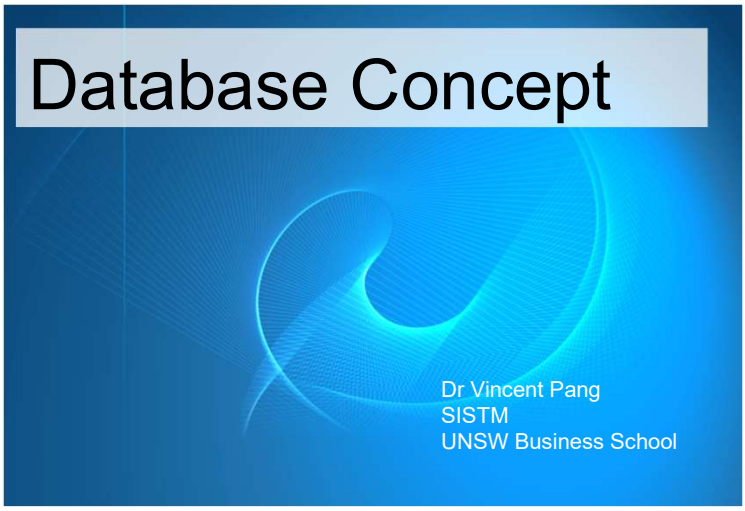
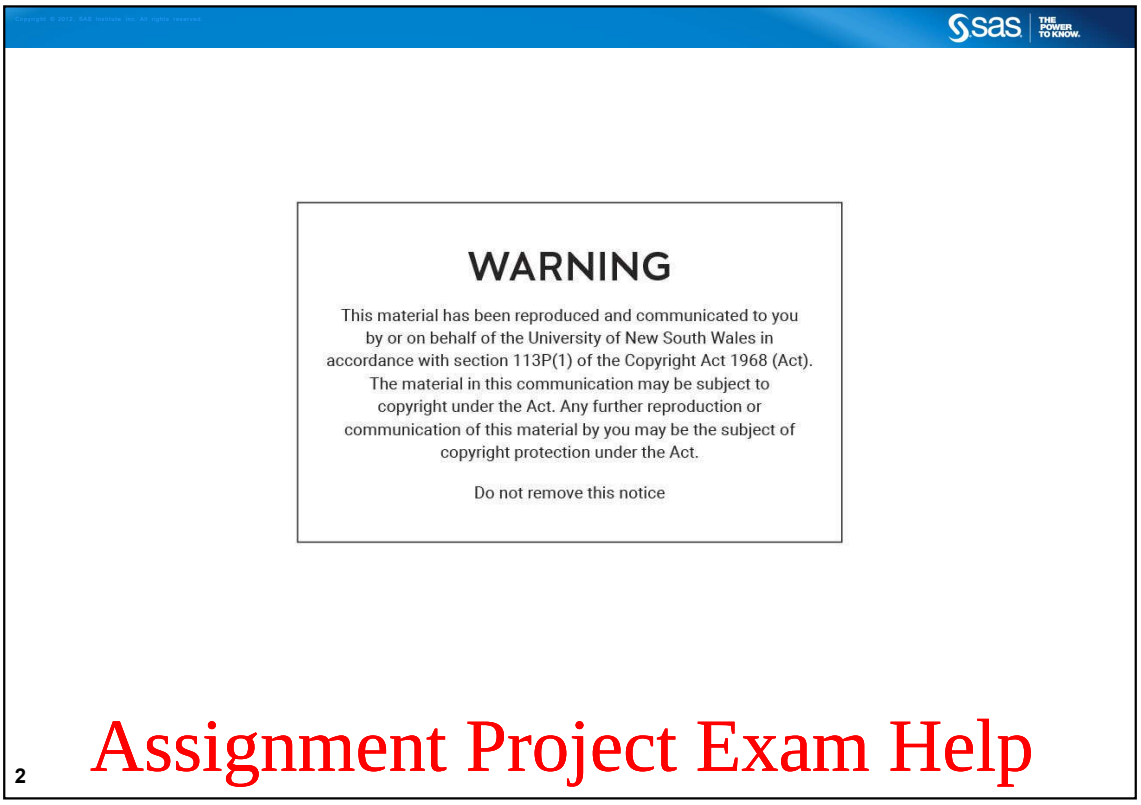
sas THE POWER TO KNOW.

INFS5710 IT Infra. for BA



1



sas THE POWER TO KNOW.

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.**

UNSW SYDNEY

3

SAS THE POWER TO KNOW.

Country

Acknowledgement of Country

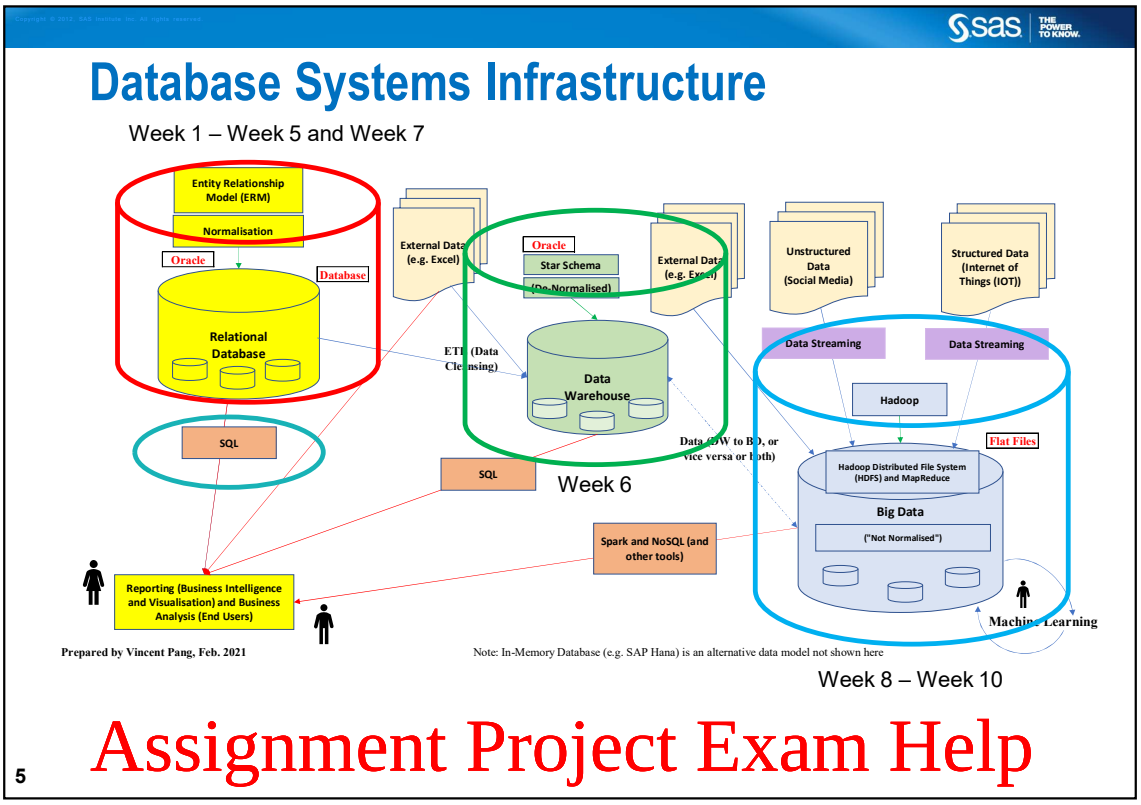
UNSW Business School acknowledges the Bidjigal (Kensington campus) and Gadigal (City campus) the traditional custodians of the lands where each campus is located.

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. (2021, July 24). *Acknowledgement of Country* [online video]. Retrieved from https://vimeo.com/369229957/d995d8087f

4



This diagram covers what we are going to do in the next ten weeks: data infrastructure, data warehouse and big data. We cannot cover everything in this course, and we can only touch the surface of infrastructure as shown.

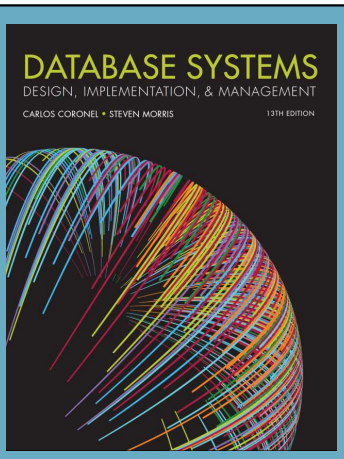
Data infrastructure is critical because this is how you will control how the data are captured and control flow of the data.

You will learn about normalisation design of the database using Entity-Relationship Diagram or ERD. In the lab, you will learn to draw and use Oracle to create an ERD.

In the lab, you will learn SQL or Sequel. You will use SAS because it can handle a large dataset without any issues.

We will touch on how you can design a Data Warehouse using Stars schema in Week 6.

In the last few weeks, we will talk about big data system management and its infrastructure. We will briefly discuss Hadoop, MapReduce, and Sparks behind the building of the big data. If time permits, we will look at AWS as well.



Chapter 1

Database Systems

1-1 to 1-7 (pp. 3 to 28)

We will follow the textbook very closely.

