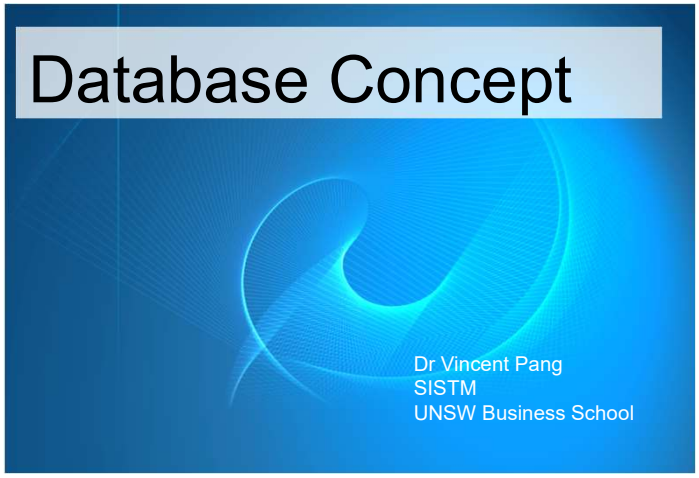
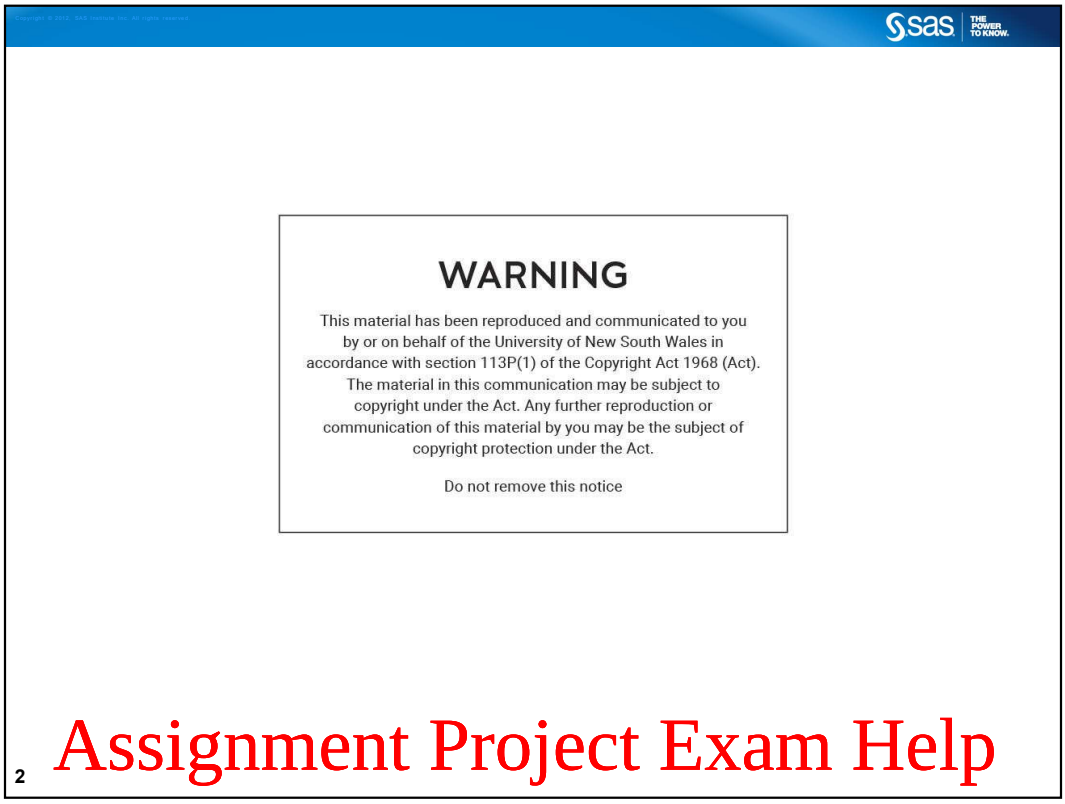
sas THE POWER TO KNOW.

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

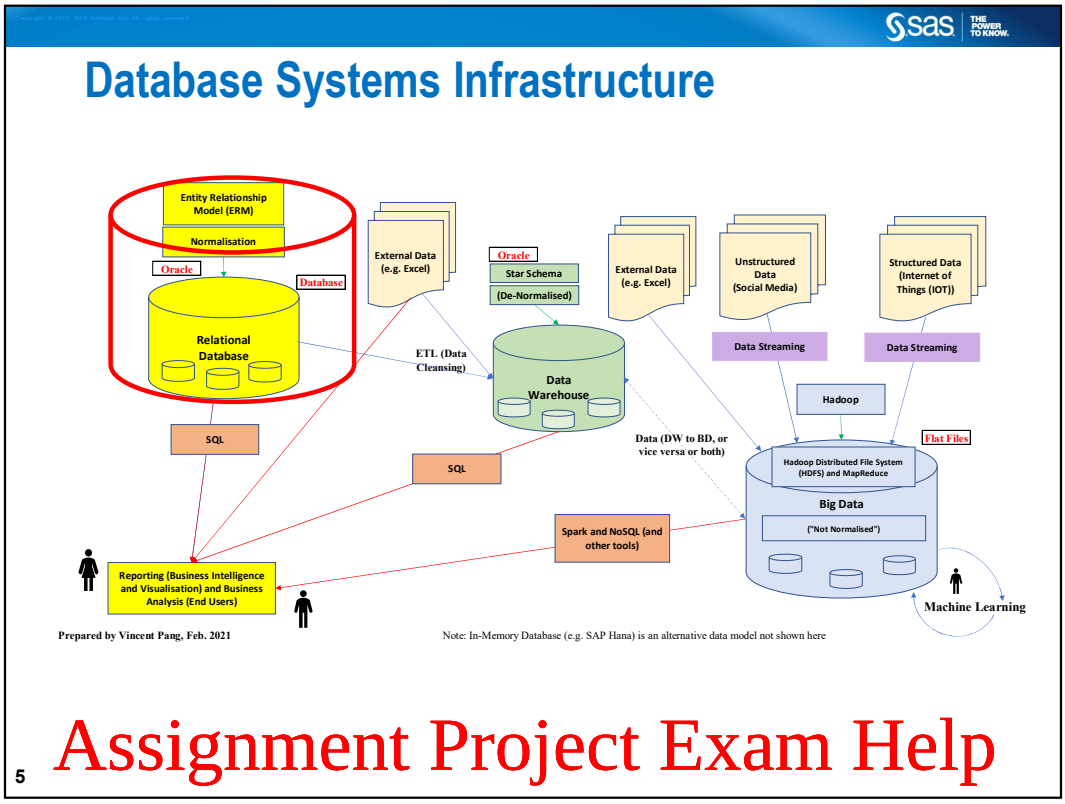
UNSW Business School acknowledges the Bidjigal (Kensington campus) and Gadigal (City campus) the traditional custodians of the lands where each campus is located.

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

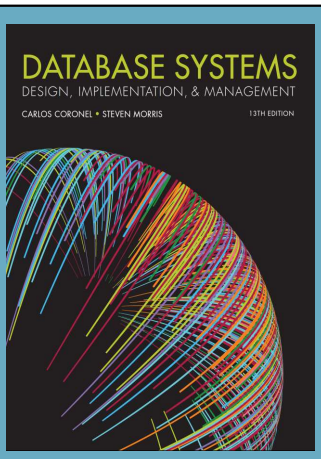
4



The notes in the speaker notes originally were for my use only, but students saw my notes and they asked for them, so I start to share with you all. It is not perfect but hopefully, it makes sense to you!

For the next two weeks, we will be looking at relational database model, particularly Entity Relationship Modelling or ERM. or sometimes we just call it Entity Relationship Diagram or ERD.

This week we will be looking at relational table, and next week, we will be looking at normalisation of database tables. So, that is the red cylinder.



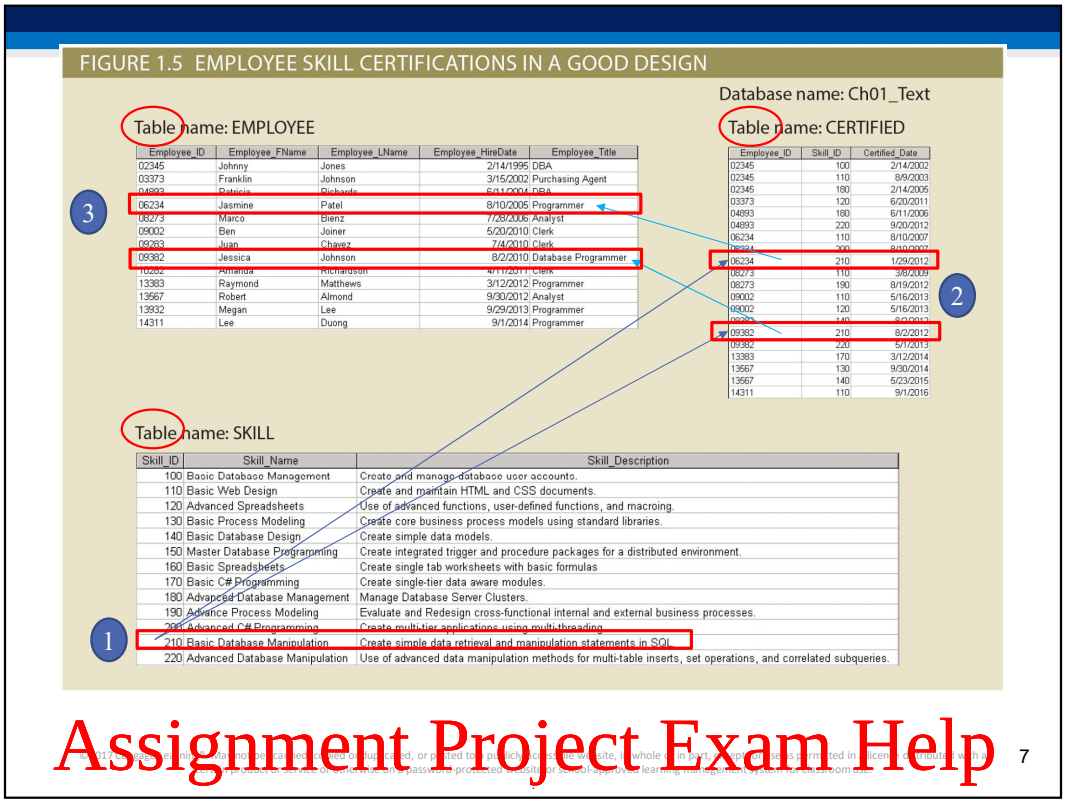
Chapter 3

The Relational Database Model

3-1 to 3-7

pp. 69-99

We will not cover indexing in 3-8.



