Week 2 Tutorial Notes

▼ Agenda

- Q1 ... ice-breaker
- Q7 ... relatively straightforward ER data modelling exercise
- Q9 ... different relationships in ER
- Q15 ... a substantial ER data modelling exercise (if you have time)
- Q16 .. to ensure they understand these core relational concepts

Pre-tute

- Introductions 🙂
- Discourse!!
- Quiz due Friday 11:59pm

Course Outline

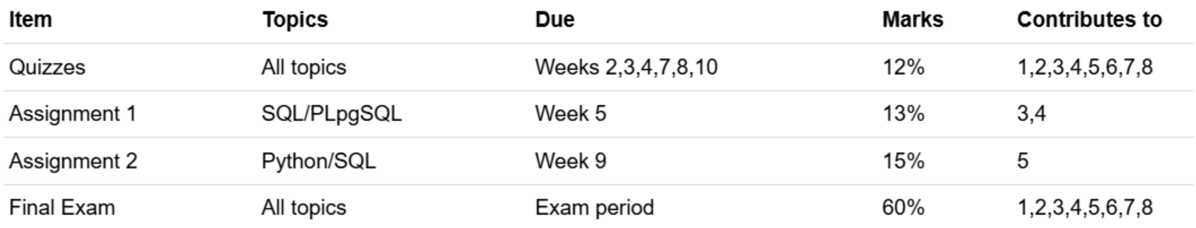
Content

- Topic Videos
- Lectures
- Textbooks (OPTIONAL)

Activities

- Tutorials
- Prac exercises
- **Quizzes**
- **Assignments**
- **Exam**

▼ Mark Distribution



Item	Topics	Due	Marks	Contributes to
Quizzes	All topics	Weeks 2,3,4,7,8,10	12%	1,2,3,4,5,6,7,8
Assignment 1	SQL/PLpgSQL	Week 5	13%	3,4
Assignment 2	Python/SQL	Week 9	15%	5
Final Exam	All topics	Exam period	60%	1,2,3,4,5,6,7,8

Contact

- Administrative
  - cs3311@cse.unsw.edu.au
- Technical / Coursework
  - Discourse!!
  - z5419507@ad.unsw.edu.au

Setup

1. Access the vxdb02 server
   - from Vlab:

     ```
     $ ssh vxdb02
     ```

   - from Home:

     ```
     $ ssh YourUserName@vxdb02.cse.unsw.edu.au
     ```

2. (If first time running)

   ```
   $ 3311 pginit
   ```

3. Tell vxdb02 the source of the psql server and start the server

```
$ source /localstorage/$USER/env
$ p1
```

4. Work with a specific database

   ```
   $ psql SomeDatabase
   ```

5. Stop PSQL server

   ```
   $ p0
   ```

After this, look into ‘help’ command

Data Modelling

Aims to:

- describe what **information** is contained in the database
- describe **relationships** between data items
- describes **constraints** on data

Exercise: Let’s try brainstorming some of these scenarios. Some examples are:

- Instagram
- Gmail
- UNSW Handbook

Entity-Relationship (ER) Data Modelling

ER has three major modelling constructs:

▼ attribute

data item describing a property of interest

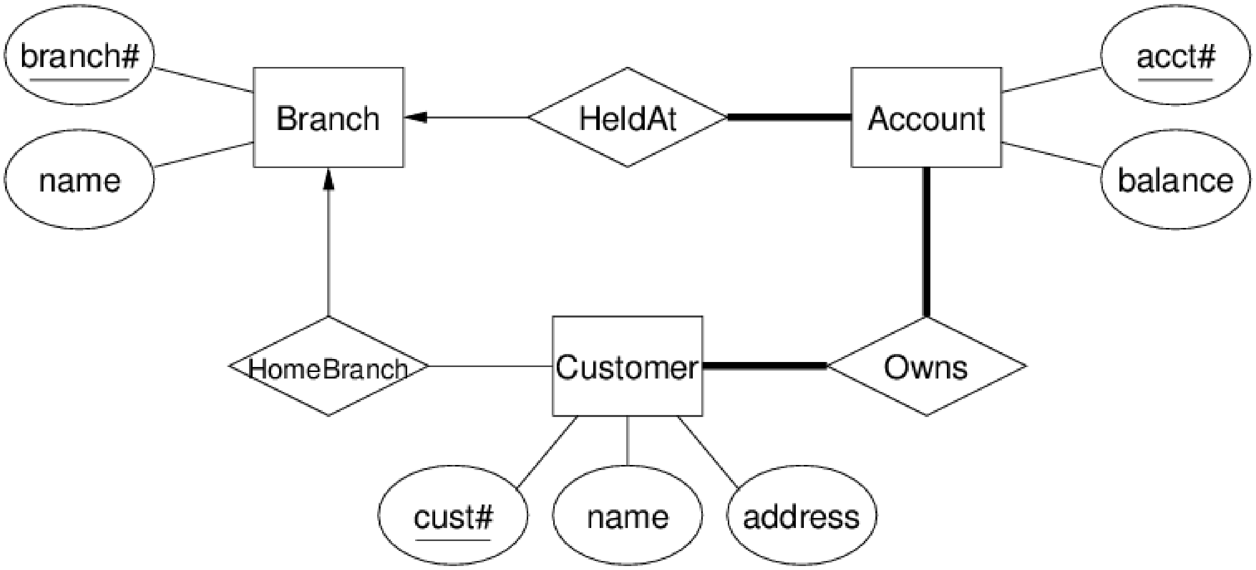
▼ entity

collection of attributes describing object of interest

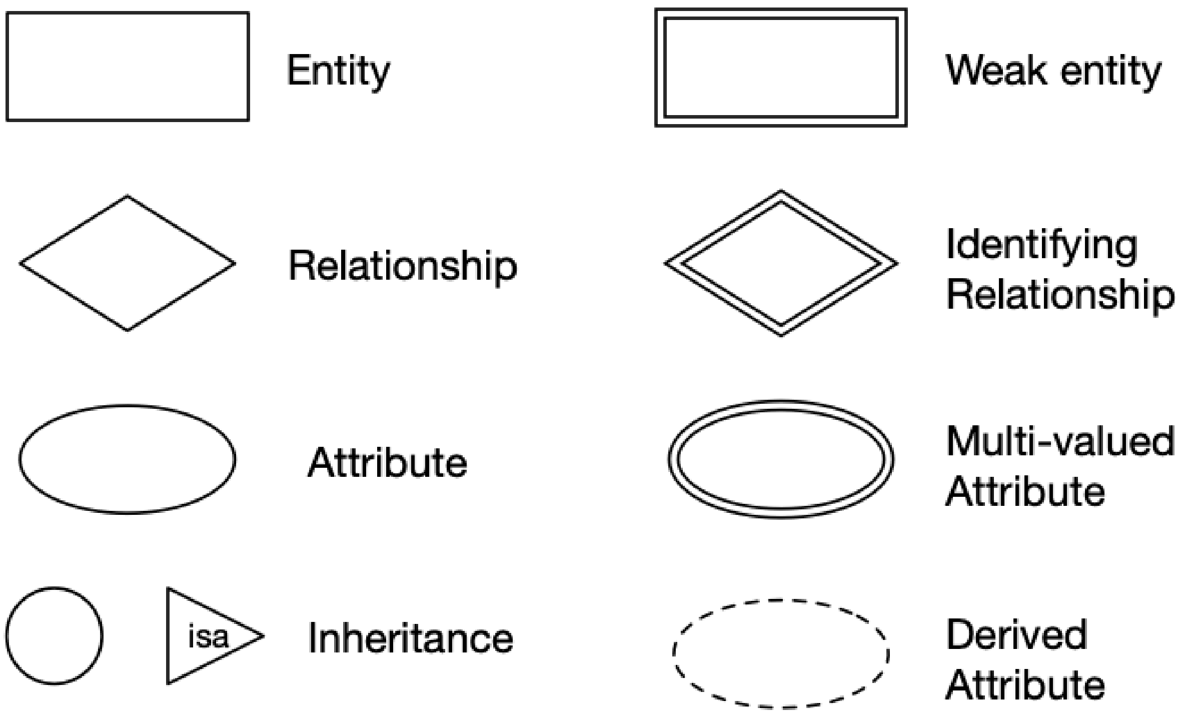
▼ relationship

association between entities (objects)