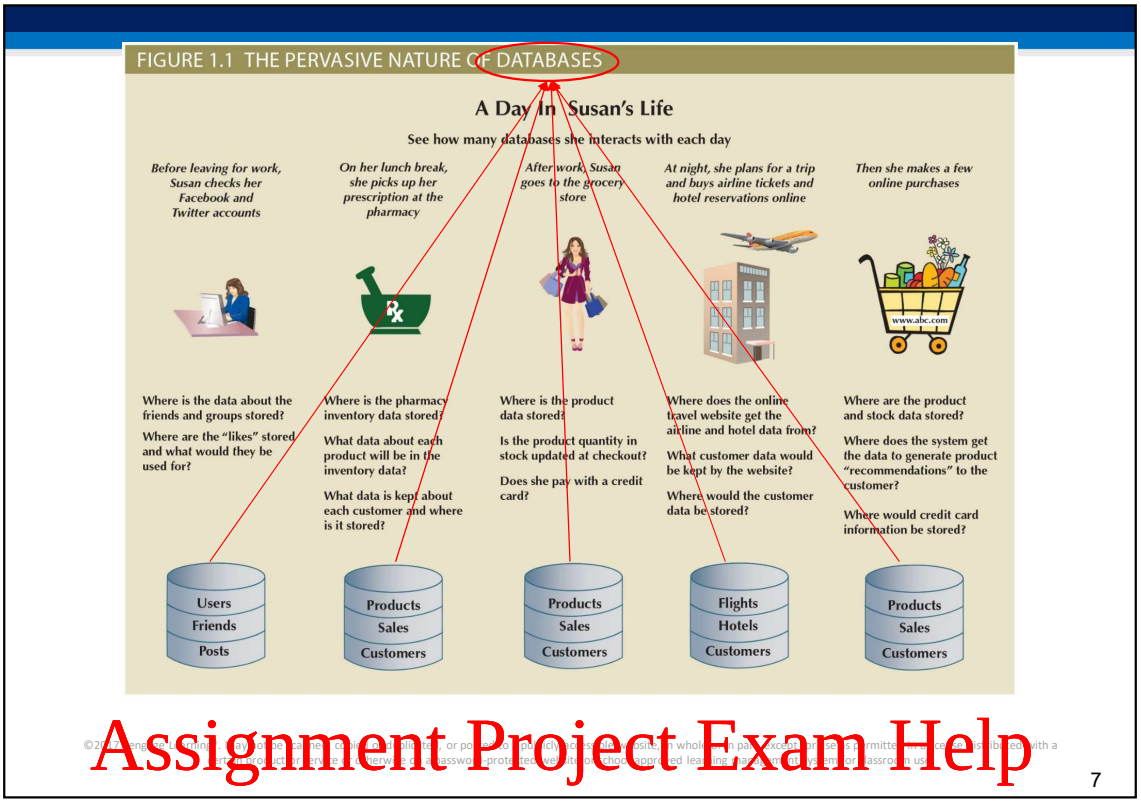
As you have paid over A$60 for the textbook, you might as well make full use of it. The reason we use a textbook is some students in the past wanted something they can refer to.

The textbook covers the basic concepts in a database system. I will add more materials for big data such as Hadoop and MapReduce as there are not enough materials in the textbook on this topic.

The first two chapters are related to database concepts. We will cover both chapters today but there are plenty to read, and you need to read them at home.

I will give you the references what you need to read (usually most of the chapter). There will be MCQs (Multiple-choice questions choice questions) and probably short questions in based on the textbook for individual assessments.

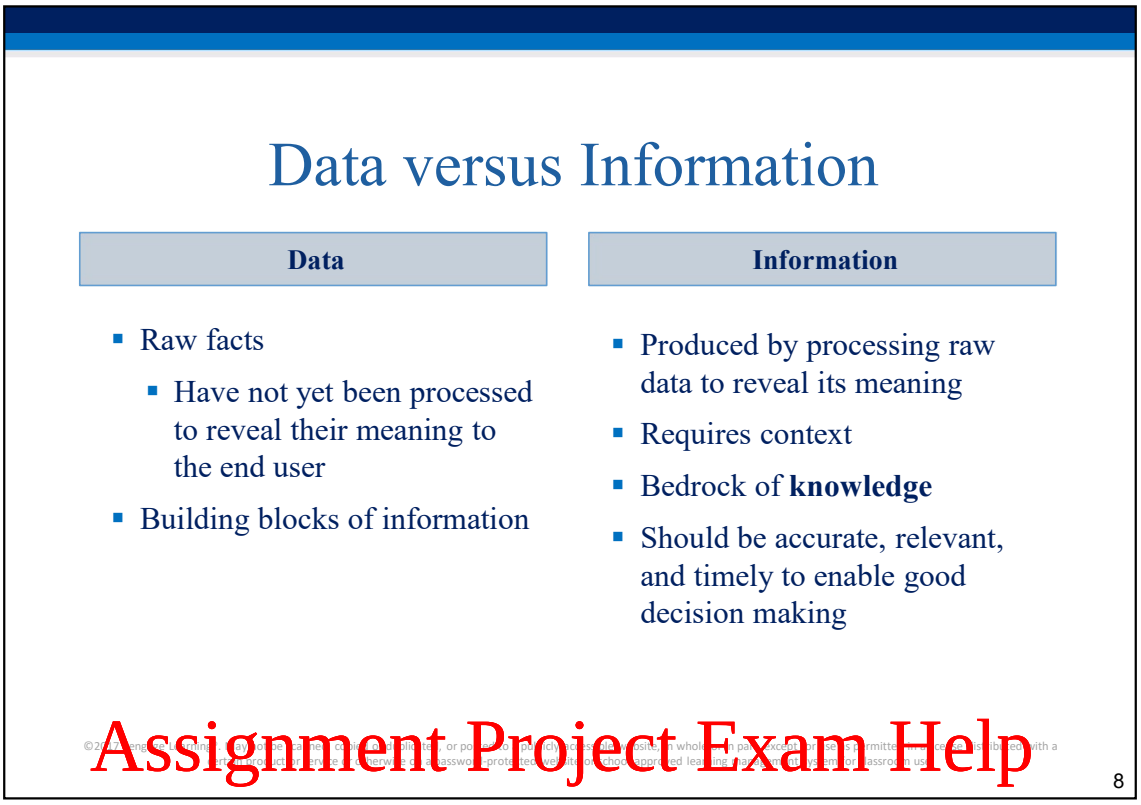
We cannot cover everything in class, so you have read it yourself at home, or may be when you cannot sleep □ As we just have a one-hour lecture, I can only cover the basic concepts. I will skip some slides due to time, but the materials and speaker notes are all enclosed.



So, where do you find database? Today, data is everywhere. It is not limited to numbers; you can also have text, or even taking a photo is data.

If you look at a day in Susan's life, she spends time using social media such as Facebook, Instagram, Tik Tok, Twitter, WeChat, QQ, WhatsApp, and so on. It will create gigabyte of data over times. It can include text, pictures, videos and so on. After work, when she goes to do some shopping, data will be captured by the supermarket, and credit card she uses. The supermarket and bank will know where, when, and how she spends her money. The marketing people would like to know someone like Susan where and what she spends her money on, so they if she is the right person to target her for a new products and promotions in the future.

All the databases have their purposes.

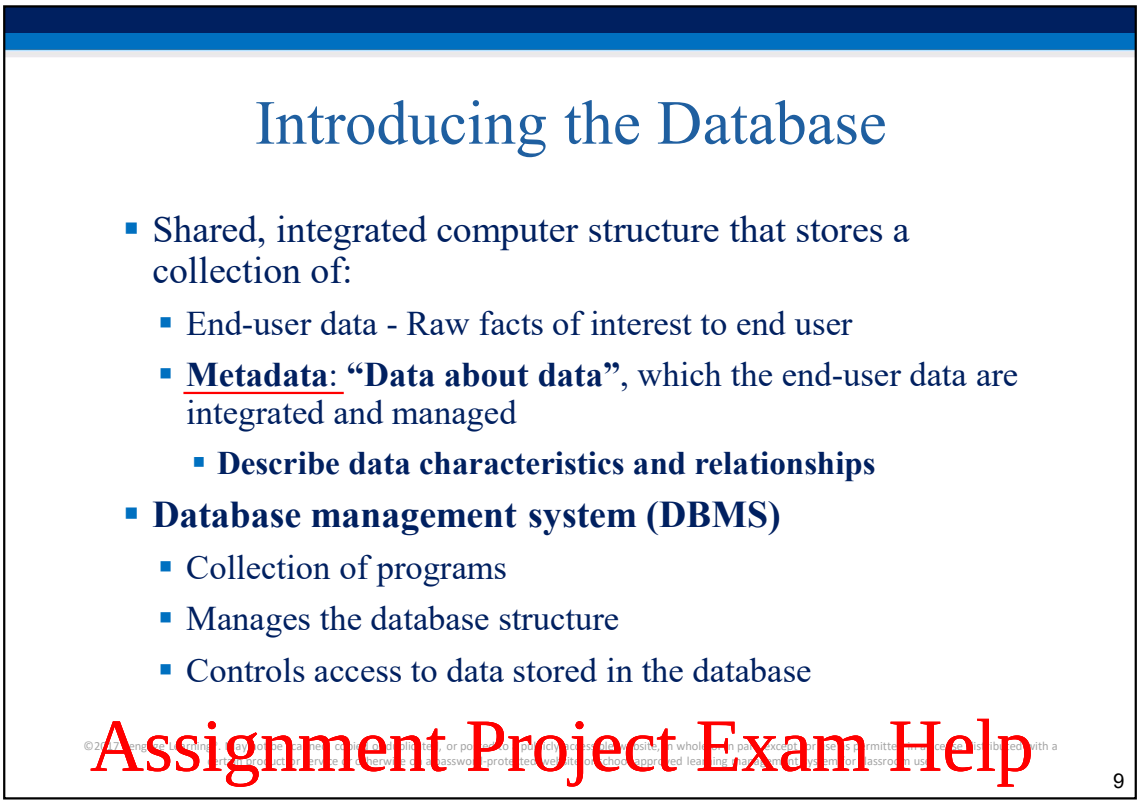


Data consists of raw facts whereas information is the result from processing raw data to reveal its meaning. Information creates knowledge and it is the foundation of decision making.

