COMP3311 Week 2

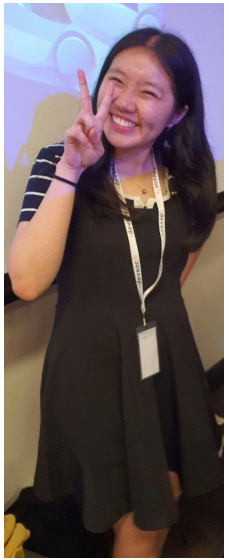
Welcome everyone!! :)

By Manhua Lu

Introductions

Manhua Lu 🐣 (not manga xd) manhualu.dev

- 4th year Computer Science and Commerce (Finance)
- Always on the hunt for new music 🎶, food 🥞 and cool photo opportunities 📸
- Executive Vice President of Course Projects in DevSoc <devsoc/> devsoc.app
- manhua.lu@student.unsw.edu.au



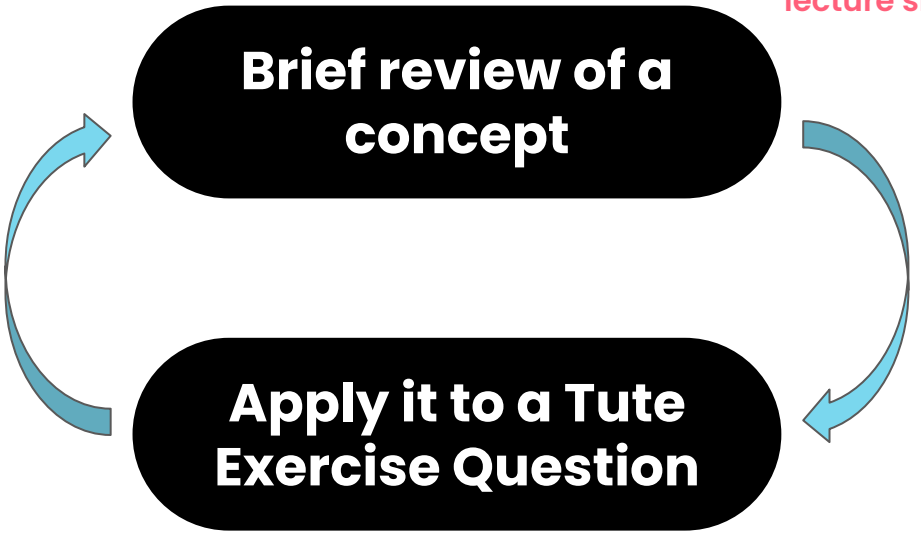
Your turn!

Please tell us your:

- Name
- What you're studying + year of study
- Any societies you've been involved with (you don't have to be a subcom in it, just attend their events)?

How will our Tutorials Work?

I will assume that you have at least looked through the lecture slides for the week 😳



Note: Unfortunately tutorials are not a replacement for the lectures!

How to HD COMP3311 😤

- Watch the **Topic Videos** every week before the live lectures!
    - In the live lectures, Armin goes through the concepts and does examples
- Attend the in-person **lectures** or watch the recording (ik it's hard 💀)
- Attend our **tutorials** every week!!
    - Please ask questions! I am here to help :)
- After the tutorial, attempt the **Tute Exercise questions** we didn't go through (solutions will be released the week after).
- Try to attempt the **Practices** that focus on concepts (not compulsory + not graded, but very helpful for assignments)

COMP3311 Assessment Weighting

Assessment Item	Weight	Relevant Dates
Quizzes - All topics **Assessment Format** Individual	12%	**Due Date** Week 2, 3, 4, 7, 8, 10
Assignment 1 - SQL/PLpgSQL **Assessment Format** Individual	13%	**Start Date** Week 3 **Due Date** Week 5
Assignment 2 - Python/SQL **Assessment Format** Individual	15%	**Start Date** Week 7 **Due Date** Week 9
Final Exam - All topics **Assessment Format** Individual	60%	**Start Date** Not Applicable **Due Date** Exam Period

Final exam has 40% hurdle

Announcements

- Quiz 1 due Friday 7 June @11:59pm AEST
  - Can make multiple submissions
  - In ‘Activities’ tab
- Have your PostgreSQL server set up by the end of Week 2

  (i.e. Practice Exercise 1)
  - Because you will need it for your assignment!