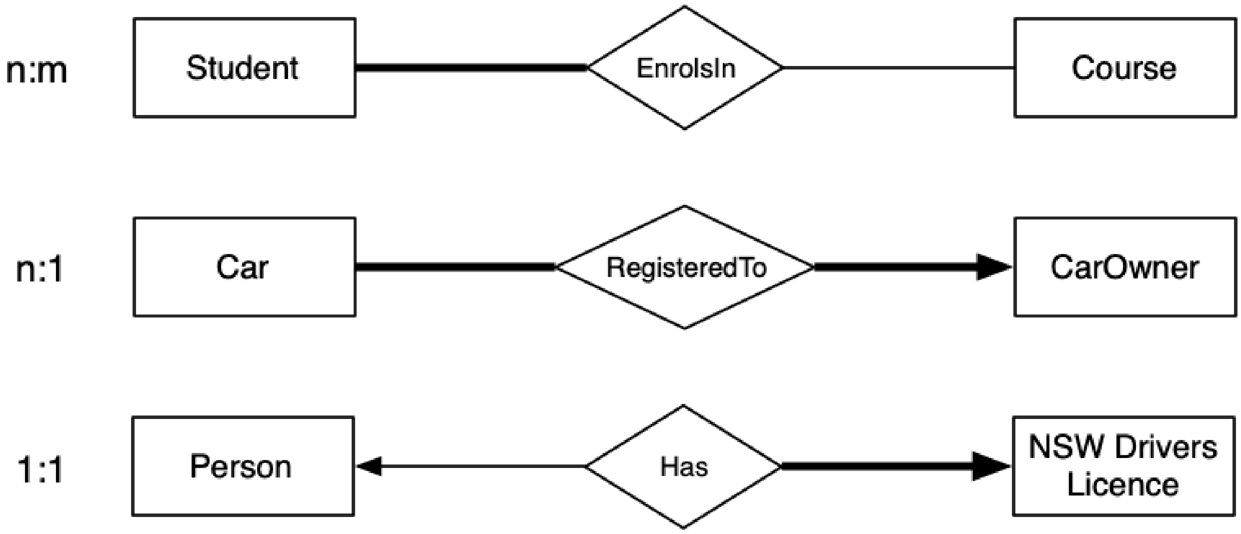
Example: Bank Account



▼ ER design elements

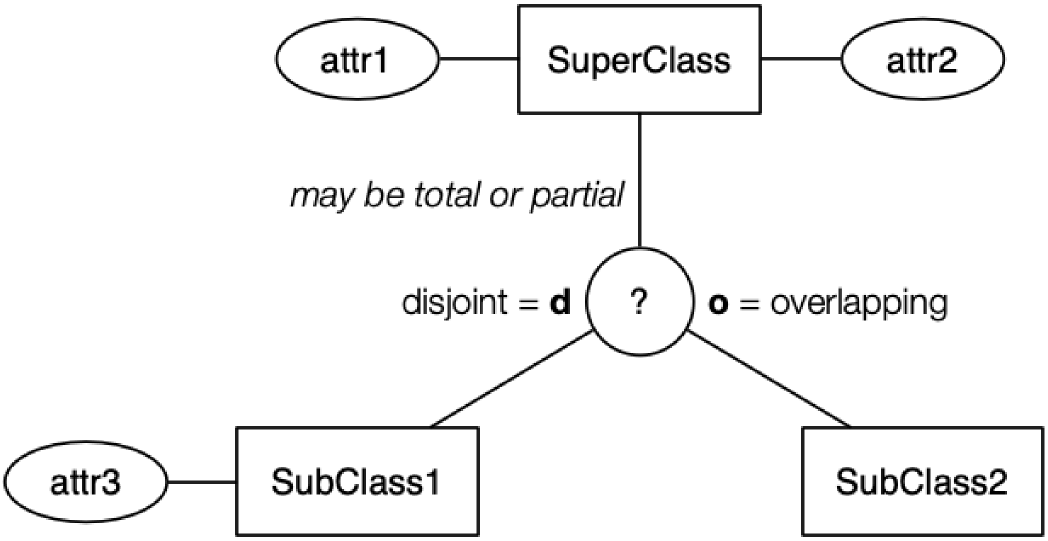


ER Relationships

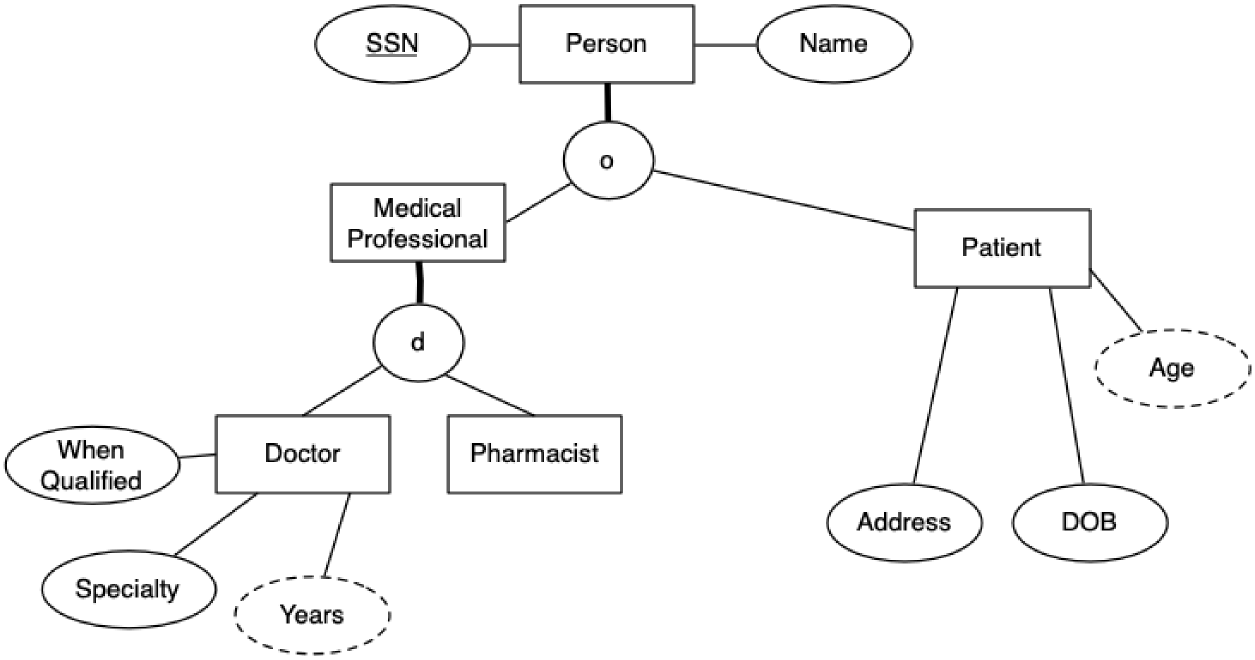


Thick line = total participation; thin line = partial participation

ER Class Hierarchies



Example: Medical Information



Relational Data Model

A collection of inter-connected **relations** (that look awfully close to tables)

Each
relation (denoted R,S,T,...) has:

- a name (unique within a given database)
- a set of attributes (which can be viewed as column headings)

Each **attribute** (denoted A,B,... or a_1, a_2,...) has:

- a name (unique within a given relation)
- an associated domain (set of allowed values)

Relation schema of R: R($a_1 : D_1, a_2 : D_2,\ ...\ , a_n : D_n$)

Tuple of R: an element of $D_1 \times D_2 \times ... \times D_n$ (i.e. list of values)