For example, data can be your marks for your quiz, marks for your assignment, and so on. For your other course such as e-commerce, you also have marks for quiz, participation, exam and so on. These are all data. We then give you a grade based on your mark for each of the courses you have done.

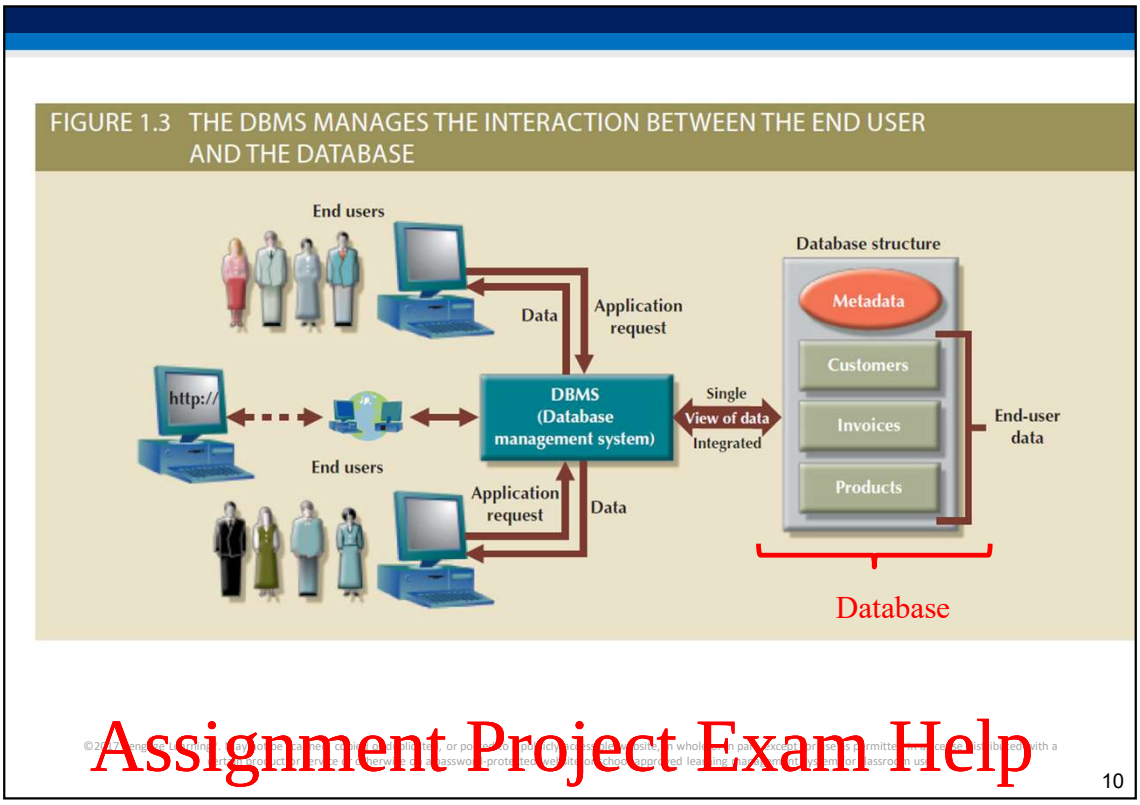
Information is when you put everything together in a report or a certificate, such as putting all your grades together and average the marks for the courses such as your WAM. The WAM and grades will tell you and your potential employer how you did at UNSW.

As for the point of decision making, using the example of your grades. When you apply for a job, the potential employer will look at your certificate compare with other potential candidates to decide if they will employ you along with your interview and so on.



Metadata is "data about data", that is you describe data characteristics and relationships. This is important when you receive a set of data, you want to know what the numbers mean. If not, you will have to ask what the numbers mean. For instance, you have a few dates on a table or in a Spreadsheet, you want to know what each of dates means. When you do analysis, you have to use the correct dates.

Database Management System or DBMS is the management of data of how and where the data are stored, how the data can be retrieved, and who can retrieve the data.



This is the first infrastructure you encounter in this course.

Infrastructure is important for data analytics because it is the foundation for storing and managing data. The infrastructure can be viewed as physical as well as logical. For instance, looking at the diagram above, a PC can logically be represented as a normal PC, a laptop or even a tablet or a mobile phone.

This is the infrastructure of a normal database system, i.e., the DBMS system, which manages the interaction between the users on the left-hand side, and the database on the right-hand side.

So how does it work? An end user on the left-hand side wants to find who bought this new product such as a new Coca Cola Vegemite flavour? Vegemite is a popular spread for kids in Australia if you don't know what it is.

A request is sent to DBMS (Database Management Systems), which, in turn, processes the request and send a request using *Structured Query Language* or *SQL* to retrieve all the customers who bought this new Coca Cola *Vegemite* flavour. You will start learning SQL language in this week's workshop.

Once the data is received, it will send back to DBMS which then processes and send back to the end user. This usually becomes information. The users might use the information in their decision making.

in the textbook, it talks about different types of databases. It covers single user database, and multi-user database date. It also talks about structured data, and unstructured data. For more information, please read chapter 1 in your textbook!

Why Database Design is important?

FIGURE 1.4 EMPLOYEE SKILLS CERTIFICATION IN A POOR DESIGN

Why are there blanks in rows 9 and 10?

How to produce an alphabetical listing of employees?

How to count how many employees are certified in Basic Database Manipulation?

Is Basic Database Manipulation the same as Basic DB Manipulation?

What if an employee acquires a fourth certification?

Do we add another column?

ID	ENum	Name	Title	HireDate	Skill1	Skill1Date	Skill2	Skill2Date	Skill3	Skill3Date
1	02345	Brian Oates	DBA	2/14/1995	Basic Database Management	2/14/2002	Advanced Database Management	2/14/2005	Basic Web Design	8/9/2003
2	08273	Marco Bienz	Analyst	7/28/2006	Basic Web Design	3/8/2009	Advance Process Modeling	8/19/2012		
3	06234	Jasmine Patel	Programmer	8/10/2005	Basic Web Design	8/10/2007	Advanced C# programming	8/10/2007	Basic DB manipulation	1/29/2012
4	03373	Franklin Johnson, Jr.	Purchasing Agent	3/15/2002	Advanced Spreadsheets	6/20/2011				
5	13567	Almond, Robert	Analyst	9/30/2012	Basic Process Modeling	9/30/2014	Basic Database Design	5/23/2015		
6	10282	Richardson, Amanda	Clerk	4/11/2011						
7	09382	Susan Mathis	Database Programmer	8/2/2010	Basic DB Design	8/2/2012	Basic Database Manipulation	8/2/2012	Advanced DB Manipulation	5/1/2013
8	14311	Duong, Lee	Programmer	9/1/2014	Basic Web Design	9/1/2016				
9					Master Database Programming					
10					Basic Spreadsheets					
11	09002	Wade Gaither	Clerk	5/20/2010	Advanced Spreadsheets	5/16/2013	Basic Web Design	5/16/2013		
12	13383	Raymond F. Matthews	Programmer	3/12/2012	Basic C# Programming	3/12/2014				
13	09283	Chavez, Juan	Clerk	7/4/2010						
14	04893	Patricia Richards	DBA	6/11/2004	Advanced Database Management	6/11/2006	Advanced Database Manipulation	9/20/2012		
15	13932	Lee, Megan	Programmer	9/29/2013						