Setting up our PostgreSQL Server at CSE

COMP3311 24T1 — Database Systems

Prac Exercise 01

PostgreSQL: a client-server RDBMS

Last updated: **Saturday 17th February 8:07pm**

Most recent changes are shown in red ... older changes are shown in brown.

Aims

This exercise aims to get you to:

- install your PostgreSQL database server at CSE
- create, populate and examine a very small database

You ought to get it done by end of Week 2, since you'll need it to start working on Assignment 1.

Live demo...

Run these Commands Every Time to Start Server

1. Login to vxdb2:
   `ssh nw-syd-vxdb2`
2. Set up your environment:
   `source /localstorage/<your_zID>/env`
3. Start the PostgreSQL server:
   `p1`
4. Work with a database:
   `psql <someDatabaseName>`
5. Stop the PostgreSQL server:
   `P0`
6. Close SSH connection:
   `exit`

Learning Objectives

01 → Data Modelling (Q2)

02 → ER Model (Q6, Q10, Q11, Q14)
- Entities
- Attributes
- Relationships

03 → Relational Model (Q18,Q19)

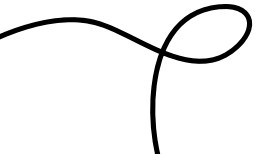
04 → ER to Relational Mapping (will be covered in Wk3 too)

01

Data Modelling

01 Data Modelling

2. In the context of database application development (aka “database engineering”), what are the aims of **data modelling**?



01 → Data Modelling

2. In the context of database application development (aka “database engineering”), what are the aims of **data modelling**?

Answer

Data modelling:

- Takes in **requirements** given by the client
- Builds a comprehensive description of the **entities** and its **attributes**
- It also shows the **relationship** among these entities.

The model constructed should ensure that all of the requirements can be met.

The data model should give enough detail that a database schema can be derived from it without significant additional information being needed.

COMP3311 covers 2 types of Database Models:

Entity Relational (ER) Model

Relational Model

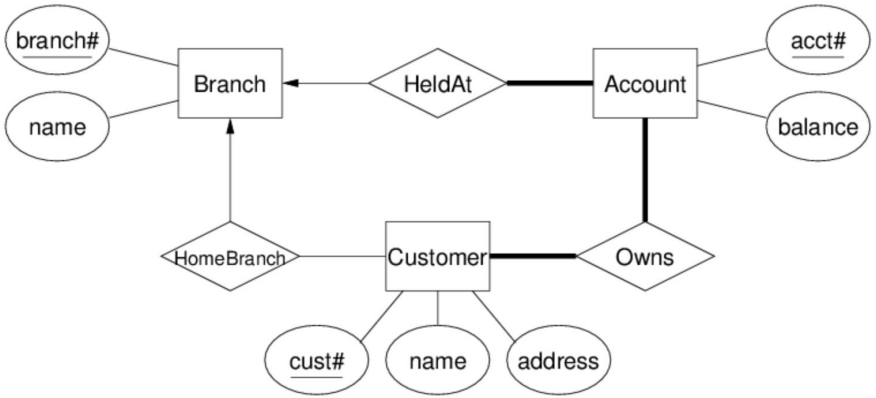
02

Entity Relational (ER) Model

02 → Entity Relational Model

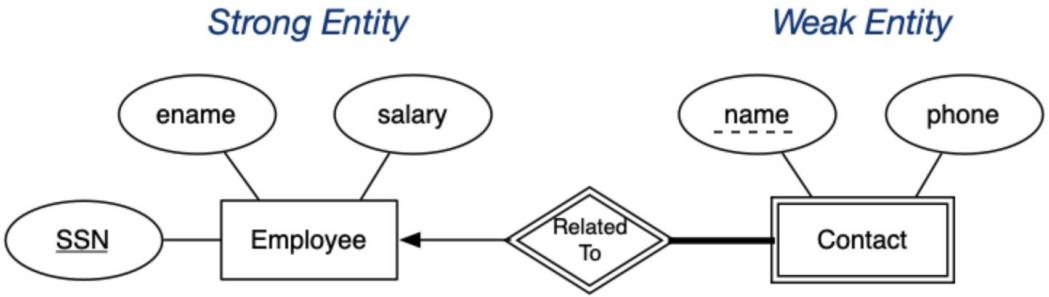
- A data model that consists of **entities** which have **attributes** and are **related** to each other.