Let's recap from last week...

Recap, you have three tables instead of one gigantic table. You will learn how to do this in ERD and Normalisation which will cover in the this week and next week. We will show how to create ERD and next week we will do normalisation to ensure tables and columns are well defined.

Just a quick recap if you forgot what these tables are about: 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 do in the lab.

Table 3.1	Characteristics of a Relational Table
1	A table is perceived as a two-dimensional structure composed of rows and columns. relation = table
2	Each table row (tuple) represents a single entity occurrence within the entity set. row = tuple = "entity"
3	Each table column represents an attribute, and each column has a distinct name. column = attribute
4	Each intersection of a row and column represents a single data value.
5	All values in a column must conform to the same data format.
6	Each column has a specific range of values known as the attribute domain.
7	The order of the rows and columns is immaterial to the DBMS.
8	Each table must have an attribute or combination of attributes that uniquely identifies each row. → called "primary key"

8

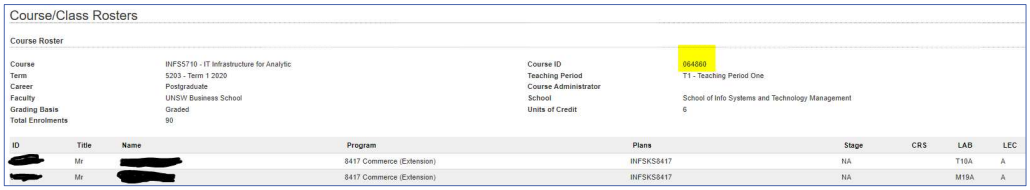
Okay, firstly, we look at a relational table. Let's look at Table 3.1 it describes characteristics of a table. A table is like a 2-dimensional spreadsheet containing rows and columns. The sheet tab in Excel is like a table in relational database. The row is the tuple, and column is the attribute, and the cell contains the data.

For each column, like in Excel, it has to be the same type of data. For example, if a column is a date, then the whole column must be all dates. If a column is numeric, then the whole column must be numeric.

For the last point, number 8, each table must have an attribute or combination of attributes that uniquely identifies each row. We also called this Primary key, which we will discuss later.

A Report

When you see a course and class rosters report, such as below, you know you need a few tables to link together.



Course/Class Rosters

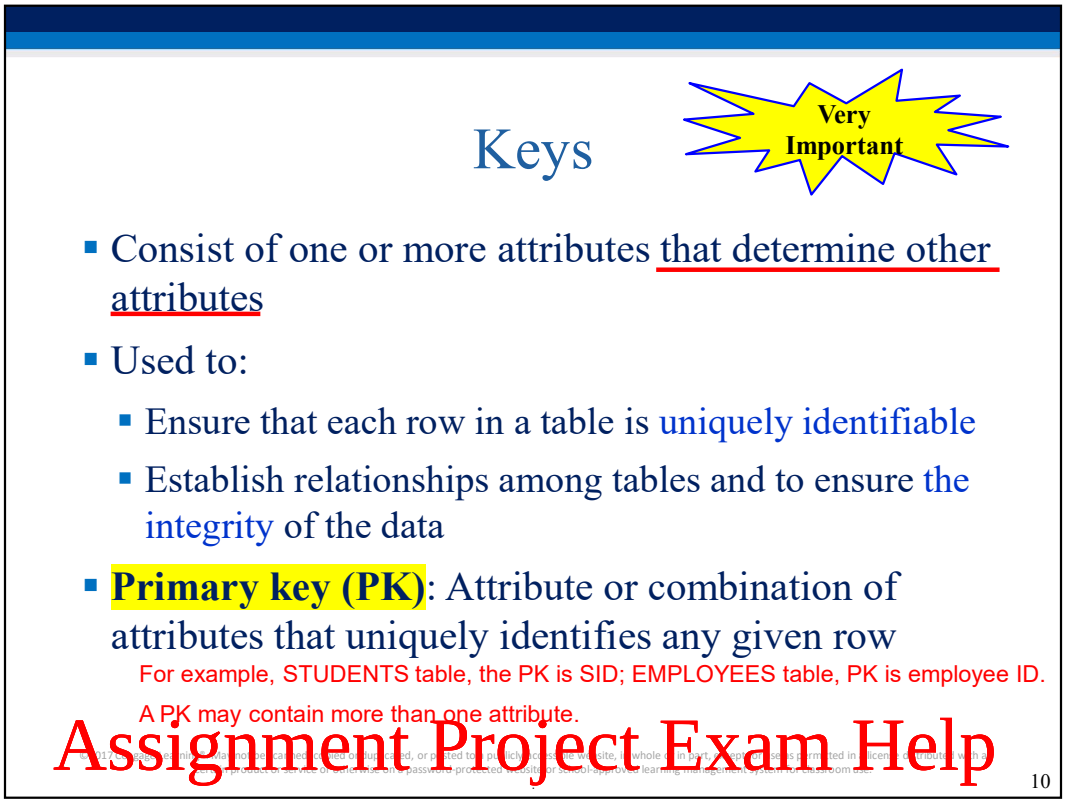
Course Roster

Course	INFS5710 - IT Infrastructure for Analytic	Course ID	064860
Term	5203 - Term 1 2020	Teaching Period	T1 - Teaching Period One
Career	Postgraduate	Course Administrator	
Faculty	UNSW Business School	School	School of Info Systems and Technology Management
Grading Basis	Graded	Units of Credit	6
Total Enrolments	90		

ID	Title	Name	Program	Plans	Stage	CRS	LAB	LEC
[redacted]	Mr	[redacted]	8417 Commerce (Extension)	INFSKS8417	NA		T10A	A
[redacted]	Mr	[redacted]	8417 Commerce (Extension)	INFSKS8417	NA		M19A	A

This is actually the report I see on your class enrolment in My UNSW. This was taken on Sunday, 23rd February, 2020.

It has your course details, the term details, undergraduate or postgraduate, which faculty offers this course, how many students enrolled, which term. and how many units of credit. You can see there is a course id **064860, which is uniquely key identified for this course**.



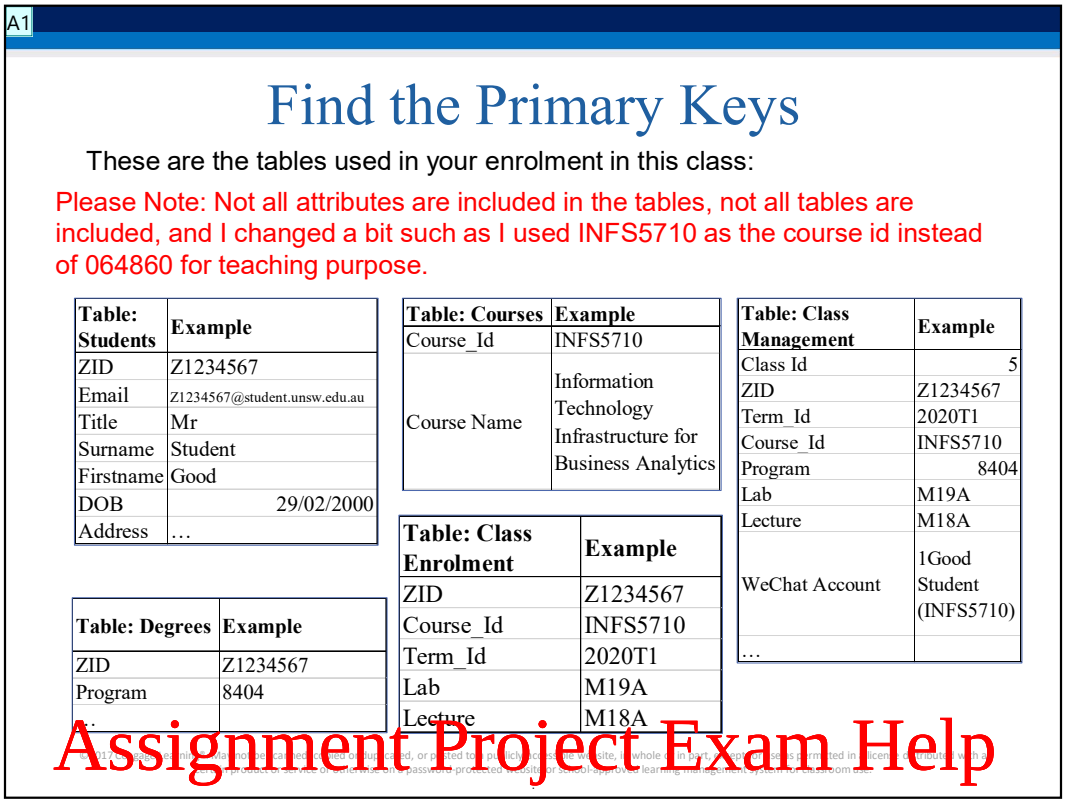
A primary key uniquely identifies one or more columns in a row or record of a table. For example, student id or student number uniquely identifies you at the UNSW, or course id 064860 for INFS5710.

So, whenever you fill in a form at the university or login using your student id, the system knows it is you. Thus, student id is a primary key and uniquely identifies you and you only.

In a S table (say) in the university system, a column ZId will be the primary key of that table. If your student id is 1234567, and if someone tries to enter a new record with ZId 1234567, it will be rejected because it is already existed in the table. Other examples include your driving license, and your tax file number.

In the definition of primary key, it can be uniquely identified by one or more columns. This means you can have one, two, three, four or even more columns combine to form a primary key.

Did you notice that the course description is “IT Infrastructure for Analytics” whereas the course description is officially “Information Technology Infrastructure for Business Analytics”. Thus, nearly all the time you do not use description as a PK.