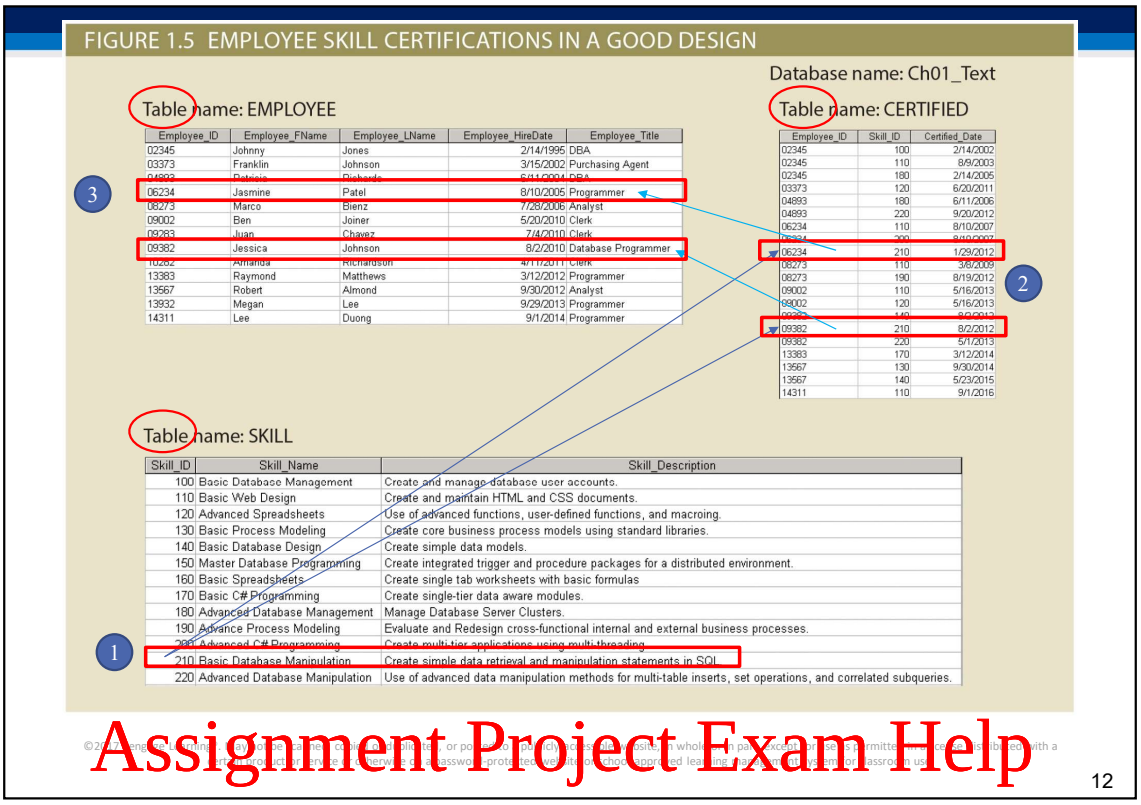
11

Why Database Design is important?

If you look at the diagram, what do you see? You see many numbers and text. You have to make sense what they mean. It might look very confusing for the first time.

Let's go into details - if you look at row #9 and row 10, why are they blank? Next, look at the Name column, do you need to separate into surname and first name or are they mixed? For example, I want to find every employees with surname "Smith", or surname "Zhang".

If I want to find all employees with skills for "Basic Database Manipulation", you have to go to column Skill1 to look for "Basic Database Manipulation", and then in column Skill2 for "Basic Database Manipulation", and then in column Skill3. **So, is there a better way to do this?**

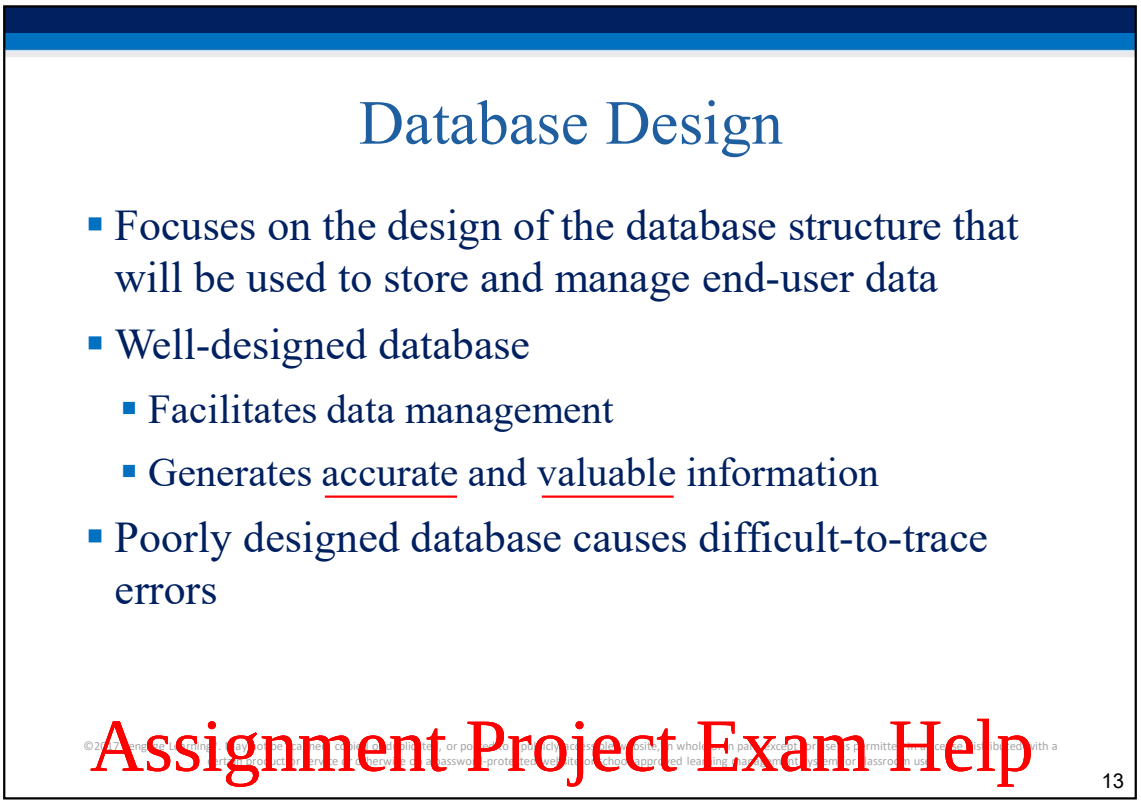


The slide shows what you will define as a good database design. You have three tables instead of one table. You will learn how to do this in ERD and Normalisation which will cover in the next few weeks.

Instead of having one gigantic table, we now have three smaller and different tables namely employee, skill, and certified. Now, if you want to find which employee who did the course Basic Database Manipulation, you can find all the employees much easier using these tables.

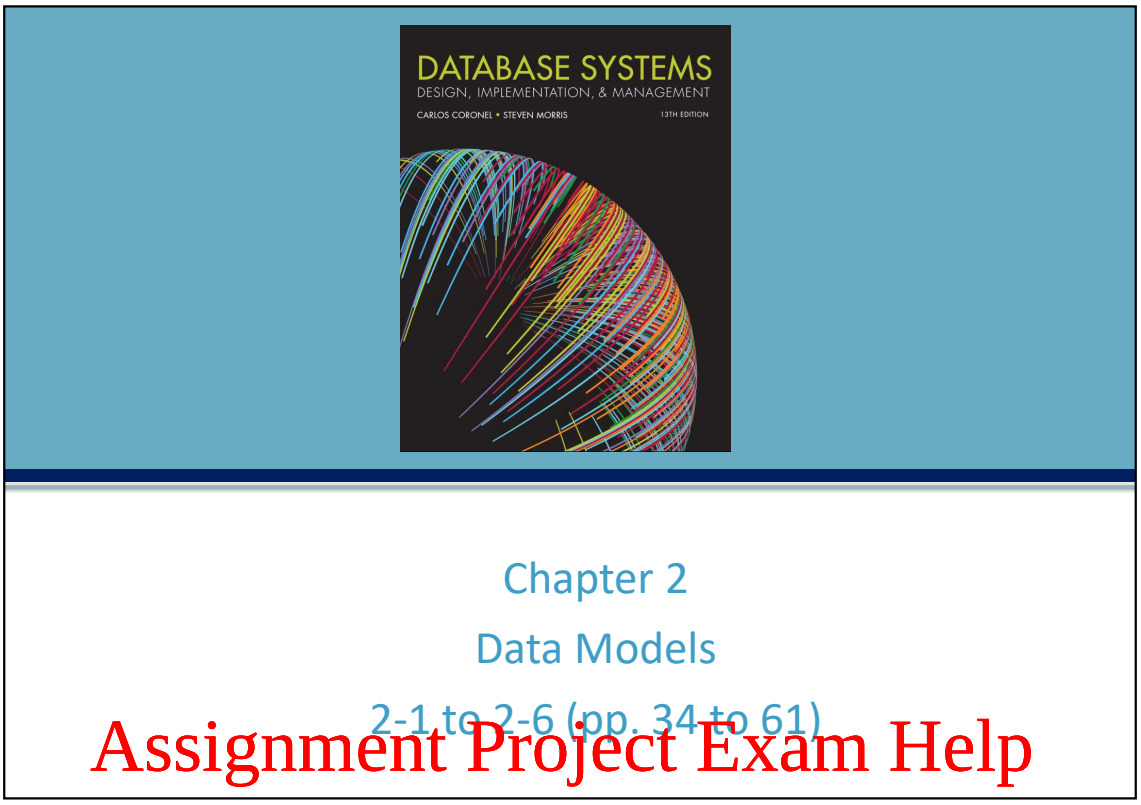
In the Skill table, basic Database Manipulation is Skill_Id 210. Now let's go to Certified table, Skill_id is the middle column. For 210, we have two 06234 and 09382. Now, we can go to Employee table; you can see 06234 is Jasmine Patel and, 09382 is Jessica Johnson.