- **Entities**: Objects (rectangle)
- **Attributes**: Properties (oval)
- **Relations**: Describe how entities interact with each other (diamond and arrows/lines)



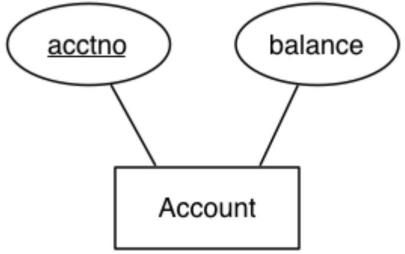
02 Entity Relational Model: Entities

- Can have **attributes**
- A **weak entity** exists only because of association with strong entities
  - They have no key of their own, instead have a discriminator **(dotted-underlined)**
  - Its primary key is made up of its **strong entity's primary key + its own discriminator**



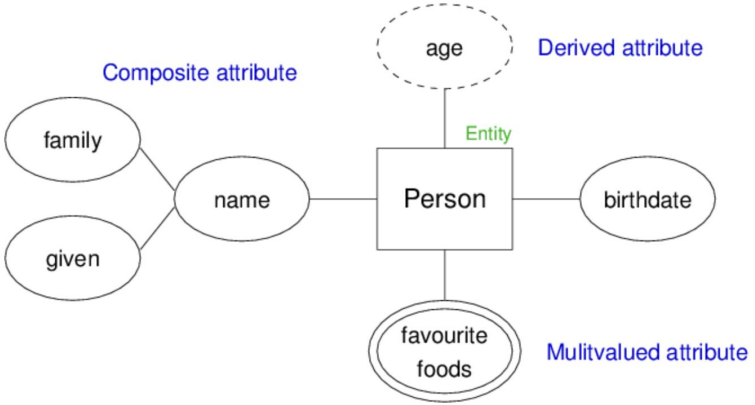
02 → Entity Relational Model: Attributes

- The attribute that is used to identify the entity is called the **primary key** and is **underlined**



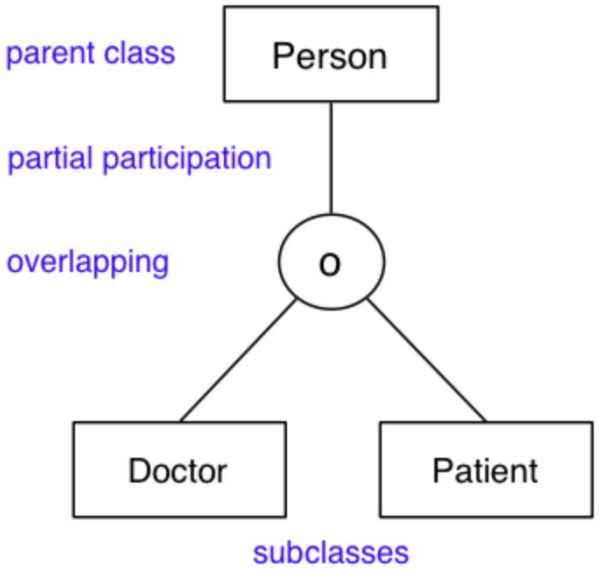
- An attribute can be **composite**, **derived** or **multi-valued**

Example of attribute notations:

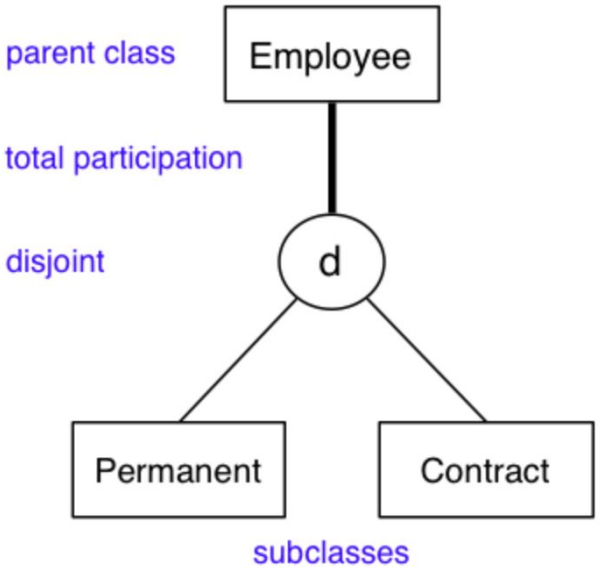


02 → Entity Relational Model: Subclasses + Inheritance

A person may be a doctor and/or may be a patient or may be neither

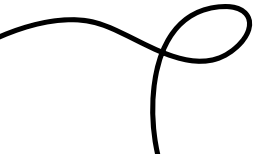


Every employee is either a permanent employee or works under a contract



02 → Entity Relational Model: Relationships

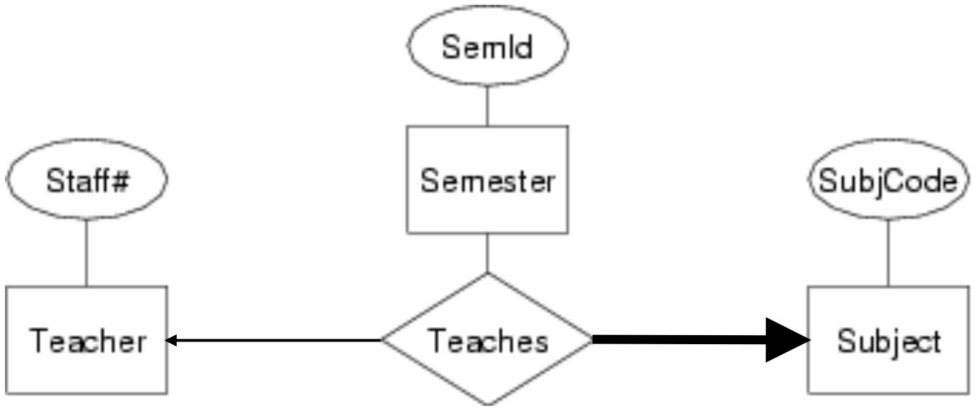
- Semantics:
  - One to one
  - One to many
  - Many to many
- Participation
  - Total (**thicc** line)
  - Partial



02 Entity Relational Model: Relationships

One to one

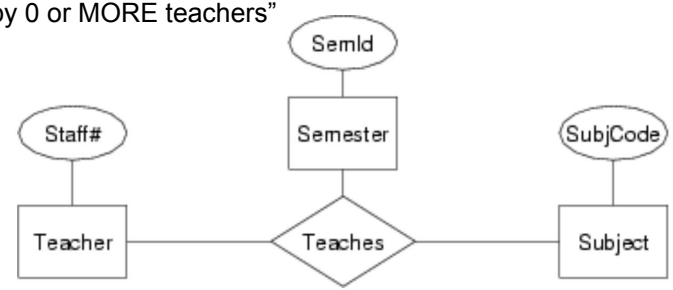
“A teacher can teach 0 or ONE subject, a subject must be taught by **ONE** person.”



02 Entity Relational Model: Relationships

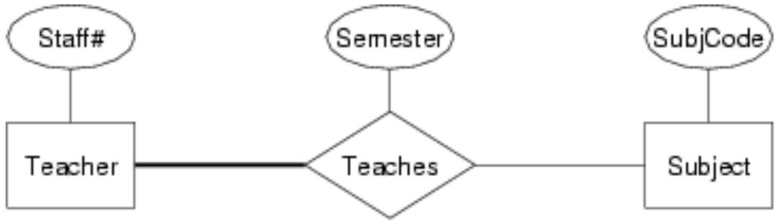
Many to many

“A teacher can teach 0 or MORE subjects, a subject can be taught by 0 or MORE teachers”