Please do not look at the answers on the next slide!

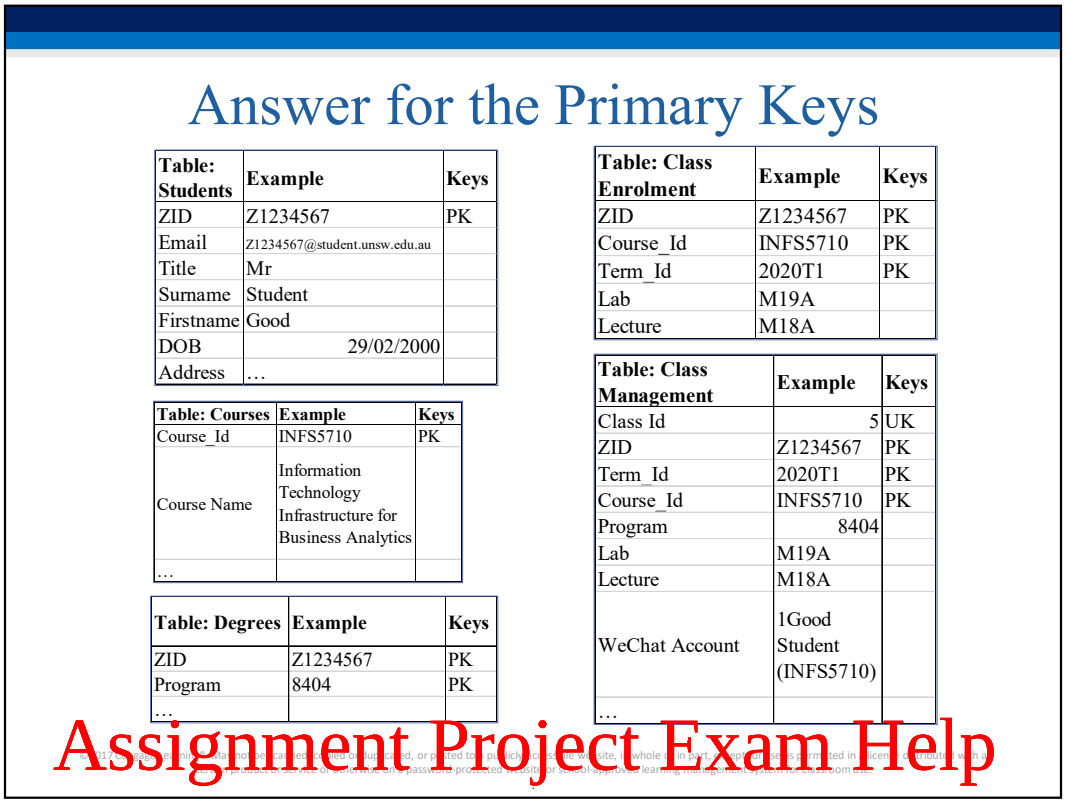
These are the tables used in your enrolment in this class!

Please Note that I did not include all attributes in the tables. I changed a bit such as I used INFS5710 as the course id instead of 064860 for teaching purpose.

Let start you off, the primary key for students table is the ZId because the ZId identifies you and you only.

How about Courses? …

A1 Author, 23/02/2020



The primary key for students table is the ZID because the ZID identifies you and you only.

Course_Id is the primary key for Courses because Course_Id identifies one course only.

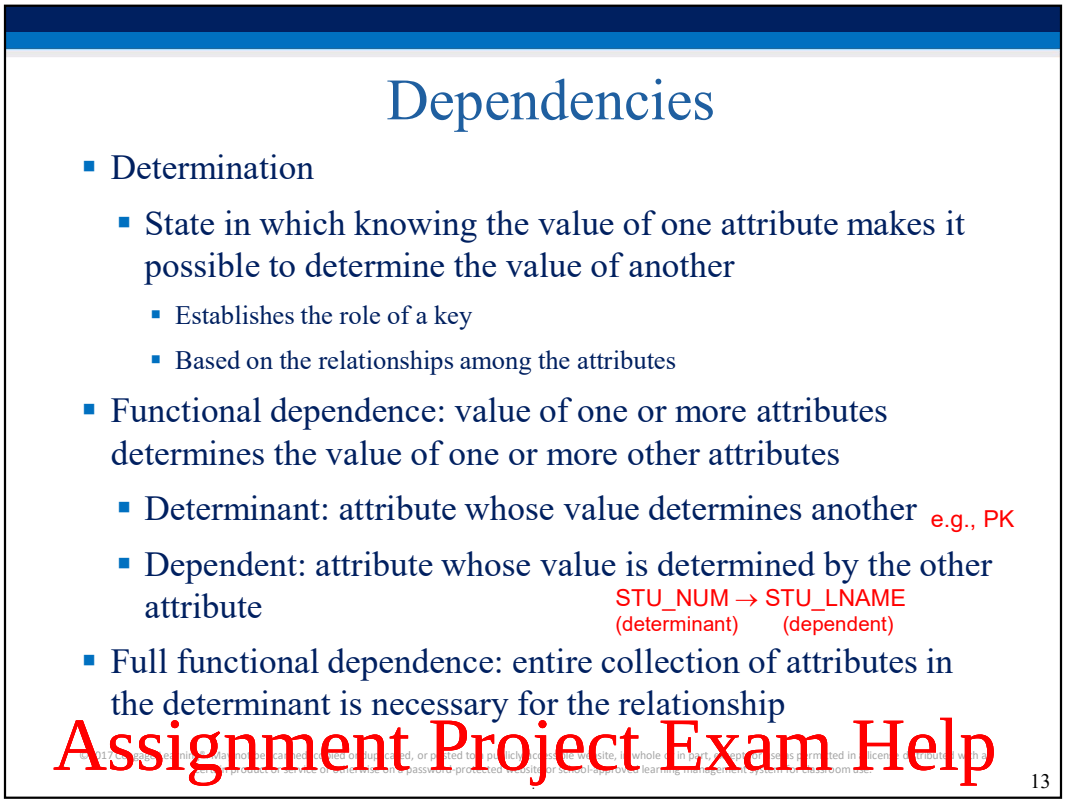
To enrol in a course, you need your student number (ZId), course you want to do (course_id), and in which term (term_id). Thus, the primary key in course_enrolment table consists of three columns, namely ZId, term_id, and course_id; sometimes this is called composite key. If you fail this course, then you have to this course again, then Term_Id will be 2020T3, and you will have different lab and lecture times.

As for Degrees table, it is actually ZID and Program because you can change a program Also, you need to take into accounts of undergraduate and postgraduate if you have done undergraduate previously.

The degrees table is simplified in our case.

As for Class Management table, again, your student number (ZId), course you want to do (course_id), and in which term (term_id) are together to form a composite key. As for class_id, it is an unique key or UK, it has to depend on the uniqueness of the composite key. I only use class_id in this class to manage your assessments, attendances, and assignments, and I also allow you to use the class id to find and sign up your group members. Please note

I will call this a local table as the class management table only used by me and I delete all records at the end of the term, and one can use this table except me. We will come back to this in a few weeks time.



Determination is the state in which knowing the value one attribute makes it possible to determine the value of another. The idea of determination is not unique to the database outright. For example, Profit = revenue minus costs. As long as you know the revenue and costs, you can determine the profit. Determination in a database environment, however, is not normally based on a formula but on the relationships among the attributes. For example, if I know the student number, I can find out the student name.

Functional dependence is when the value of one or more attributes determines The value of one or more other attributes. For example, student number and student name are shown here.

Types of Keys (1)

(A, B) (A, B) → C

- **Composite key**: Key that is composed of more than one attribute
  For example, the course enrolment table has ZId, course_id, and term_id is a composite key
- **Key attribute**: Attribute that is a part of a key
- **Superkey:** key that can uniquely identify any row in the table
- **Candidate key**: minimal superkey

A, B

- **Entity integrity**: Condition in which each row in the table has its own unique identity
  - All of the values in the primary key must be unique
  - No key attribute in the primary key can contain a null

We have already discussed a composite key. The example of course enrolment table has ZId, Course_Id, and term_id together to form a composite key.

Key attribute is when an attribute that is a part of a key

A Superkey is a key that can uniquely identify any row in the table. For example, Course_Id in Courses table, ZID in Students table, and ZId, term_id, and course_id for course_enrolment table.

A candidate key is a minimal superkey. For example, in the Student table, your mobile number might able to identify who you are. Sometimes, when you go a shop, you might forget your rewards card, they might ask you for your mobile number.

Entity integrity is when a condition in which each row in the table has its own unique identity. All of the values in the primary key must be unique and no key attribute in the primary key can be null.

Types of Keys (2)

- **Null**: Absence of any data value that could represent:
  - An unknown attribute value
  - A known, but missing, attribute value
  - An inapplicable condition
- **Referential integrity**: Every reference to an entity instance by another entity instance is valid
- **Foreign key (FK)**: primary key of one table that has been placed into another table to create a common attribute
- **Secondary key:** Key used strictly for data retrieval purposes (e.g., people do not remember their membership no. (PK), the secondary key can be their name, which may not be unique.)

15

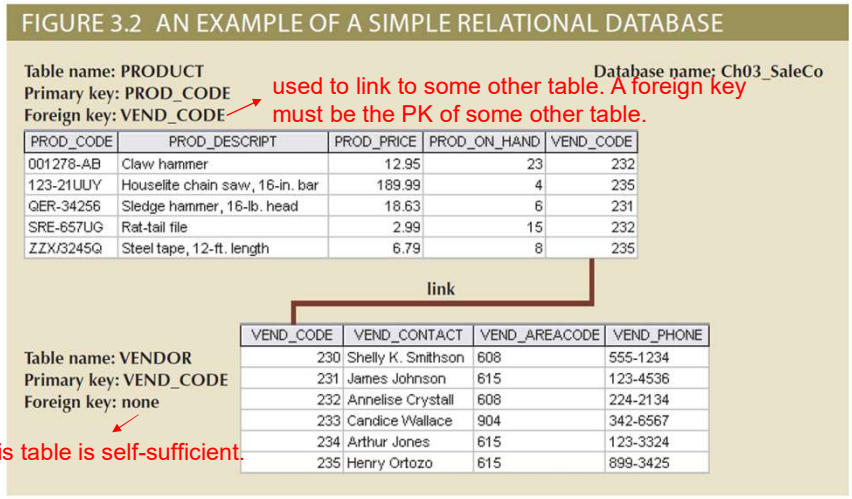
So, what is Null? Null is the absence of any data value. So what does it mean in English and how it is different from zero?