If you want an easier way, you will write SQL statements to link these tables together to get the results. This is something you can look forward to doing in the lab.



In summary, to generate an accurate and valuable information, you need to have a well-designed database.

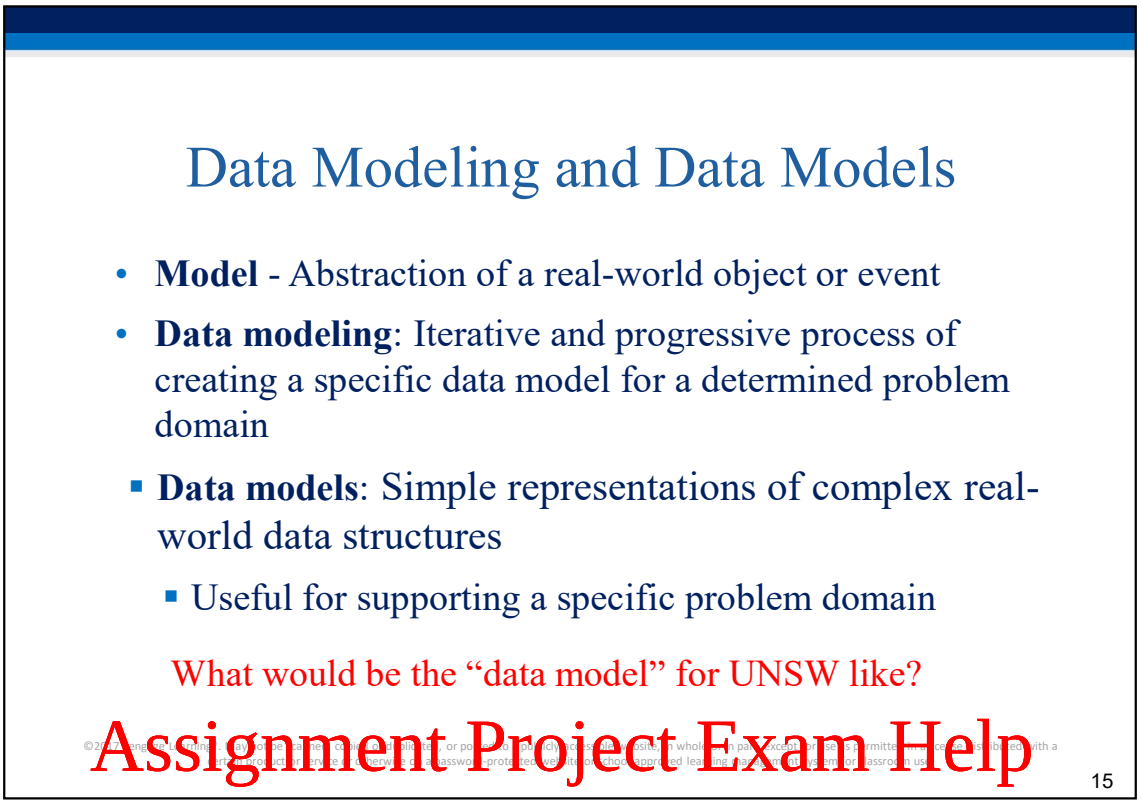
This is all I want to cover for Chapter One. You may want to read other topics I didn't cover in lecture. You may want to read through the file system. SAS is based on a file system. like a csv file, because it started in the 1970s,, that is when computer hardware was very expensive!



Next, we will look at data models.

Please note that you do not have worry about the history and years in the chapter. This chapter mainly introduces new terms and concept that use in different types of databases. You have to know what they are. For example, you have to know what hierarchy is, and what network is.

The importance of the concepts such as Entity Relationship Model will be covered in more depths in the next few weeks.



Data modelling is to model and translate the business requirements into a data model they can be used to store data which to business can use.

Data model is designed to capture data in the real world. The data can then be used in the business.

So, what would be the data model like for the UNSW?

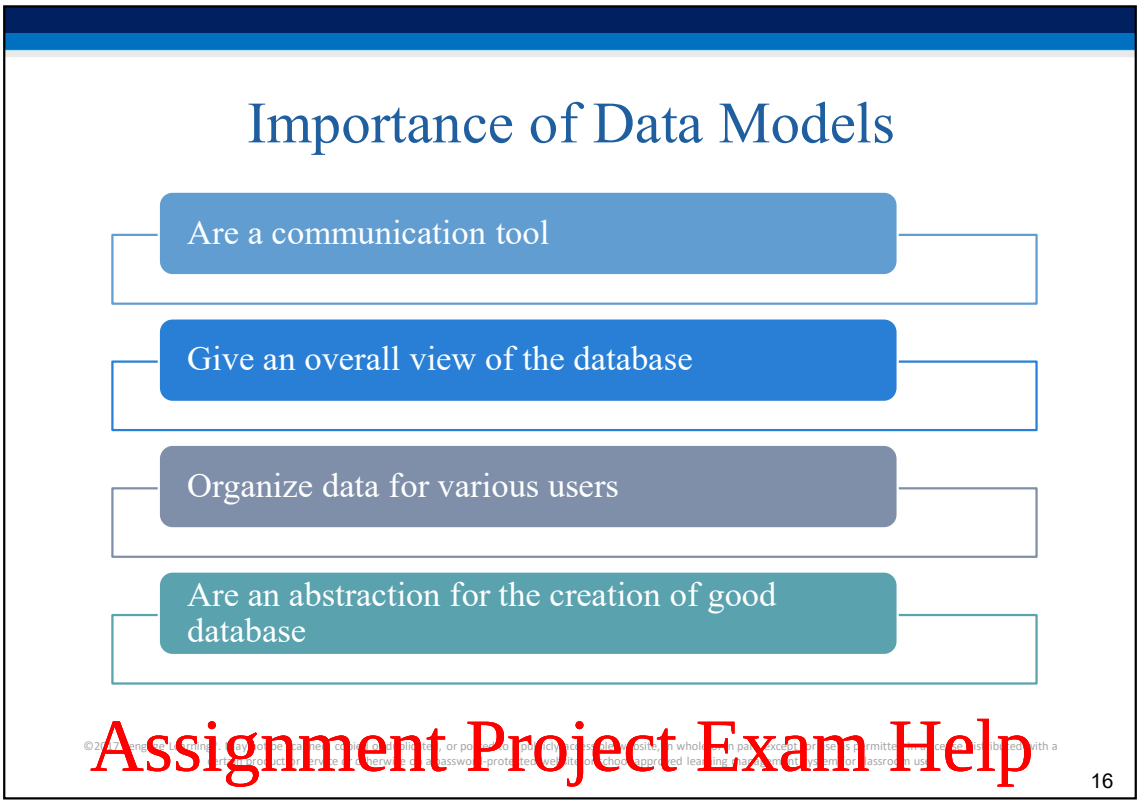
Firstly, you have to think of what data you need to capture. The university needs to capture your details as a student - your name, your address, your telephone number, your gender, and your past academic records.

The university will give you a unique Student Id which identifies who you are. When you do a course like INFS5710, it will capture your marks, your attendance records, your assignment marks, your exam results and your final mark. The system will store all the courses you have attempted, passed or failed.

The system also needs to schedule a class, i.e., where and when INFS5710 class will take place. Can it fit all the students in one class?

Actually, the UNSW system is made up of several systems. Each system has its own data model, and they all linked together. For example, you can have a student system, generate profit and loss statements from the financial system, new student recruitment system, marketing system, student payment system, human resource and payroll systems, university shop rental systems, room booking system, library systems and so on and so forth. All these Systems are joined together; that is all different data models are actually linked together to create one complex data model.