Many to many

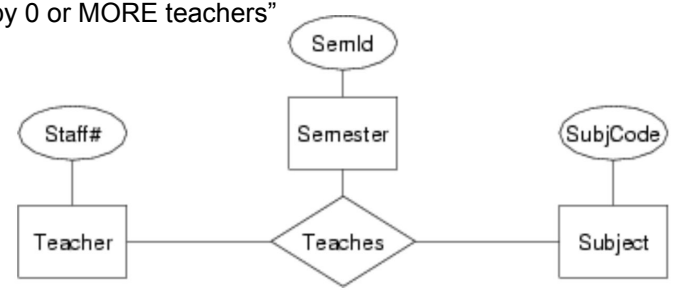
“A teacher must teach **ONE** or MORE subjects, a subject can be taught by 0 or MORE teachers”



02 Entity Relational Model: Relationships

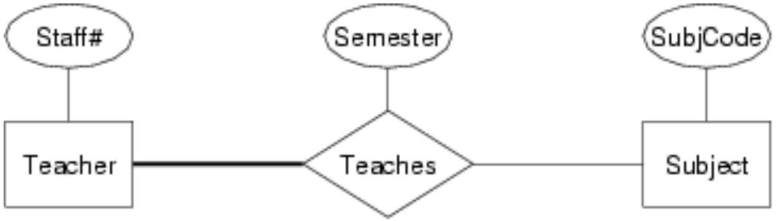
Many to many

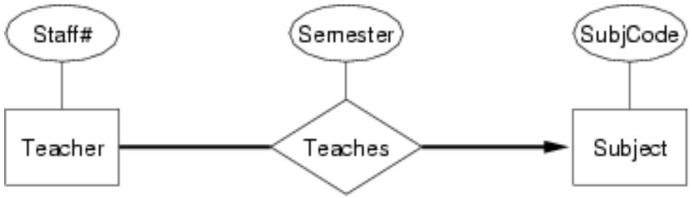
“A teacher can teach 0 or MORE subjects, a subject can be taught by 0 or MORE teachers”



Many to many

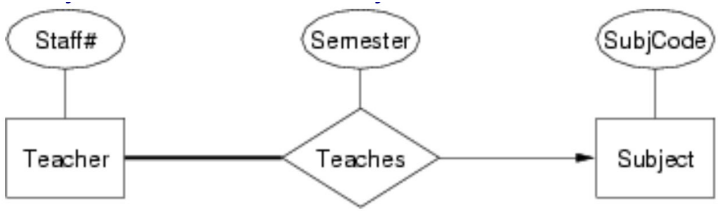
“A teacher must teach **ONE** or MORE subjects, a subject can be taught by 0 or MORE teachers”





One to many

“A teacher must teach only **ONE** subject, a subject must be taught by **ONE** or MORE teachers”