Referential integrity is when you want to ensure, for example, before you add a record with a ZID, you need to ensure the ZID exists in the students table.

A Foreign key (FK) is when you associate one attribute of an entity to an attribute of another entity.

Secondary key is the key used specifically for data retrieval.

Figure 3.2 - An Example of a Simple Relational Database



This is in the textbook, have a read through the example.

Table 3.3 - Relational Database Keys

TABLE 3.3 e.g., the set of all attributes

RELATIONAL DATABASE KEYS

KEY TYPE	DEFINITION
Superkey	An attribute or combination of attributes that uniquely identifies each row in a table
Candidate key	A minimal (irreducible) superkey; a superkey that does not contain a subset of attributes that is itself a superkey
Primary key	A candidate key selected to uniquely identify all other attribute values in any given row; cannot contain null entries
Foreign key	An attribute or combination of attributes in one table whose values must either match the primary key in another table or be null
Secondary key	An attribute or combination of attributes used strictly for data retrieval purposes

17

Read in the textbook

Integrity Rules

TABLE 3.4

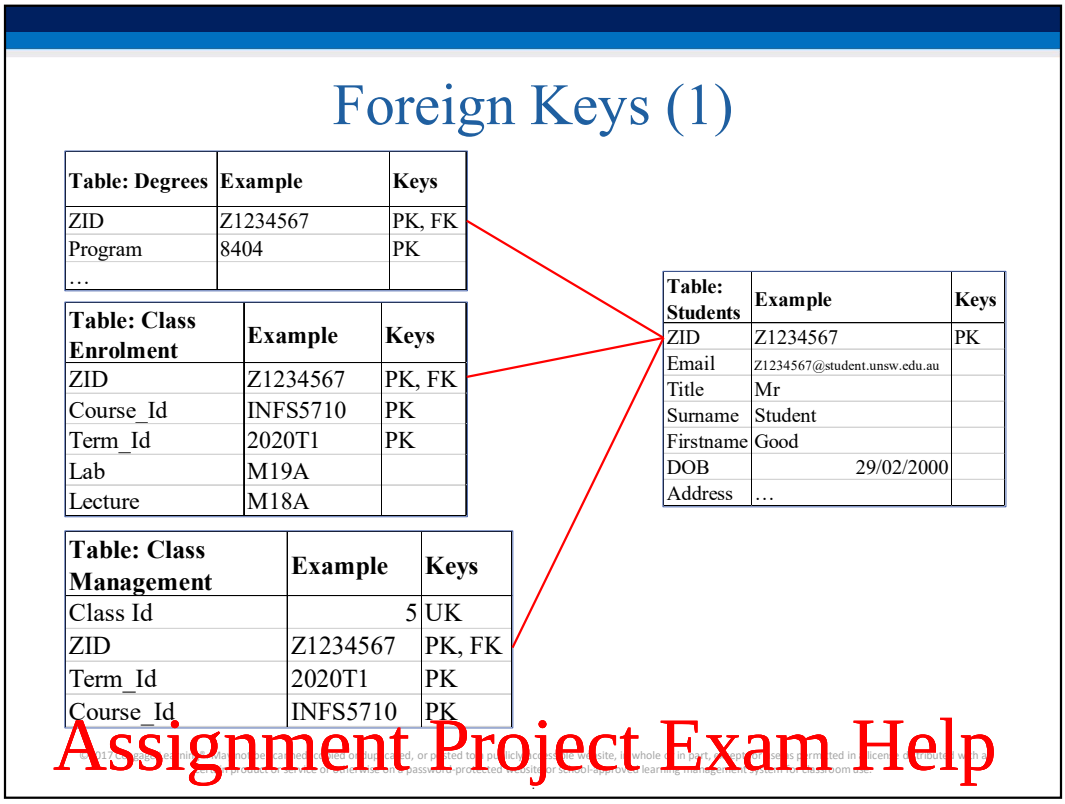
INTEGRITY RULES

ENTITY INTEGRITY	DESCRIPTION
Requirement	All primary key entries are unique, and no part of a primary key may be null.
Purpose	Each row will have a unique identity, and foreign key values can properly reference primary key values.
Example	No invoice can have a duplicate number, nor can it be null; in short, all invoices are uniquely identified by their invoice number.
REFERENTIAL INTEGRITY	**DESCRIPTION**
Requirement	A foreign key may have either a null entry, as long as it is not a part of its table's primary key, or an entry that matches the primary key value in a table to which it is related; (every non-null foreign key value *must* reference an *existing* primary key value). e.g., STUDENT.supervisor = "null" (not yet assigned)
Purpose	It is possible for an attribute *not* to have a corresponding value, but it will be impossible to have an invalid entry; the enforcement of the referential integrity rule makes it impossible to delete a row in one table whose primary key has mandatory matching foreign key values in another table.
Example	A customer might not yet have an assigned sales representative (number), but it will be impossible to have an invalid sales representative (number).

18

Read in the textbook

So, what are Foreign Keys of these tables?



ZId of Degrees table, ZId of Class Enrolment, and ZId of class management are all foreign keys and they are associated with primary key of students table.

As for referential integrity, the ZId must exist in students table otherwise it will violate referential integrity. What this means is ZID Z1234567 must exist in the students table, otherwise, you will get an error.

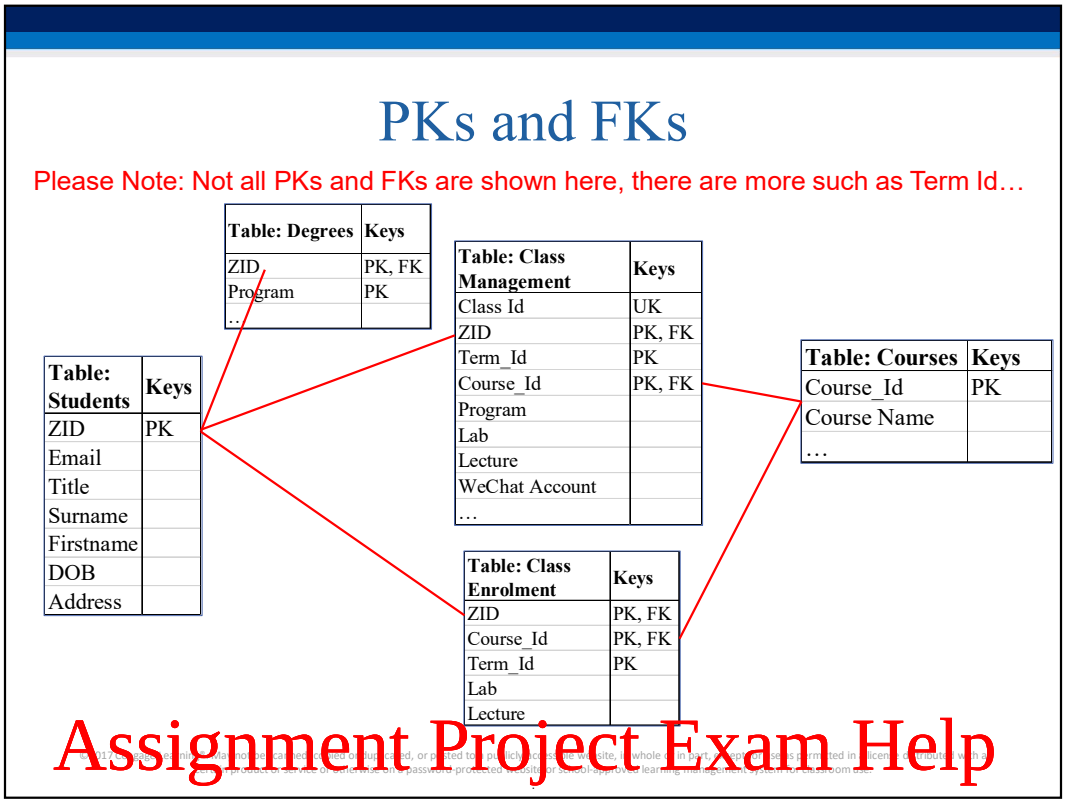
Foreign Keys (2)

Table: Class Enrolment	Example	Keys
ZID	Z1234567	PK, FK
Course_Id	INFS5710	PK, FK
Term_Id	2020T1	PK
Lab	M19A	
Lecture	M18A	

Table: Courses	Example	Keys
Course_Id	INFS5710	PK
Course Name	Information Technology Infrastrucutre for Business Analytics	
…		

Table: Class Management	Example	Keys
Class Id	5	UK
ZID	Z1234567	PK, FK
Term_Id	2020T1	PK
Course_Id	INFS5710	PK, FK
Program	8404	

Course_id of Class Enrolment table, and Course_id of Class management are all foreign keys and they are associated with primary key Course_id of Courses table.



PKs and FKs for all the tables linked to Students and Courses.