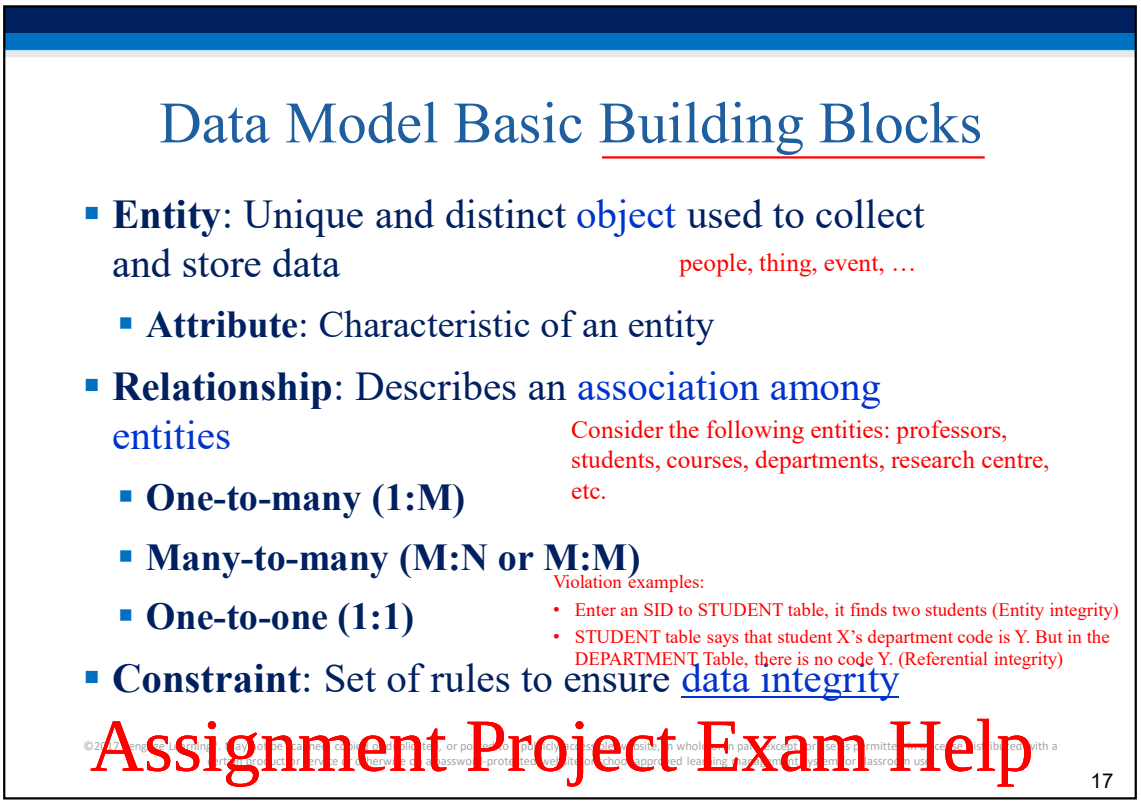
So, a data model can be very complex. The UNSW systems took years to develop, and the systems consistently modified by the IT. Thus, the data models will continually be modified to fit the future needs. For instance, we now have online system as well as face-to-face classes for the same course and are running at the same time.



The importance of data models...

However, getting the data model right or near what it supposes to do is not easy. People sometimes deliberately do not want to tell you some facts. Alternatively, an IT personnel did not ask the right question, or the question is misinterpreted. On the other hand, it might be the question is so simple that no one has thought about it.

For example, in one of the projects when I was working in the industry. The project is about transferring data between two different systems. A document circulated and went through by a few consultants and a few people until we had a telephone meeting, I asked a question, ”how do I know if the dollar amount is a credit or a debit (i.e. positive or negative)” There was a silence for about a minute, I then asked, “hello, anyone there?” I said to my boss, ”the connection must have dropped out." A voice came from the other end, “No, we are still here. We did not look at that, and we need to go and check, and then come back to you..."



They are two parts of database design. One is logical, and the other is physical. In the past, it might take a few months to do the logical design, and then it might take 1 to 2 months to do the physical design. It will take time to make any changes. However, today, logical design and physical design are linked such as Oracle SQL Developer. At the initial design phase, this makes life much easier as you can easily change the logical design, and it will reflect in the physical design.

The basic building blocks of all data models are entities, attributes, relationships, and constraints. If I were to ask you to design a data model, I will get 50 (assuming 50 students here) different designs and notations, and the ones which are the same is because the person sits next to you copies yours.

An entity is a person, place, thing, or event about which data will be collected and stored; for example, a student entity.

An attributes is a characteristics of an entity; for example, for our student entity, you have attributes such as a student last name, student first name, student address, student status, for instance, a local student or an international student.

A relationship describes an association among entities. In our student example, you have

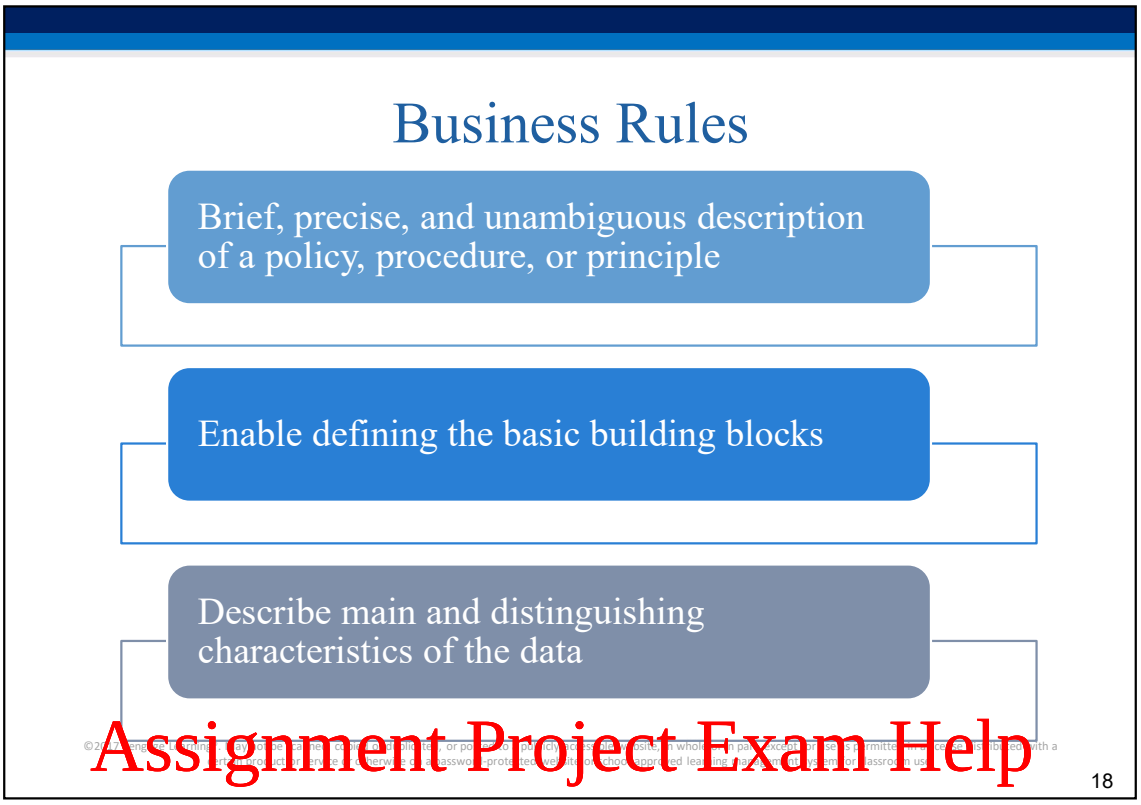
one student belongs to one faculty only, such as the Business School, or one student has to do several courses to be graduated. The relationship, we have cardinalities such as one-to-many, many-to-many. and one-to-one.

For example, for the case of one-to-one, you can only belong to one faculty. For one-to-many, an example could be one student can enrol in many courses.

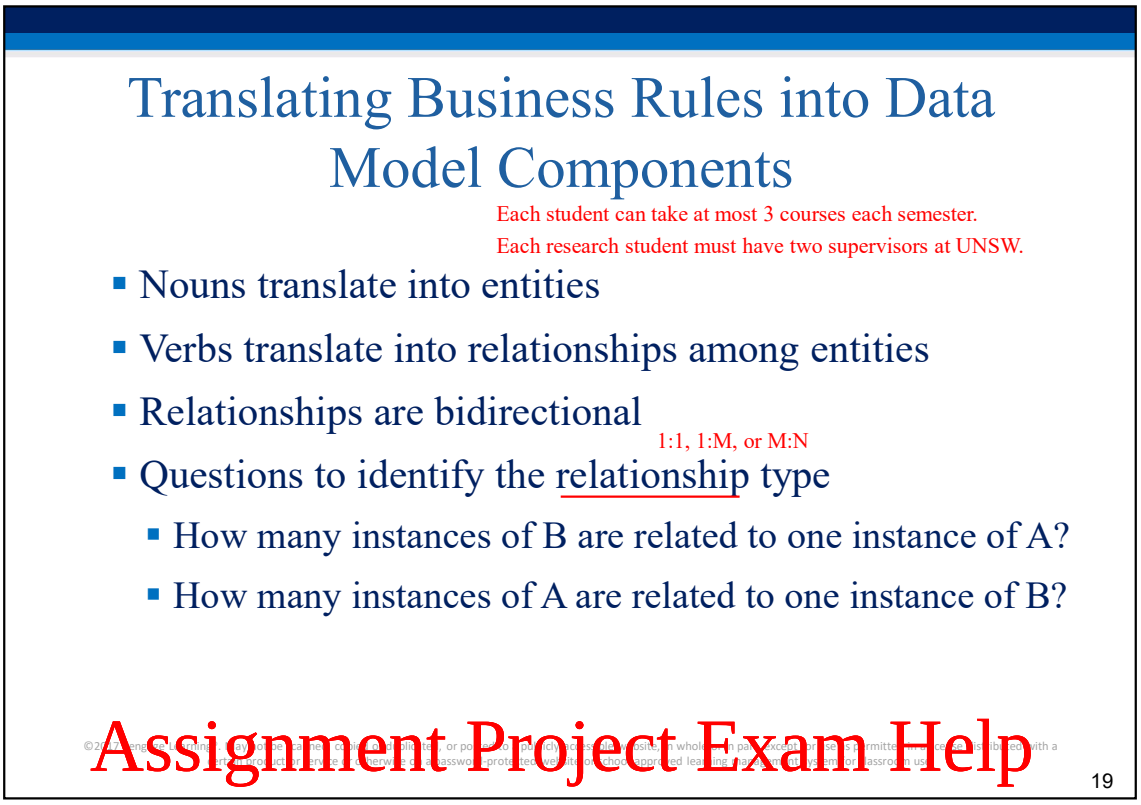
As for many-to-many, it is something you must address in your logical data modelling because you cannot save the data in the physical form.

