

Week 2 Tutorial Notes

▼ Agenda

- Q4 → after 'Data Modelling'
- Q7 ... relatively straightforward ER data modelling exercise
- Q8 → after discussing multivalued attributes
- Q9 ... different relationships in ER
- Q11 → first ER for class to try
- Q15 ... a substantial ER data modelling exercise (if you have time)

Pre-tute

- Introductions 😊
- Discourse!!
- Quiz due Friday

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 10	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!!
- If you're unsure:
 - z5419507@ad.unsw.edu.au

Setup

Find more detailed instructions in Prac Exercise 01

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

Example - describe the data, relationships and constraints in the following:

- Instagram
 - Gmail



Try it yourself: tutorial Q4

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

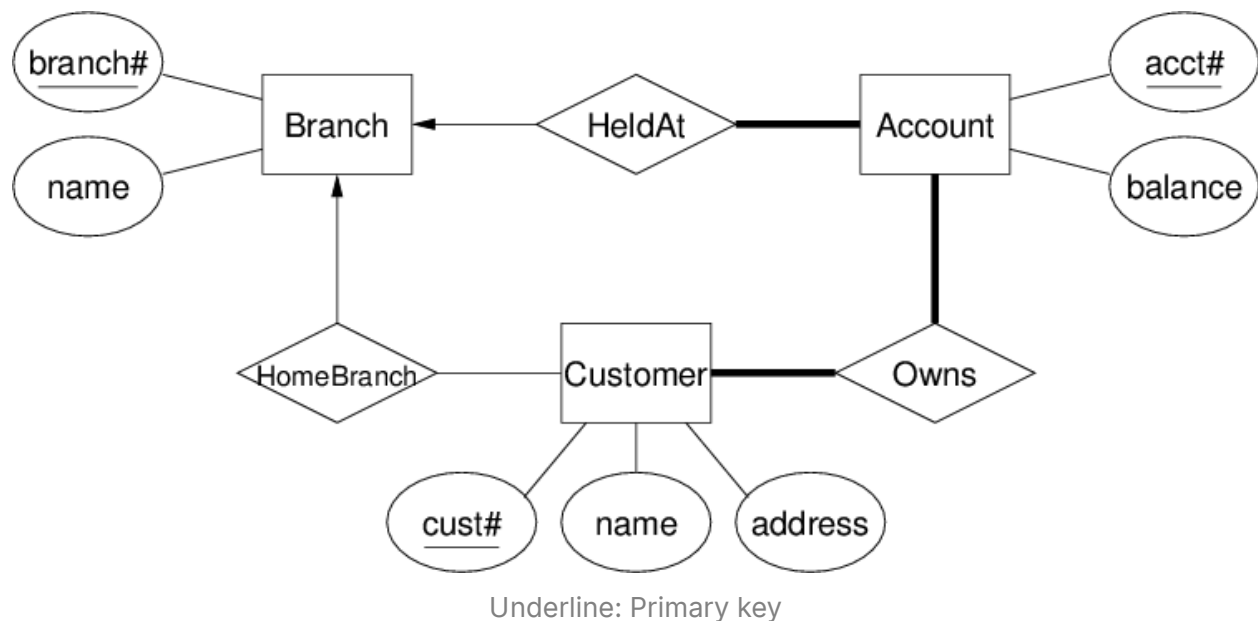
▼ Relationship

Association between entities (objects)