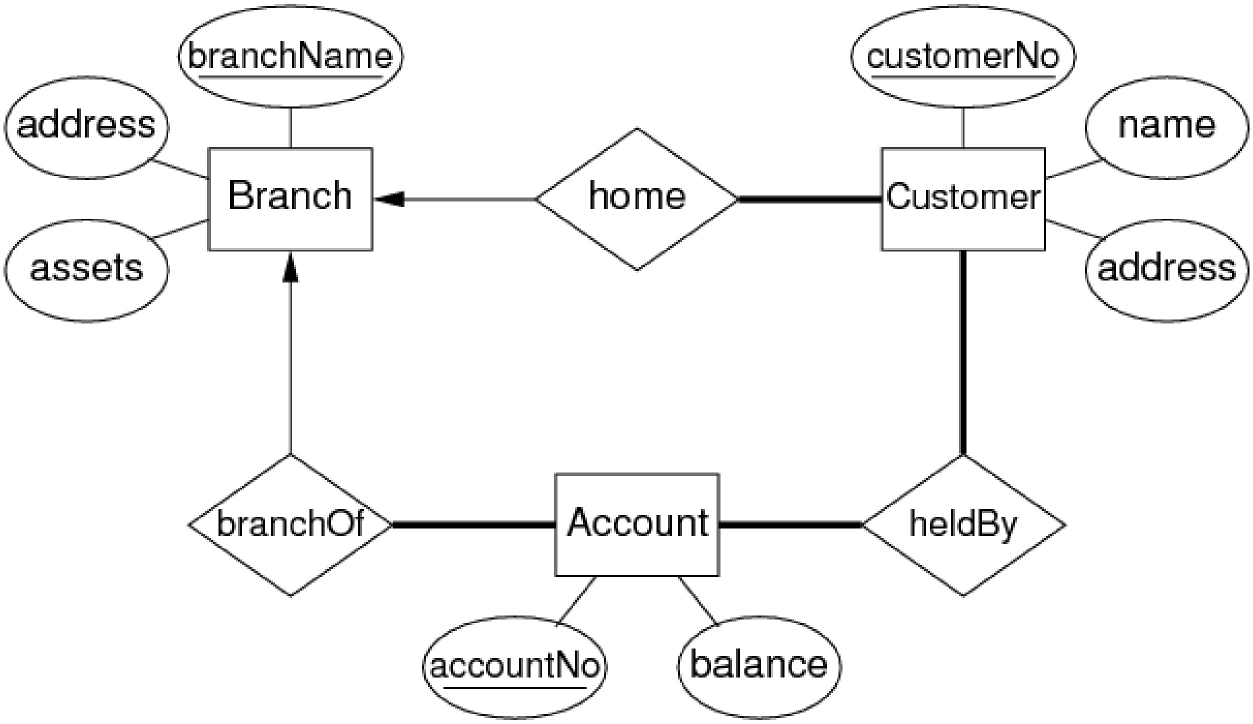
Instance of R: subset of $D_1 \times D_2 \times ... \times D_n$ (i.e. set of tuples)

Example: Bank Account

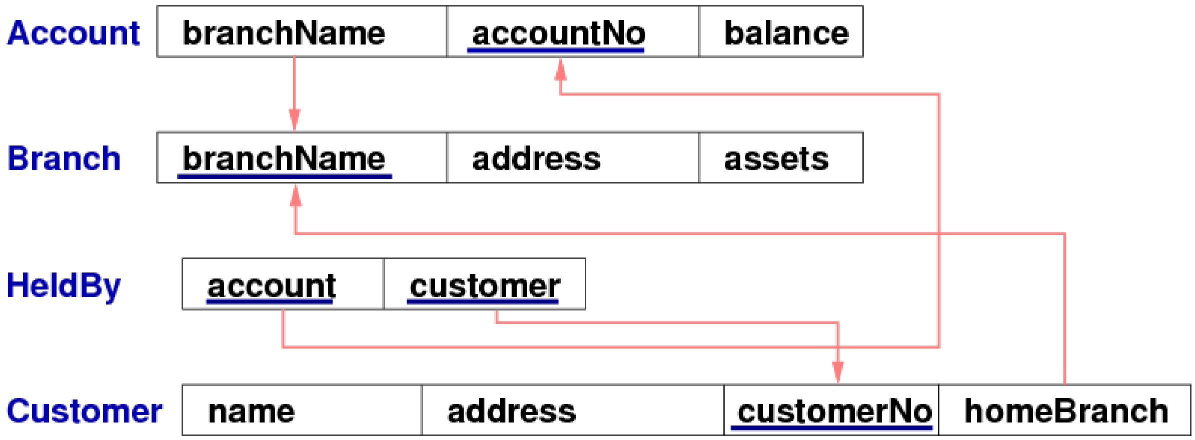
A relation: **Account(branchName, accountNo, balance)**. An *instance* of this relation:

```
{
  (Sydney, A-101, 500),
  (Coogee, A-215, 700),
  (Parramatta, A-102, 400),
  (Rouse Hill, A-305, 350),
  (Brighton, A-201, 900),
  (Kingsford, A-222, 700)
  (Brighton, A-217, 750)
}
```

ER → Relational



▼ Relational. **Identify the Primary and Foreign Keys**



SQL DDL

SQL data definition language (DDL) is the formal way of describing the above relational schemas. The primary SQL DDL construct is table creation

```
create table TableName (
  attr1Name  type [constraints],
  attr2Name  type [constraints],
  attr3Name  type [constraints],
  ...
  primary key (attrxName ),
  foreign key (attryName)
          references OtherTable (attrzName )
);
```