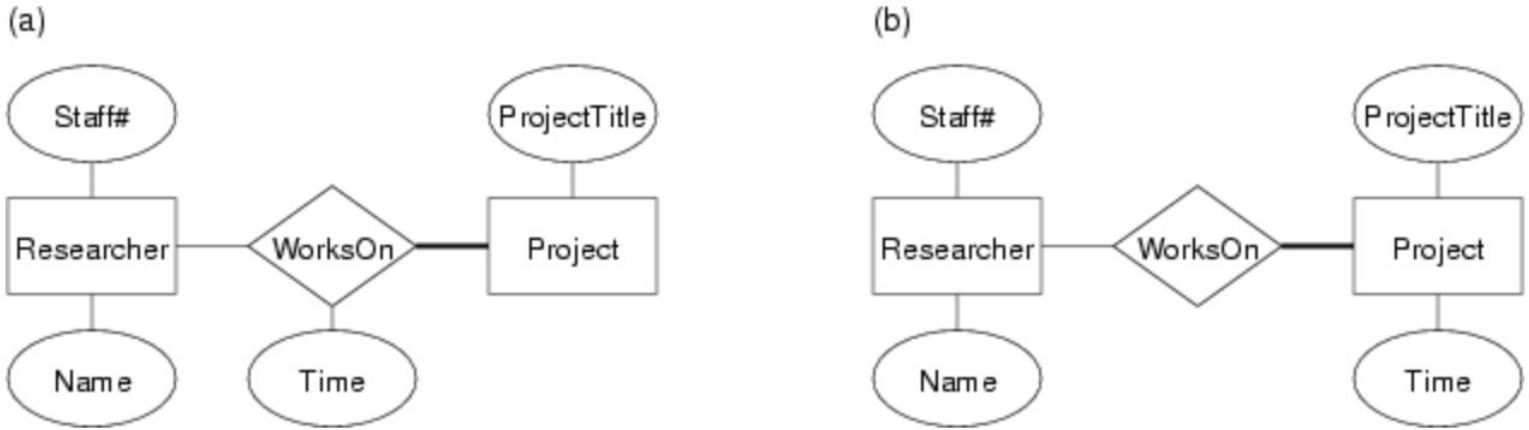


One to many

“A teacher must teach only **ONE** subject, a subject can be taught by 0 or MORE teachers”

02 Entity Relational Model: Relationships Question

6. Researchers work on different research projects, and the connection between them can be modelled by a `WorksOn` relationship. Consider the following two different ER diagrams to represent this situation.



Describe the different semantics suggested by each of these diagrams.

02 → Entity Relational Model: Relationships Question

Answer

Between the 2 diagrams, the *Time* attribute is in different places.

a) We can get how much time each researcher spent on a particular project.
b) We can only get the total amount of time everyone has spent on a project. We don't know how much time each researcher spent on a project.

Both diagrams are saying: A researcher can work on 0 or more projects, a project must be worked on by **ONE** or MORE researchers.

02 → Entity Relational Model Question

10. Assume there is a `Person` entity type. Each person has a home address. More than one person can live at the same home address.

    a. Create two, different ER diagrams to depict `Persons` and their addresses, one with `Address` as an attribute, the other with `Address` as an entity.

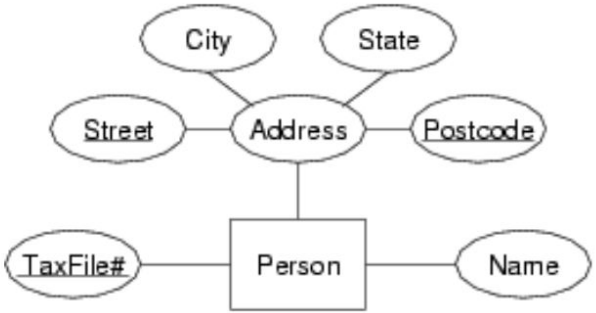
    b. Why would we choose one rather than the other?

    c. Assume that we have a `ElectricCompany` entity type. Only one of these companies supplies power to each home address. Add that information to each ER diagram.

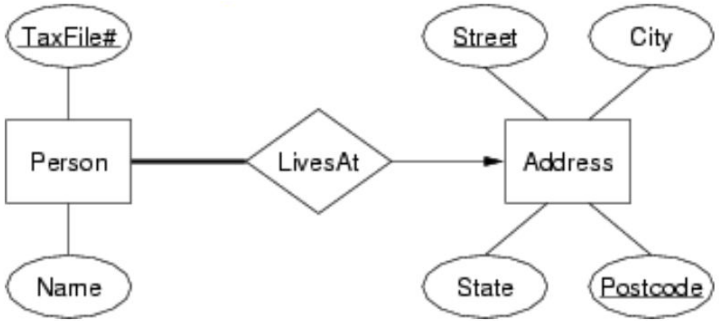
02 → Entity Relational Model Question

Answer for a)

Address as attribute



Address as entity



02 → Entity Relational Model Question

Answer for b)

The second diagram is generally better because an Address could have multiple Persons.

> In the second diagram, we wouldn't have to repeat storing the address attribute every time if multiple Persons live at the Address.