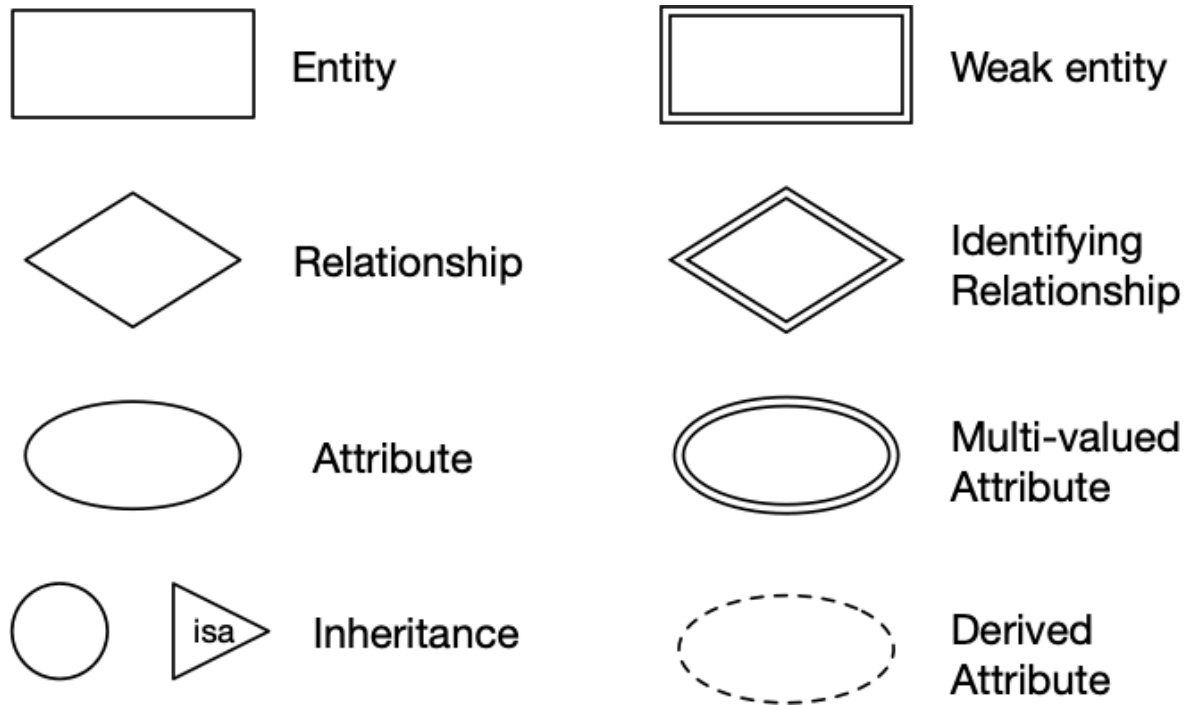


Attributes are atomic values. Entities are collections of attributes.