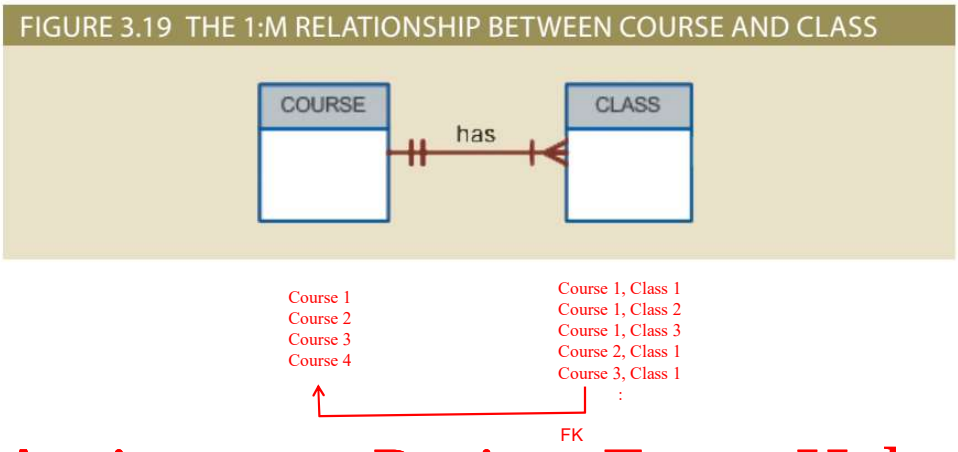
Relationships within the Relational Database

- 1:M relationship - Norm for relational databases
- 1:1 relationship - One entity can be related to only one other entity and vice versa
- Many-to-many (M:N) relationship - Implemented by creating a new entity in 1:M relationships with the original entities
  - **Composite entity** (**Bridge** or **associative entity**): Helps avoid problems inherent to M:N relationships, includes the primary keys of tables to be linked

23

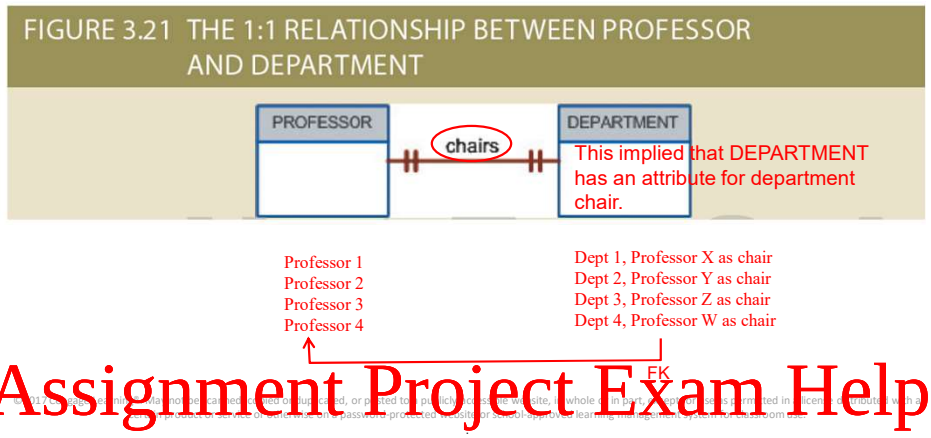
As stated in the slide

Figure 3.19 - The 1:M Relationship between COURSE and CLASS



This is one to many relationship. In plain English, One course has many classes. For example, INFS5710 has two SAS Lab workshops.

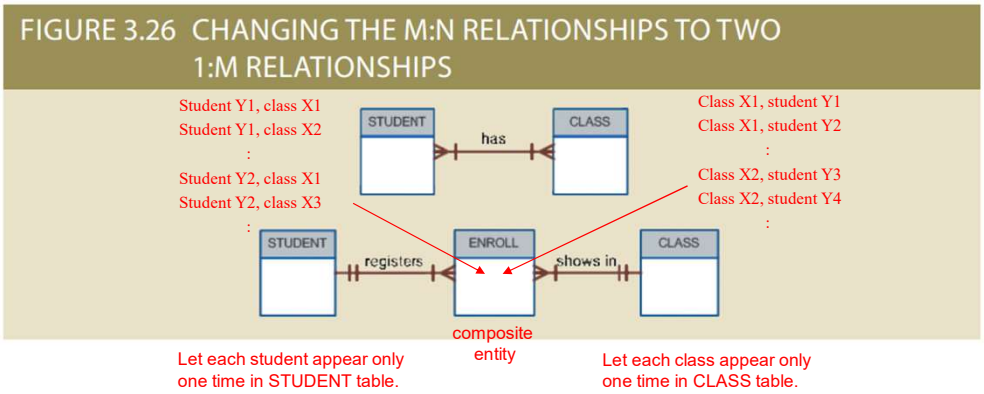
Figure 3.21 - The 1:1 Relationship between PROFESSOR and DEPARTMENT



25

In this one to one relationship, the professor can chair one department, i.e. one department can only have one head.

Figure 3.26 - Changing the M:N Relationship to Two 1:M Relationships



In many to many relationships you have to resolved into at least two one to many relationships. Otherwise, you can never implement in a database. Even if you can implement these tables, you will get in trouble soon or later.

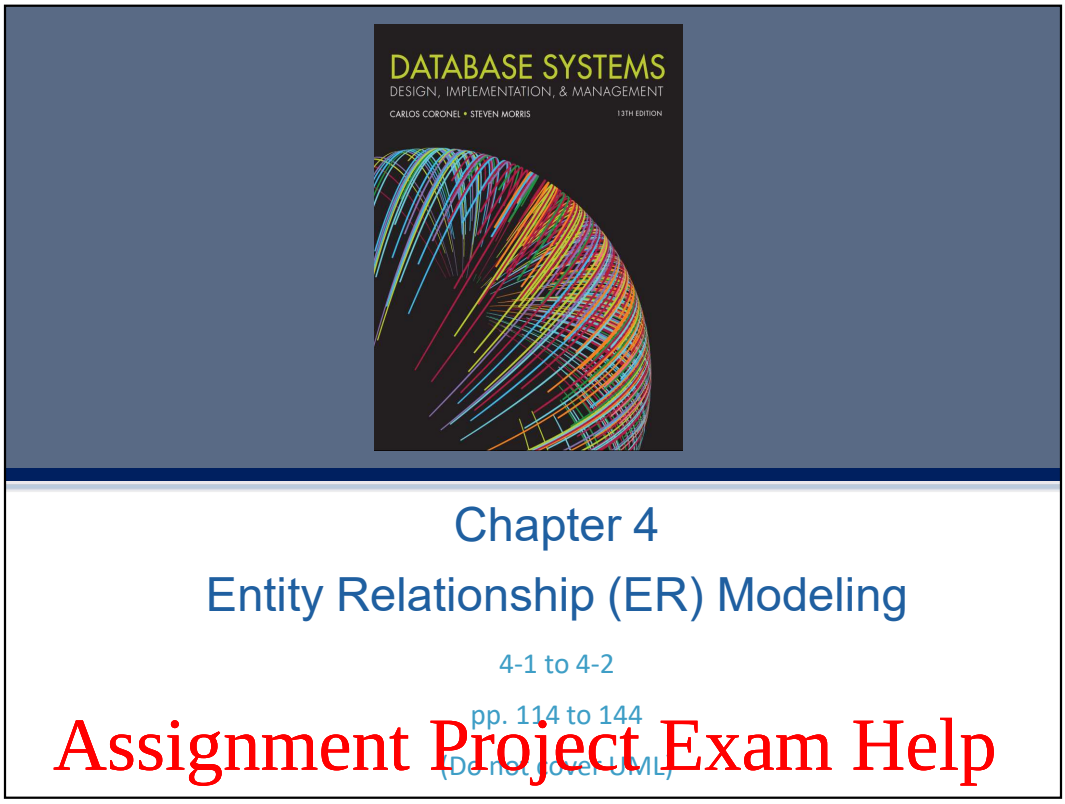
We will cover more later.

Data Redundancy Revisited

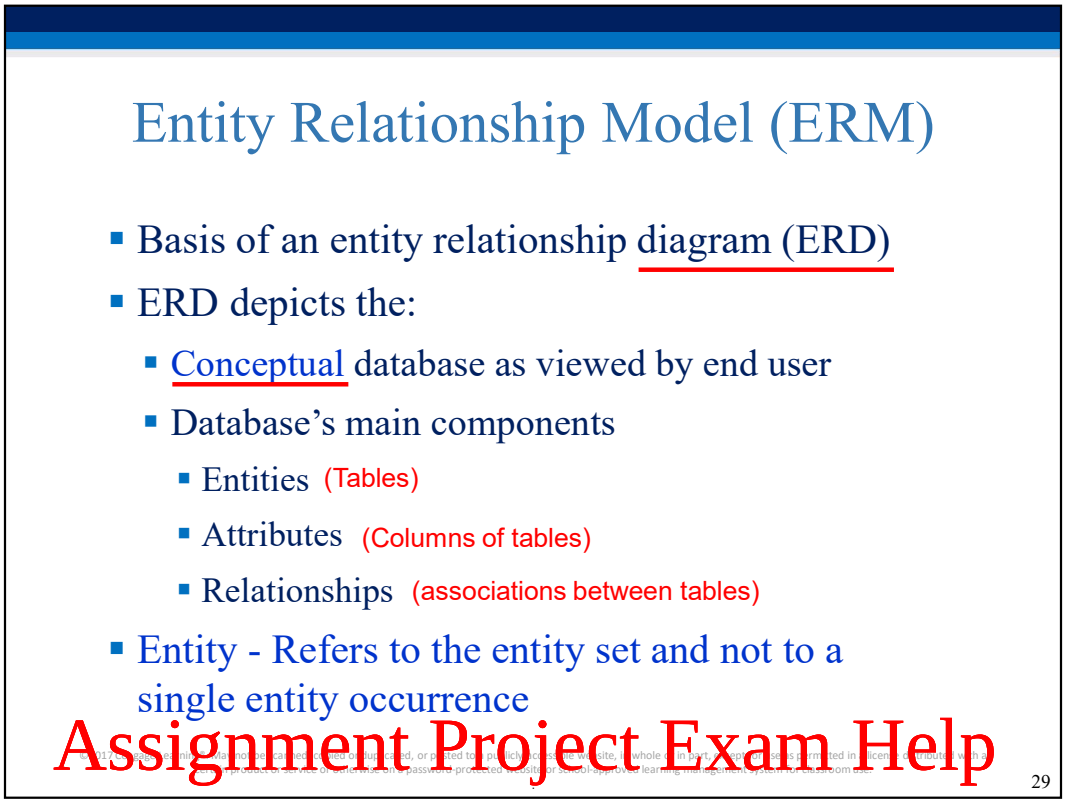
- Relational database facilitates control of data redundancies through use of foreign keys
- To be controlled except the following circumstances
  - Sometimes data redundancy must be increased to make the database serve crucial information purposes
  - Sometimes data redundancy exists to preserve the historical accuracy of data

27

As stated in the slides:



Only cover 4-1 and 4-2.



Entity relationship model is a conceptual design of a database.