It is also more flexible for if we wanted to add things like c) asks for

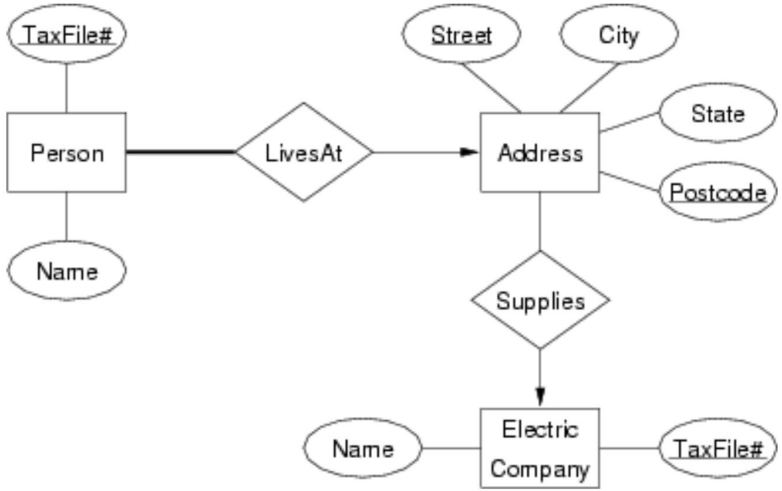
02 → Entity Relational Model Question

Answer for c)

When Address is an attribute, it is not possible to relate electric companies directly to an Address.

- Instead they can only be related to Person and through them indirectly related to addresses

When Address is an entity, it is possible.



02 → Entity Relational Model Question

11. [Based on GUW 2.1.3] Give an ER design for a database recording information about teams, players, and their fans, including:

    a. For each team, its name, its players, its captain (one of its players) and the colours of its uniform.

    b. For each player, their name and team.

    c. For each fan, their name, favourite teams, favourite players, and favourite colour.

(Skip if no time in tute but heavily recommend you do this in your own time!)

02 → Entity Relational Model Question

14. Give an ER design to model the following scenario ...

- for each person, we need to record their tax file number (TFN), their real name, and their address
- everyone who earns money in Australia has a distinct tax file number
- authors write books, and may publish books using a ``pen-name'' (a name which appears as the author of the book and is different to their real name)
- editors ensure that books are written in a manner that is suitable for publication
- every editor works for just one publisher
- editors and authors have quite different skills; someone who is an editor cannot be an author, and vice versa
- a book may have several authors, just one author, or no authors (published anonymously)
- every book has one editor assigned to it, who liaises with the author(s) in getting the book ready for publication
- each book has a title, and an edition number (e.g. 1st, 2nd, 3rd)
- each published book is assigned a unique 13-digit number (its ISBN); different editions of the same book will have different ISBNs
- publishers are companies that publish (market/distribute) books
- each publisher is required to have a unique Australian business number (ABN)
- a publisher also has a name and address that need to be recorded
- a particular edition of a book is published by exactly one publisher

State all assumptions used in developing your data model.

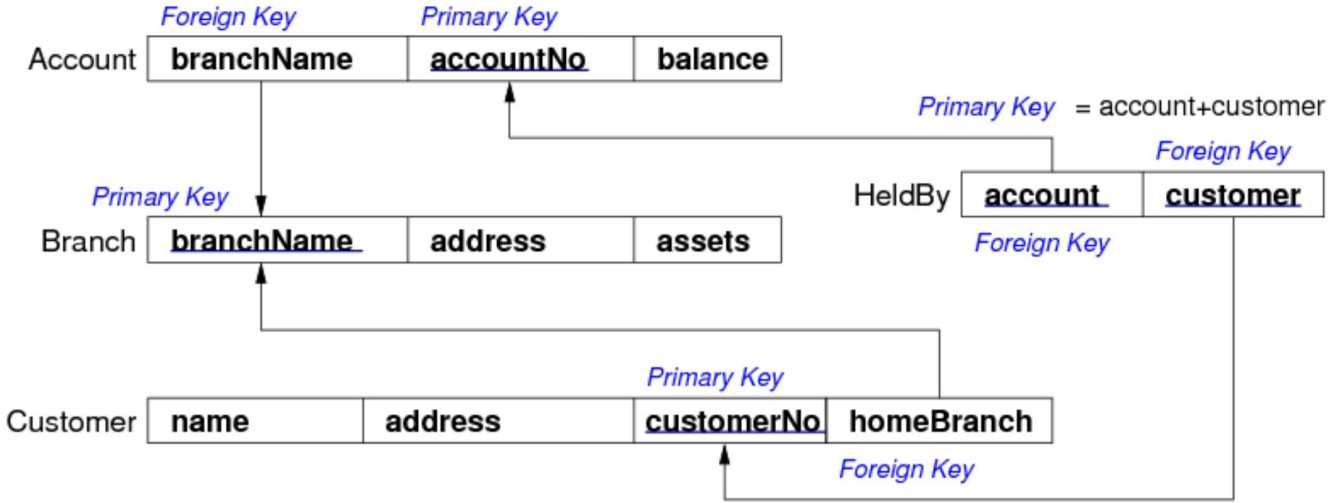
(Skip if no time in tute but heavily recommend you do this in your own time!)