Many-to-many must be resolved at the logical design stage, and we will come back later when we do the Entity Relationship Diagram or ERD.

A constraint is a restriction put in placed on the data. Constraints are important because they help to ensure data integrity. In our student example, we can have a grade of High Distinction which is above 85%, a Distinction grade is between 75% and 84%, a Credit between 65 to 74%, and a pass above 50%,. And below that you fail the course. The data integrity here is to ensure all the grades are matched to the marks.



We have business rules, to design a data model!



Nouns such as student or course can be translated into *entities*. For example, each **student** can take at most 3 **courses** per term.

Verbs such as **take** can be a *relationship*.

When it comes to relationship type, it can be one of the three - 1:1, 1:M, or M:N. You have to work it out yourself based on the three business rules. The business rules drive the relationship between entities.

For example, if you have two entities, one is taxi driver and the other one is car, i.e. taxi and car.

- If a taxi driver can only drive one car only and cannot drive another car, then the relationship between the entities is one to one (1:1)
- However, if the taxi driver can drive any cars, then the relationship between taxi drive entity and car entity is one to many (1:M)

Naming Conventions

- Entity names - Required to:
  - Be descriptive of the objects in the business environment STUDENT, EMPLOYEE, DEPARTMENT, etc.
  - Use terminology that is familiar to the users
- Attribute name - Required to be descriptive of the data represented by the attribute
- Proper naming: STUDENT: SID, DOB, GENDER, FIRST_NAME, LAST_NAME, …
  - Facilitates communication between parties
  - Promotes self-documentation

20

Companies usually have standard related to naming conventions.

The Relational Model (Ch 3)

By Codd in 1970

- Produced an "automatic transmission" database that replaced "standard (manual) transmission" databases
- Based on a relation
  - **Relation** or **table**: Matrix composed of intersecting tuple and attribute

    Can you see why a "table" reflects a "relation"?
    - **Tuple**: Rows
    - **Attribute**: Columns
- Describes a precise set of data manipulation constructs

Each intersecting tuple crossing attributes 1, 2, …, n means this "entity" meets the values of these attributes in the tuple.

21

The relation model was introduced in 1970 by E.F. Codd. He worked at IBM when he wrote the paper on relational model.

The relational model is based on mathematics, and it is break through for the end users and database designers.

To use an analogy, the relational model produced an automatic transmission database to replace the standard transmission databases that was before it.

The conceptual idea of having a relation changes the way database is designed.

Until today, we still use relational models but now we have other big data models because we have too much data. We will cover relational model in more details next week.

Read the textbook, it explains more…

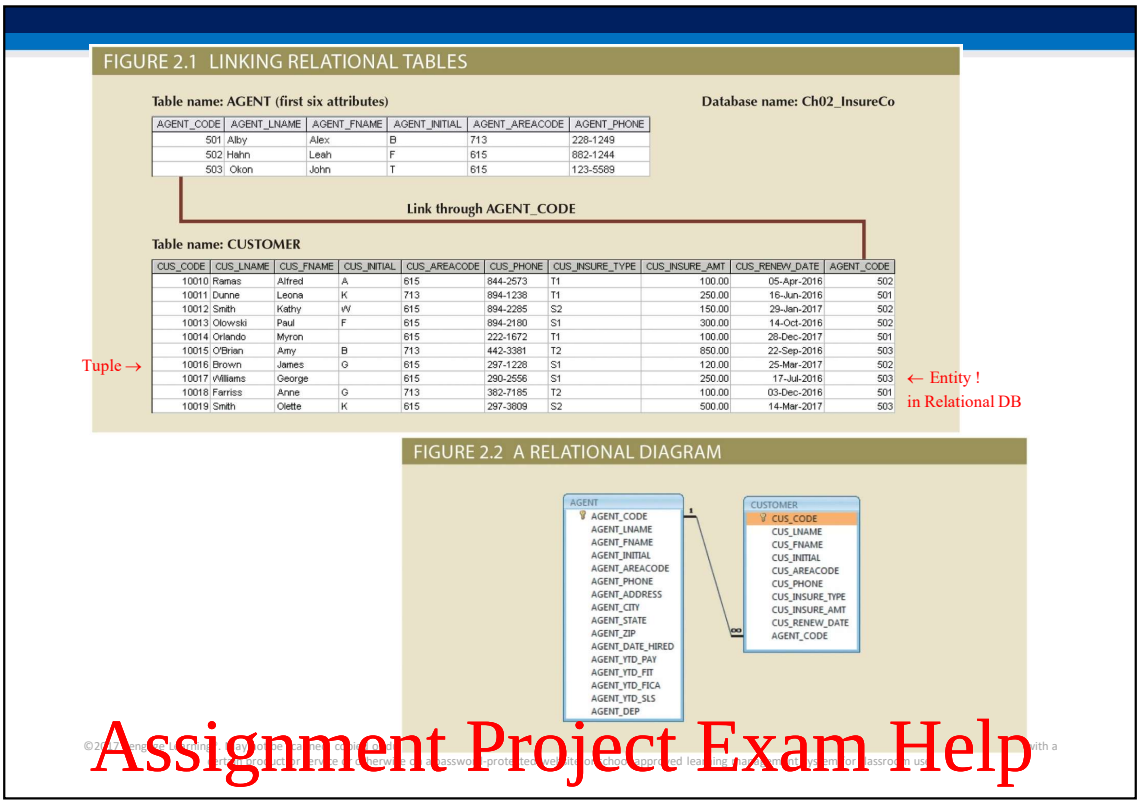


FIGURE 2.1 LINKING RELATIONAL TABLES

Table name: AGENT (first six attributes)

Database name: Ch02_InsureCo

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Link through AGENT_CODE

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2016	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2016	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2017	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2016	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2017	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2016	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2017	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2016	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2016	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2017	503

Tuple →

← Entity ! in Relational DB

FIGURE 2.2 A RELATIONAL DIAGRAM

Here we have two tables. We now want to see how the

In the first diagram we have agent code links two tables, namely table agent and table customer, together.

The second diagram shows the relation between the two tables.

Agent code is a primary key or unique in table Agent, and agent code in the customer table links to agent.

In Figure 2.2, you can see an agent code must exist in agent table before it can be stored in customer table.

As for the cardinality, you can see 1 next to Agent Code in Agent table and many in customer table. The reason is you can only have one unique agent code in agent table, but in the Customer table, you can have many same agent code stored many times.

The Entity Relationship Model (Ch 4)

By Chen in 1976

- Graphical representation of entities and their relationships in a database structure
- **Entity relationship diagram (ERD)**
  - Uses graphic representations to model database components

An entity is a table in ER model!

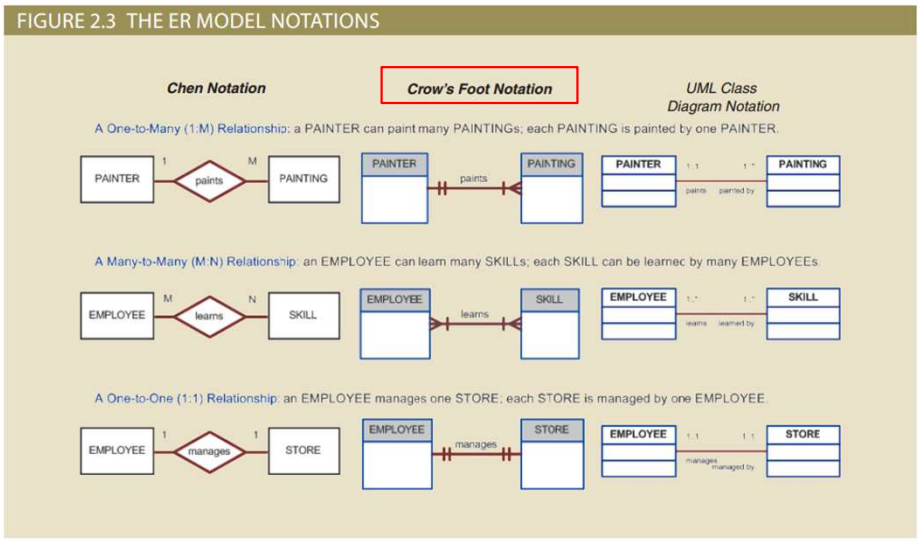
- **Entity "instance" or entity "occurrence"**
  - Rows in the relational table
- **Connectivity**: Term used to label the relationship types

A relationship refers to the associations between tables.