It mainly has three components, namely Entities, attributes, and relationships.

We will cover in class but please read the textbook.

Attributes (1)

Characteristics of entities

- **Required attribute**: Must have a value, cannot be left empty
- **Optional attribute**: Does not require a value, can be left empty
- **Domain**: Set of possible values for a given attribute
- **Identifiers**: One or more attributes that uniquely identify each entity instance

called Keys in the relational model

As stated in the slide

Figure 4.1 - The Attributes of the Student Entity: Chen and Crow's Foot

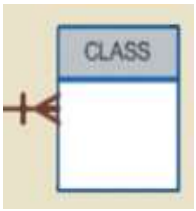
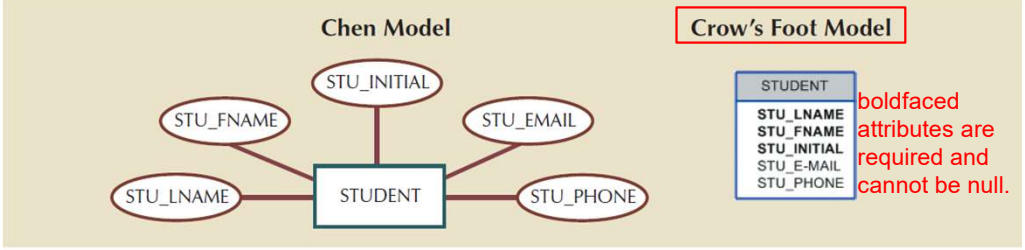


FIGURE 4.1 THE ATTRIBUTES OF THE STUDENT ENTITY: CHEN AND CROW'FOOT



There are many notations used to draw ERW. I was taught in Chen's notations. Chen wrote his thesis on how to represent conceptual database design based on entities and relationships.

In Chen, you underline PK and you underline with dots or something similar for attributes to represent FK

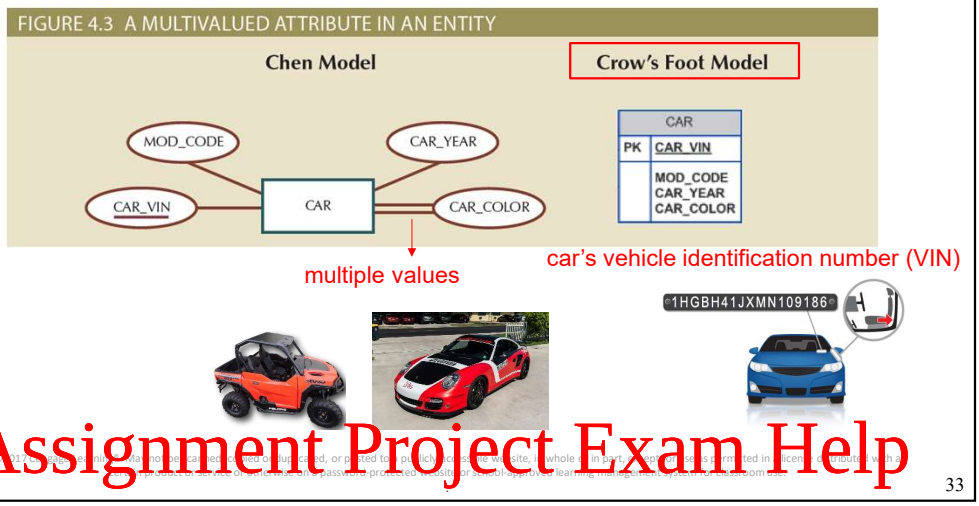
We will focus on using Crow's Foot Model because it uses in the textbook. Why does it call Crow's Foot because it looks a crow foot (see diagram) 😊

Attributes (2)

(composite key)

- **Composite identifier**: Primary key composed of more than one attribute (think about the ENROLL table on slide 14.)
- **Composite attribute**: Attribute that can be subdivided to yield additional attributes e.g., name – John F. Kennedy;
- **Simple attribute**: Attribute that cannot be subdivided DOB – 29/5/1917 e.g., gender – Male
- **Single-valued attribute**: Attribute that has only a single value e.g., gender – Male
- **Multivalued attributes**: Attributes that have many values e.g., diploma – BS, MS, MBA, etc.

Figure 4.3 - A Multivalued Attribute in an Entity



In this example, we are looking at the car entity. Car's vehicle identification number or VIN Is used to identify a car, and that is why VIN is the primary key. VIN can only exist for one car and one car only.

The car can have multiple values, i.e. a car can have different colours, for example, looking at this car, you can have red, white, and black.

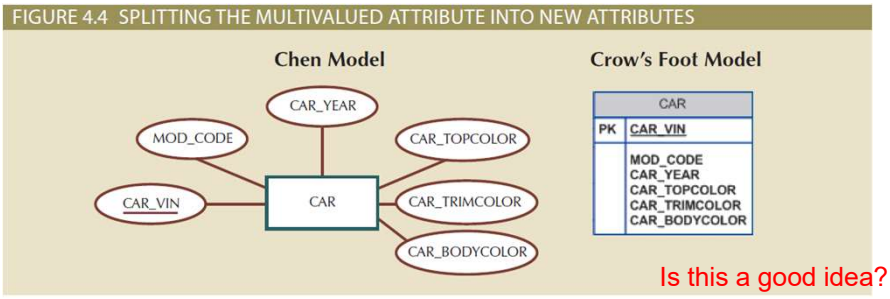
Attributes

- **Multivalued attributes**: Attributes that have many values and require creating:
  - Several new attributes, one for each component of the original multivalued attribute
  - A new entity composed of the original multivalued attribute's components
- **Derived attribute**: Attribute whose value is calculated from other attributes
  - Derived using an algorithm e.g., age. There seems no reason that a database needs to store one's age.

34

As stated in the slide

Figure 4.4 – Splitting the Multivalued Attributes into New Attributes



Is this a good idea?

35

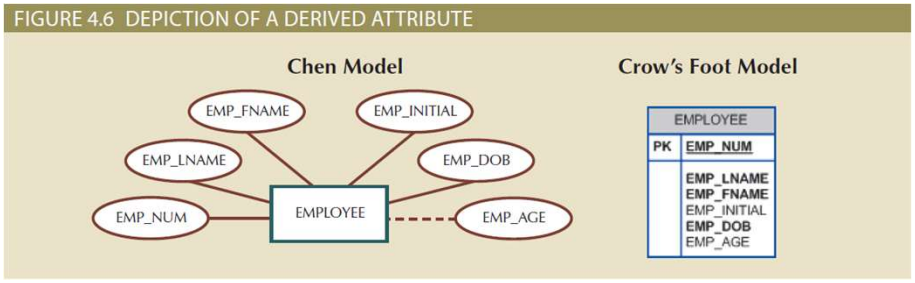
So instead one colour attributes we now have 3 different colour attributes one for top colour, one for trim colour and one for body colour.

However, is this a good idea?

What's happens if you have another colour on the car handle, another colour for the bumper, another colour for the screen wiper, and so on? So, are you going to add more attributes?

This is all depends on how you design.

Figure 4.6 - Depiction of a Derived Attribute



Derived attribute is when the value is calculated from other attributes.

Do we need emp_age?

Technically, we do not store the employee's age because we can calculate from date of birth of the employee. Otherwise, you have to update the age every day.

However, sometimes because of the nature of the business, you might need to do the calculation upfront in order to speed up the data retrieval. For example, on Alibaba's 11/11 Single's Day in China, similar to Black Friday sales or Cyber Monday sales, in 2019, it has a record of US$38 billion in sales in just one day!

That was the biggest sales in just one day. Imagine you can get 1% of the sales, i.e. $380 million.

There are plenty of orders going through every second, so you have to make sure you get those orders, you do not want your customer to wait, that is the important factor. You do not want to waste time doing calculation, such as the unit promotion price, for example. So, you might in this case want to add the unit price to the entity or table. This is exception rather than norm as it heavily relies on the business rules.

Table 4.2	Advantages and Disadvantages of Storing Derived Attributes	
	Derived Attribute: Stored	**Derived Attribute: Not Stored**
Advantage	Saves CPU processing cycles Saves data access time Data value is readily available Can be used to keep track of historical data	Saves storage space Computation always yields current value
Disadvantage	Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change	Uses CPU processing cycles Increases data access time Adds coding complexity to queries

37

As stated in the slide

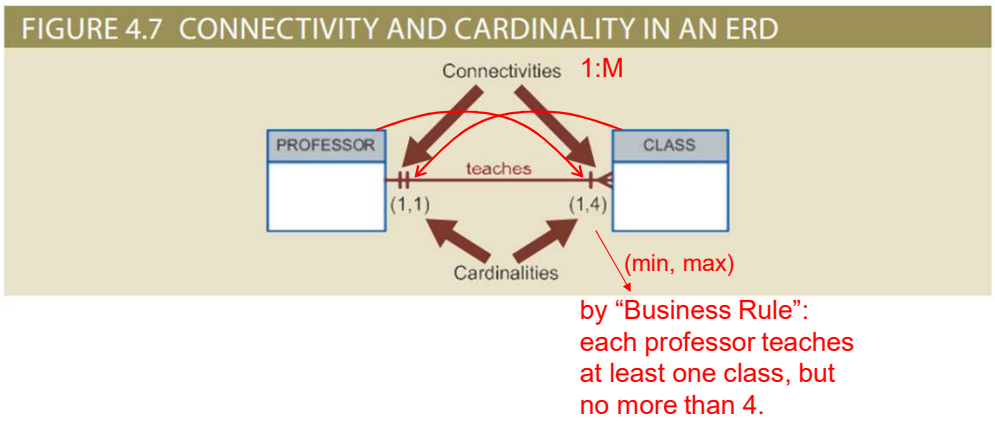
Relationships

- Association between entities that always operate in both directions
  tables
- **Participants**: Entities that participate in a relationship
- **Connectivity**: Describes the relationship ("relationship" previously)
  1:1, 1:M, and M:N
  classification
- **Cardinality**: Expresses the minimum and maximum number of entity occurrences associated with one occurrence of related entity
  e.g., how many diplomas at most one can enter to the table.