03

Relational Model

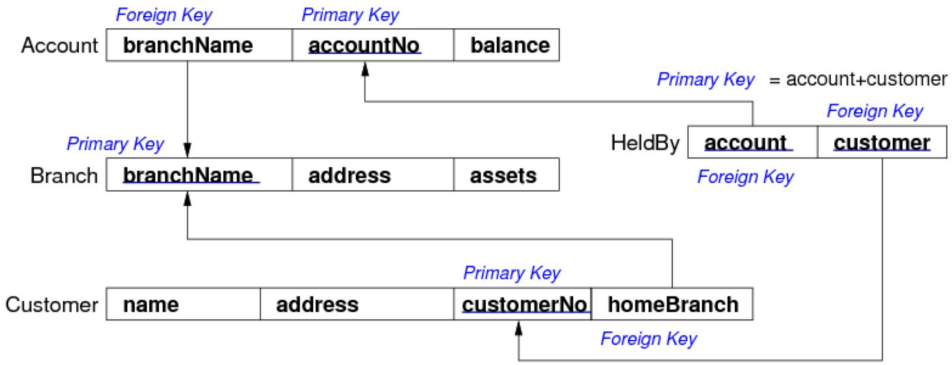
03 Relational Model

- Model data as a set of **relations** with attributes.
- The general process is to create a relational model from ER diagram.



03 Relational Model

- **Attributes**: columns
- **Domain**: the set of values which an attribute can take e.g. integer, string, date
- **Tuples/Records**: rows
- **Relation**: table
- **Primary key (underlined)**: attribute that uniquely identifies each tuple
- **Foreign key**: a referencing mechanism that allows us to link 2 tables together



03 → Relational Model Question

18. Consider the following simple relational schema:

```
R(a1, a2, a3, a4)
S(b1, b2, b3)
```

which of the following tuples are not legal in this schema? Explain why the iillegal tuples are invalid.

```
R(1, a, b, c)  R(2, a, b, c)  R(1, x, y, z)

R(3, x, NULL, y)  R(NULL, x, y, z)

S(1, 2, x)  S(1, NULL, y)  S(2, 1, z)
```

03 → Relational Model Question

19. Consider the following relations which form a tiny part of the schema for the MyUNSW database:

```
Person(zID, zPass, familyName, givenName, dateOfBirth, countryOfBirth, ...)
Student(zID, degreeCode, WAM, ...)
Staff(zID, office, phone, position, ...)
Course(cID, code, term, title, UOC, convenor)
Room(rID, code, name, building, capacity)
Enrolment(course, student, mark, grade)
```

Identify all of the primary keys and foreign keys, and suggest suitable domains for each attribute. You can introduce new relations if you think would likely be used to represent objects not in the current tables. Discuss which attributes could have NULL values, and the circumstances under which this might occur.

04

ER to Relational Mapping

04 → ER to Relational Mapping

- Entities become tables
- **Many to many** relations become tables
- **One to one** relations collapse into either entity
- **One to many**, put the foreign key of the ‘one’ side in the ‘many’ entity

(We will do questions on this in Wk3, so no stress if you don’t get it just yet!!)