23

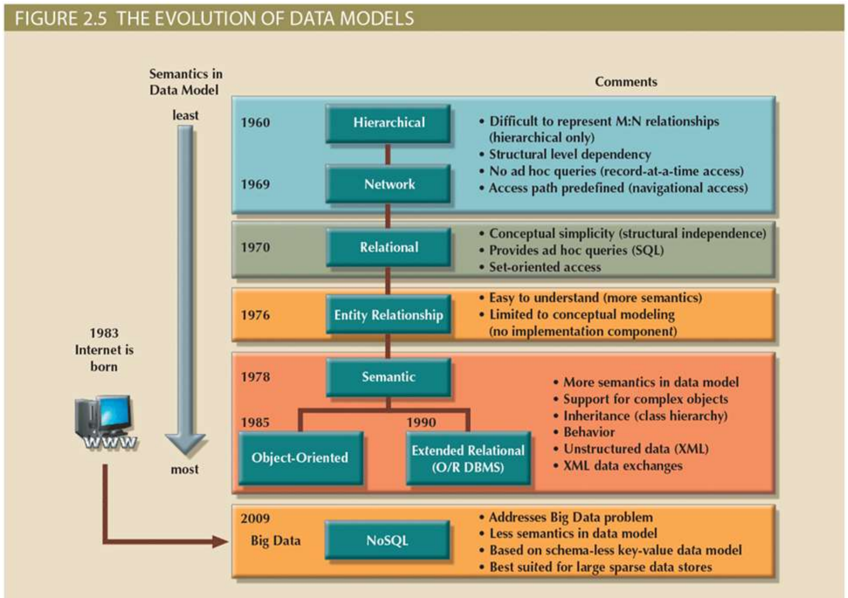
As for ER model, we will chat about it next week.

Figure 2.3 - The ER Model Notations



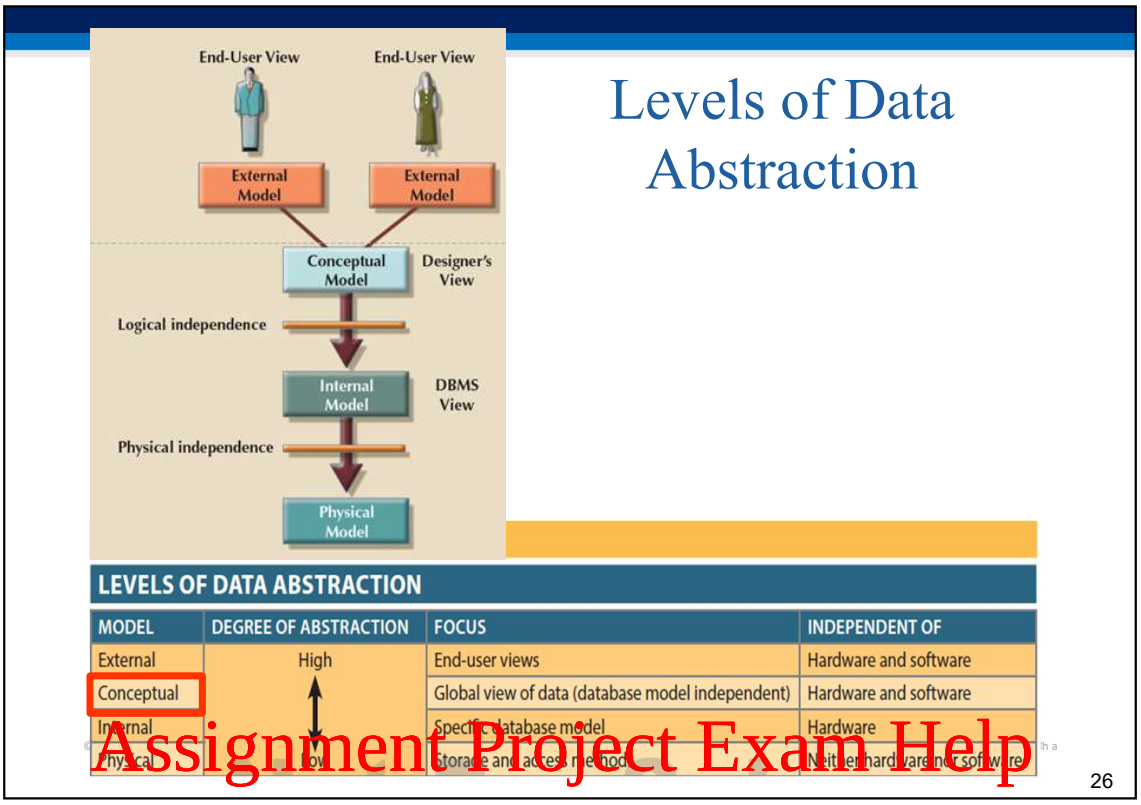
There are a few notations around. The book uses Crow's foot notation, so we will use that notation. I used Chen Notation when I was a student. There is other notations in Oracle which you will expose in the lab in a few weeks time.

The Evolution of Data Models



25

In the textbook, it explains very well what these data models are, what is a hierarchical, what is a network and what is a NoSQL. We will focus and discuss more on relational, entity relationship, and NoSQL in the last few weeks.



From the definition, Data abstraction is the reduction of a particular body of data to a simplified representation of the whole.

Abstraction, in general, is the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristic.

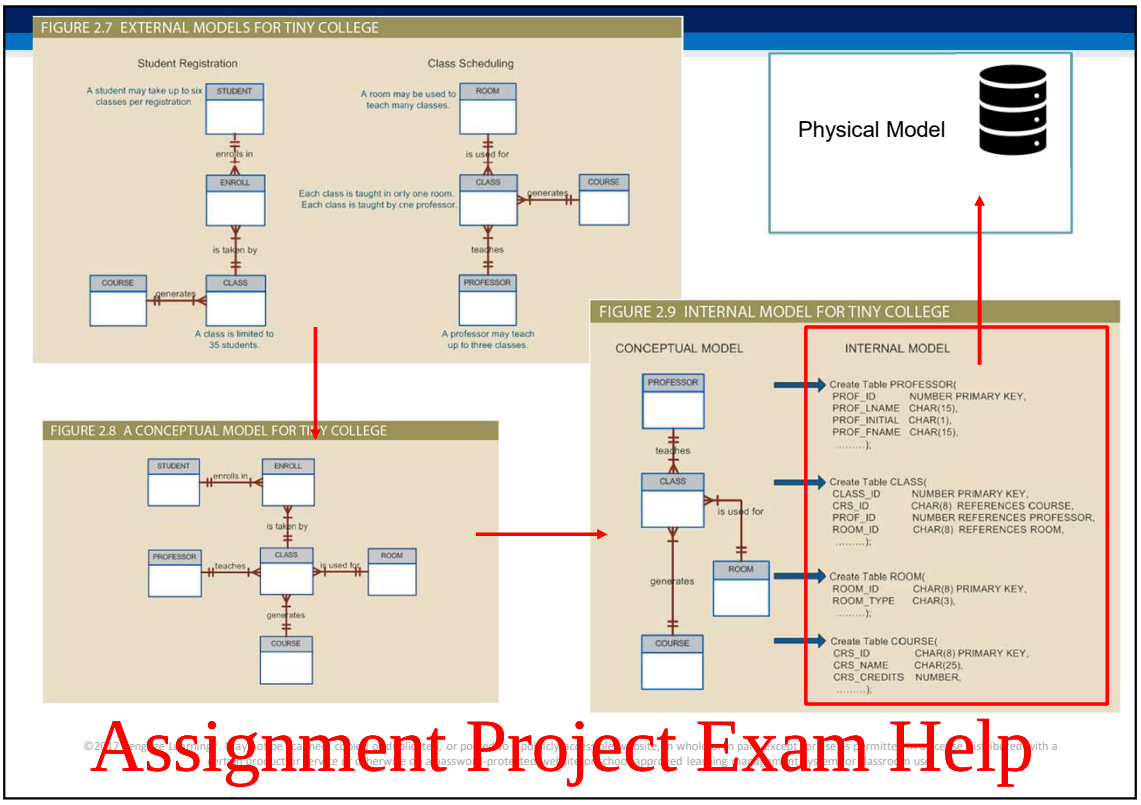
In the textbook, it talks about how a car designer designs a car.

An engineer then designs to see how it can be transferred from a concept car to a car in production.

The engineer needs to design specifically down to the screw and size of the light bulb to be used in the car.

You can see from the diagram the viewpoint of getting the business rules form the end user all the way to designing a physical model.

As for the data abstraction, it is high at the end users and when it comes down to a low level at the physical model.



This shows how the model is transferred from one level of model to another. From External model to Conceptual mode to internal model and then to physical model. In Oracle, you can do that.

Nowadays, in reality, due to time constraints and money, conception model, will be built as the first model and then click of a button, it translates into a physical model. ☺

Questions?

https://powcoder.com

Add WeChat powcoder