As stated in the slide https://powcoder.com

Add WeChat powcoder

Figure 4.7 - Connectivity and Cardinality in an ERD



by "Business Rule":
each professor teaches
at least one class, but
no more than 4.

39

One professor teaches many classes. So, for cardinalities is 1 and 1 on the professor's side but minimum 1 and maximum 4 on the class side as "Business Rule" states each professor teaches at least one class, but no more than 4.

Usually, if you do not know the maximum classes professor can teach, then you can just put a capital M. This is very common in the case of ERDs.

Table 4.3 - Crow's Foot Symbols

TABLE 4.3		
CROW'S FOOT SYMBOLS		
CROW'S FOOT SYMBOLS	CARDINALITY	COMMENT
	(0,N)	Zero or many; the "many" side is optional.
	(1,N)	One or many; the "many" side is mandatory.
	(1,1)	One and only one; the "1" side is mandatory.
	(0,1)	Zero or one; the "1" side is optional.

optional vs. mandatory

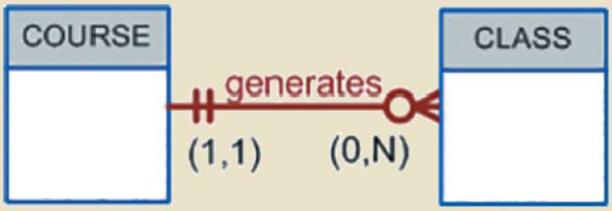
Min = 0 Min = 1

40

Table 4.3 shows the Crow's Foot symbols. It explains more in the textbook, so please read the text book.

Figure 4.13 - CLASS is Optional to COURSE

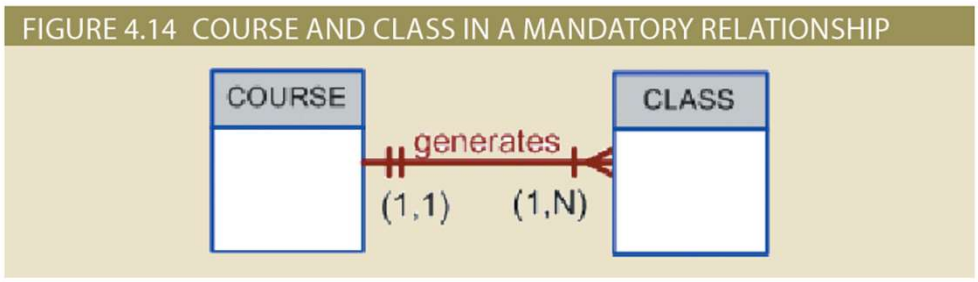
FIGURE 4.13 CLASS IS OPTIONAL TO COURSE



The 'O' Crow foot means optional. Optional participation is when one entity occurrence does not require a corresponding entity occurrence in a particular relationship.

Read the textbook on explaining why this is optional.

Figure 4.14 - COURSE and CLASS in a Mandatory Relationship



Mandatory participation is when one entity occurrence requires a corresponding entity occurrence in a particular relationship.

Read the textbook on explaining why this is a mandatory relationship.

I know what the book said (Fig 4-13 is more likely the case), but in long term, a database designer is more likely to use figure 4.14 to ensure when you have course, you have a class. There are ways to get around this, please see next slide.

Students have been allocated to lecture and lab workshop in Term 1. If you are still on the waiting list, then you will be allocated to section CR01.

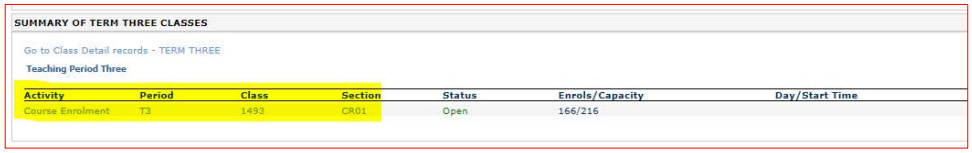
SUMMARY OF TERM ONE CLASSES

Go to Class Detail records - TERM ONE

Teaching Period One

Activity	Period	Class	Section	Status	Enrols/Capacity	Day/Start Time
Course Enrolment	T1	3758	CR01	Open	0/15	
Laboratory	T1	9819	M19A	Open	47/50	Mon 19:00 - 21:00 (Weeks:1-8,10-11)
Laboratory	T1	9820	T10A	Open	44/56	Tue 10:00 - 12:00 (Weeks:1-10)
Lecture	T1	9818	A	Open	91/106	Mon 18:00 - 19:00 (Weeks:1-8,10-11)

Students have not been allocated to any lecture and lab workshop in Term 3. Unlike in Term 1, no class rooms have been allocated.



SUMMARY OF TERM THREE CLASSES

Go to Class Detail records - TERM THREE

Teaching Period Three

Activity	Period	Class	Section	Status	Enrols/Capacity	Day/Start Time
Course Enrolment	T3	1493	CR01	Open	166/216	

You can create a course for people to enrol in – this is can act as a "dummy" first. Once you know roughly how many students express their interest to enrol, then you can open classes based on the numbers. In this case, if you are on the waiting list, you are on the "dummy" list till someone withdraws.

If you have more students want to enrol, then you can increase the numbers in T3.

Next time, when you look at the timetable, you know why, but there is another factor called time. There might be a delay of showing the changes.

Associative (Composite) Entities

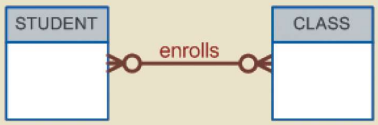
- Used to represent an M:N relationship between two or more entities
- Has a 1:M relationship with the parent entities
  - Composed of the primary key attributes of each parent entity
- May also contain additional attributes that play no role in connective process

As stated in the slide

Many to Many Relationship

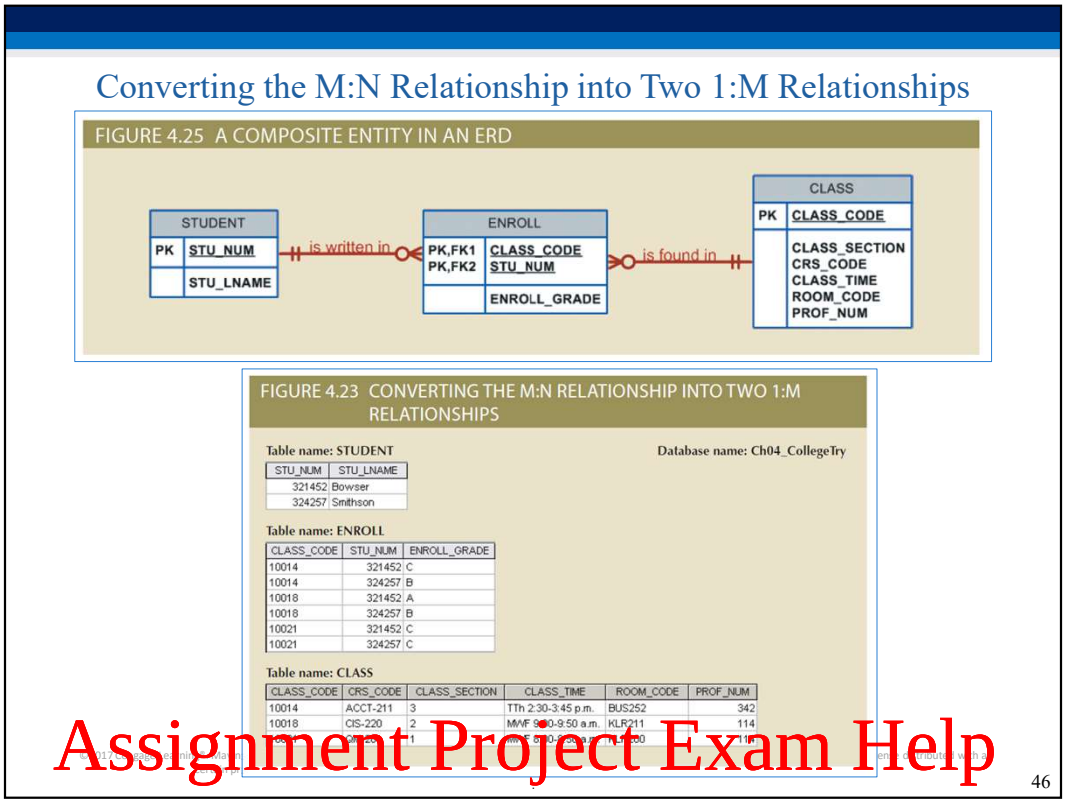
Below shows that many students enrol in many classes. However, this cannot be implemented.

FIGURE 4.24 THE M:N RELATIONSHIP BETWEEN STUDENT AND CLASS



In ERD, you need to resolve all the many to many relationship entities. They will be your nightmare!

If you have one in your ERD, and you did not resolve, then you will face a big problem!



Now, you have converted into one many to many relationship between student and class entities into two one to many relationships by adding enroll entity as shown.

Normally, the primary key of the new entity is made up of a composite key. In this case, class code and student number combine to form a composite key for enroll entity.

Also, have a look at the tables…

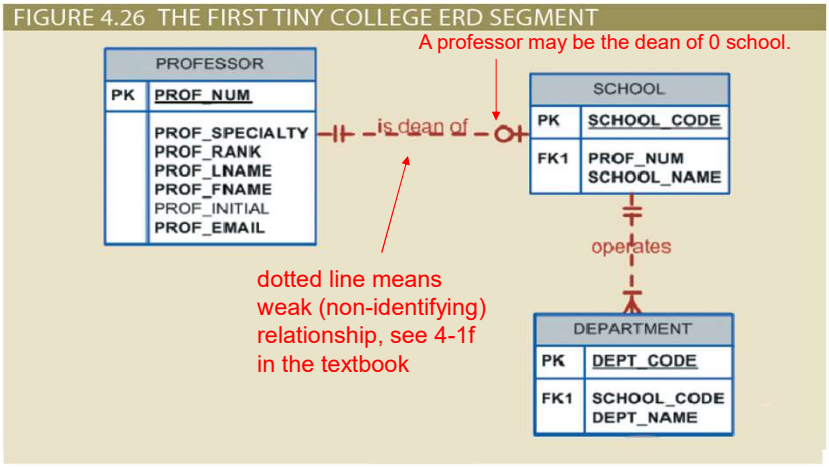
Developing an ER Diagram

- Create a detailed narrative of the organization's description of operations
- Identify business rules based on the descriptions
- Identify main entities and relationships from the business rules
- Develop the initial ERD
- Identify the attributes and primary keys that adequately describe entities
- Revise and review ERD

47

As stated in the slide

Figure 4.26 - The First Tiny College ERD Segment



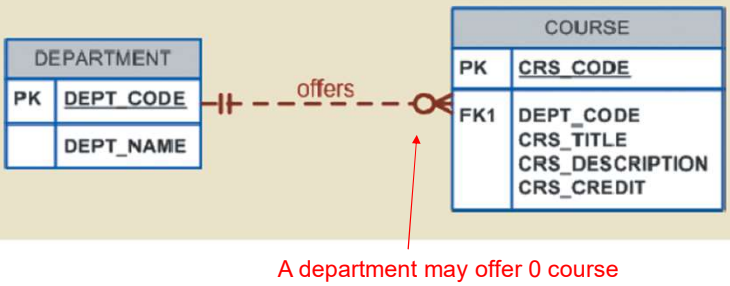
Conditions of a weak entity is when it has an existence-dependent, and it has a primary key that is partially or totally derived from parent entity in the relationship.

That is, a Weak (non-identifying) relationship is when a primary key of the related entity does not contain a primary key component of the parent entity. Read the textbook for explanations.

Database designer determines whether an entity is weak based on business rules. However, when it comes to implementation, it just becomes tables.

Figure 4.27 - The Second Tiny College ERD Segment

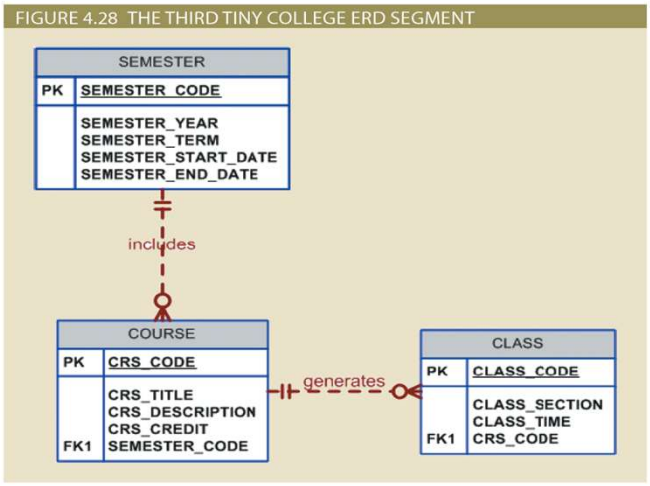
FIGURE 4.27 THE SECOND TINY COLLEGE ERD SEGMENT



A department may offer 0 course

A department may offer 0 course. What's new? Read the textbook for explanations.

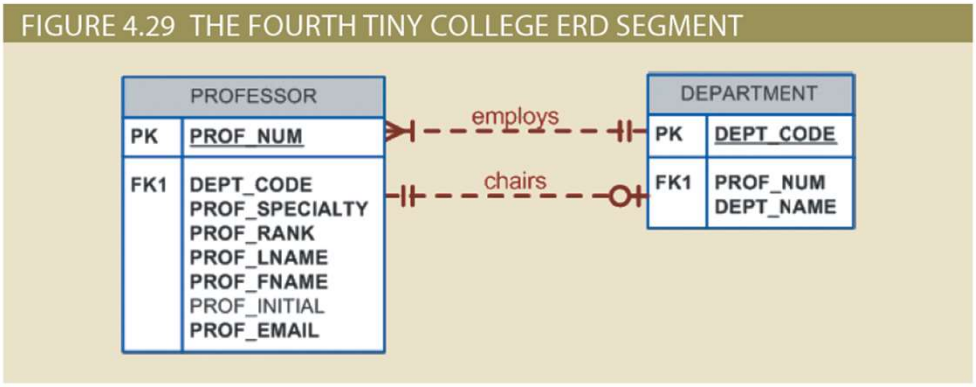
Figure 4.28 - The Third Tiny College ERD Segment



50

As stated in the slide and read the textbook for explanations.

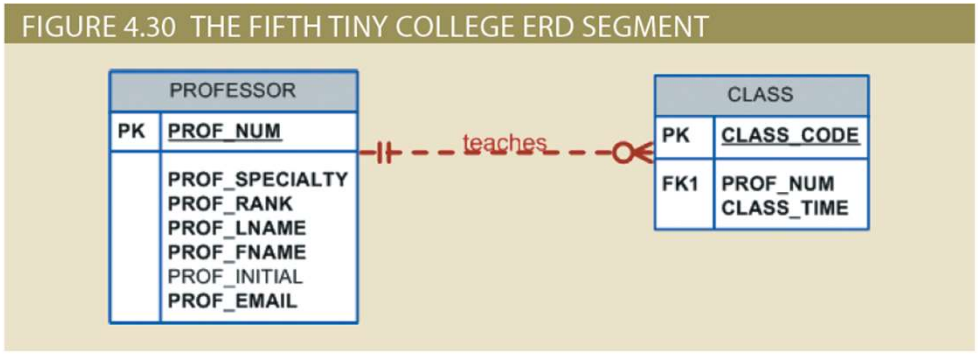
Figure 4.29 - The Fourth Tiny College ERD Segment



51

As stated in the slide and read the textbook for explanations.

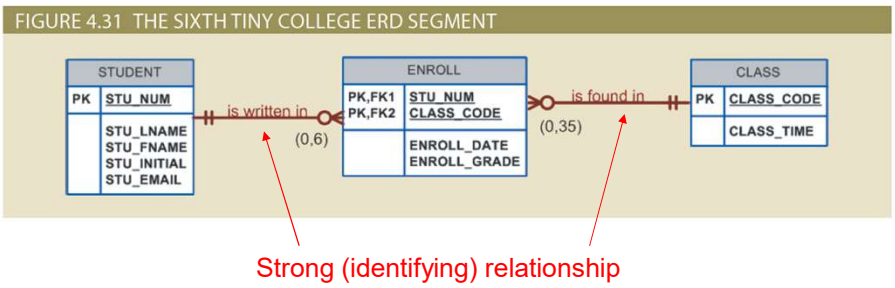
Figure 4.30 - The Fifth Tiny College ERD Segment



52

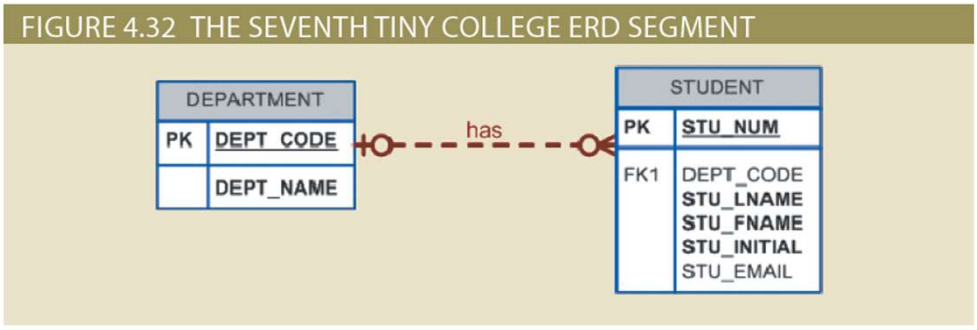
As stated in the slide and read the textbook for explanations.

Figure 4.31 - The Sixth Tiny College ERD Segment



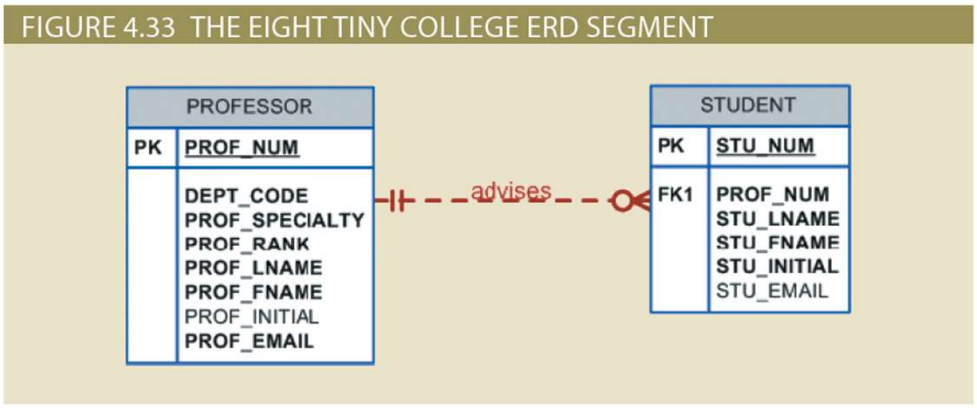
This is an example of Strong (identifying) relationships, which is when a Primary key of the related entity contains a primary key component of the parent entity, i.e. Student number in Enroll entity associates with student number from Student table, and class code in in Enroll entity associates with class code from class entity.

Figure 4.32 - The Seventh Tiny College ERD Segment



As stated in the slide and Read the textbook for explanations.

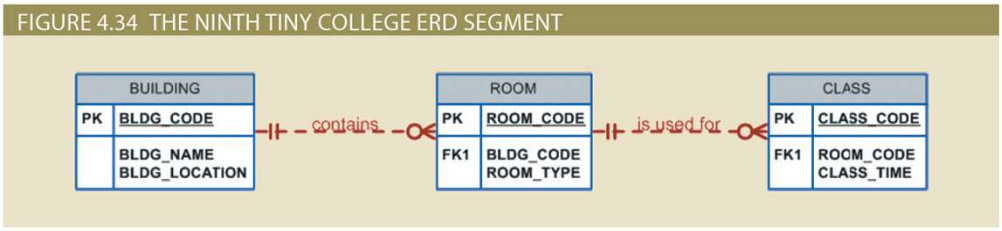
Figure 4.33 - The Eighth Tiny College ERD Segment



55

As stated in the slide and read the textbook for explanations.

Figure 4.34 - The Ninth Tiny College ERD Segment



As stated in the slide and Read the textbook for explanations.