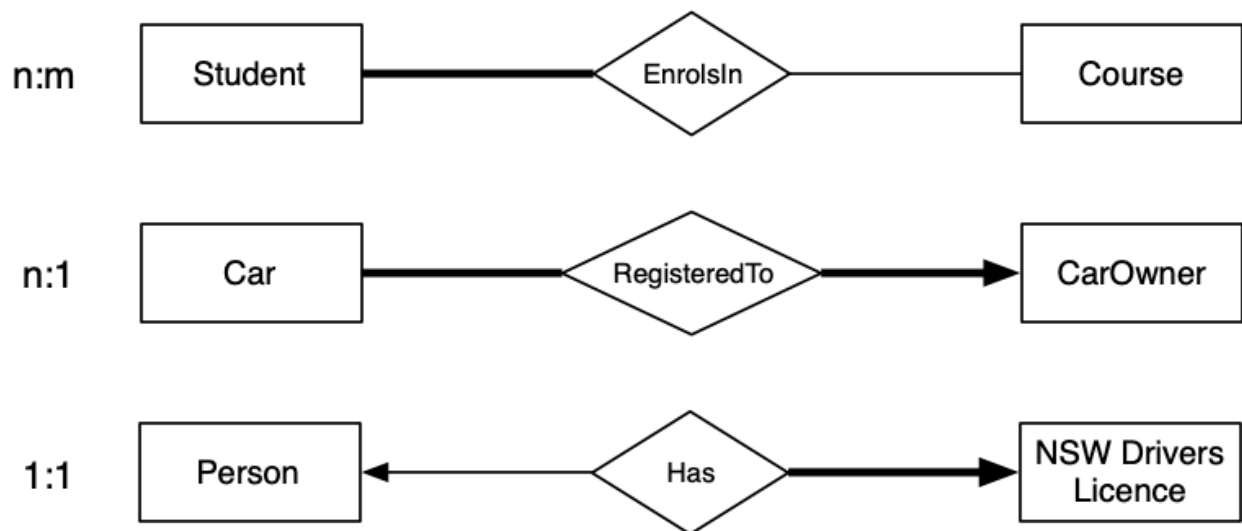
Example: Bank Account



▼ ER design elements



ER Relationships

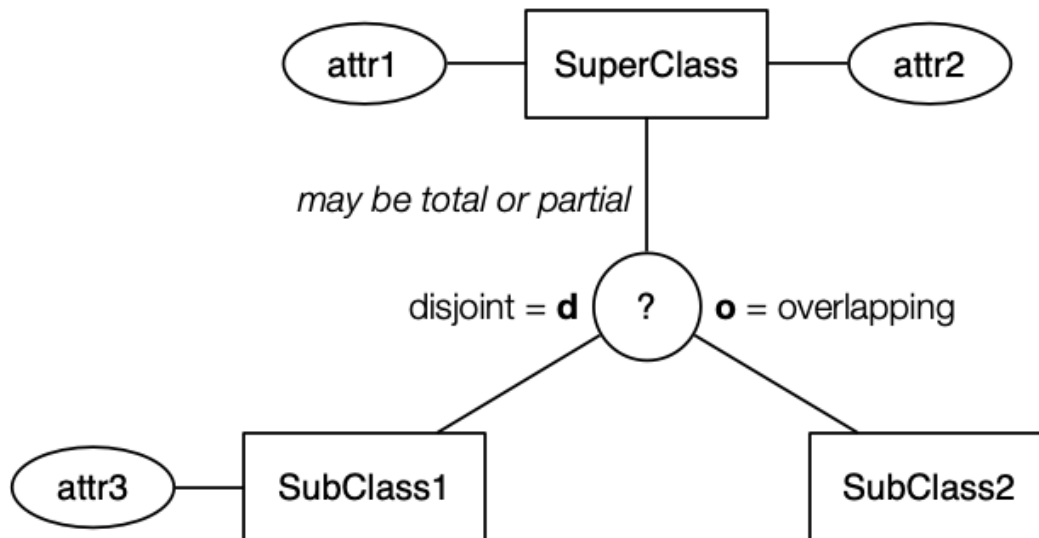


Thick line = total participation; thin line = partial participation

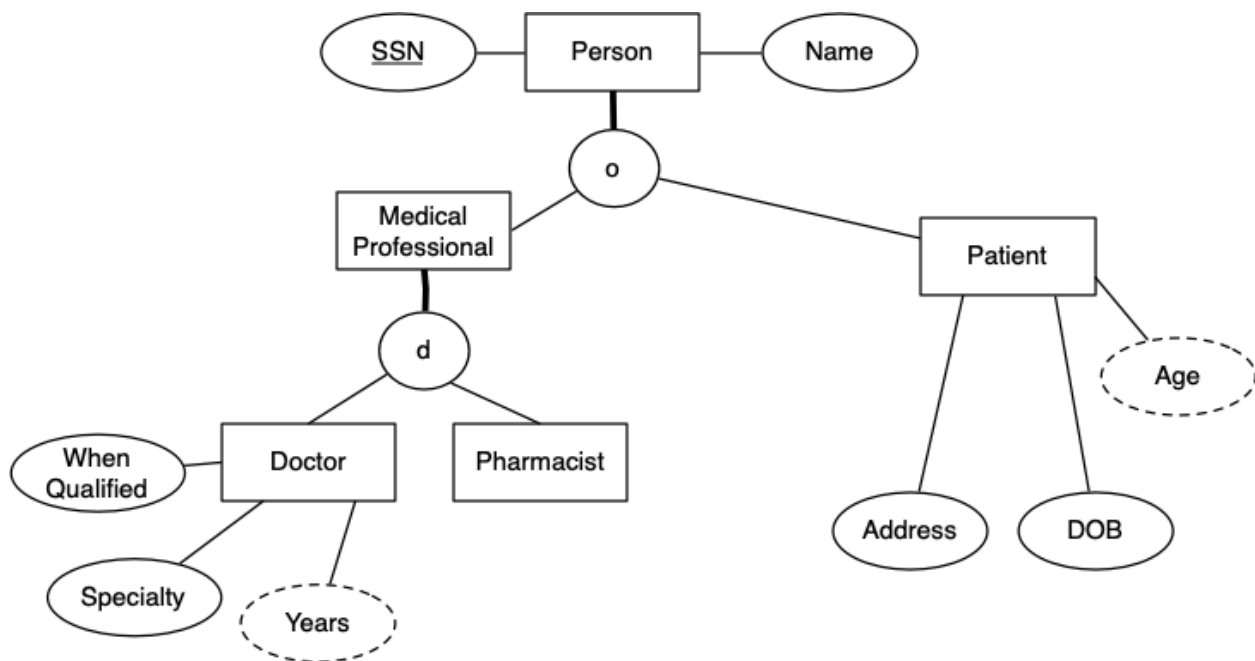


Try: tutorial Q7, Q8, Q9

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, \dots) has:

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

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

Relational schema: a collection of relation schemas

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

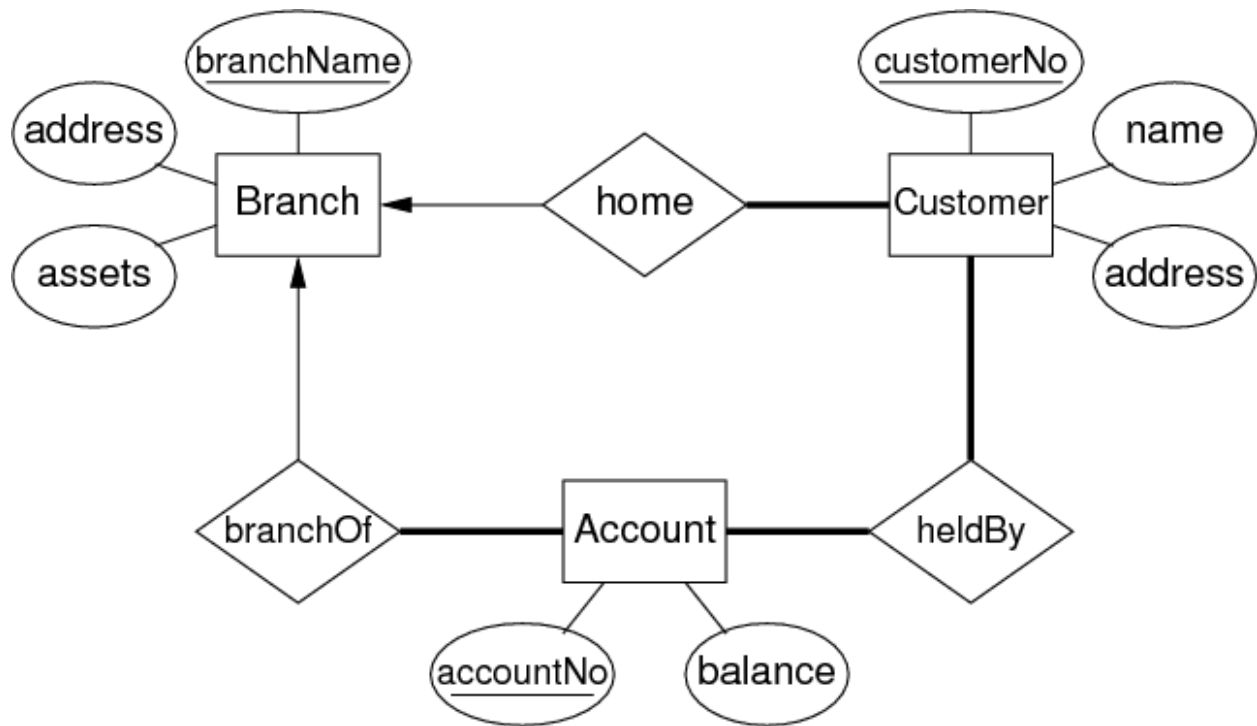
Instance of R: subset of $D_1 \times D_2 \times \dots \times D_n$ (i.e. set of tuples)

Example: Bank Account

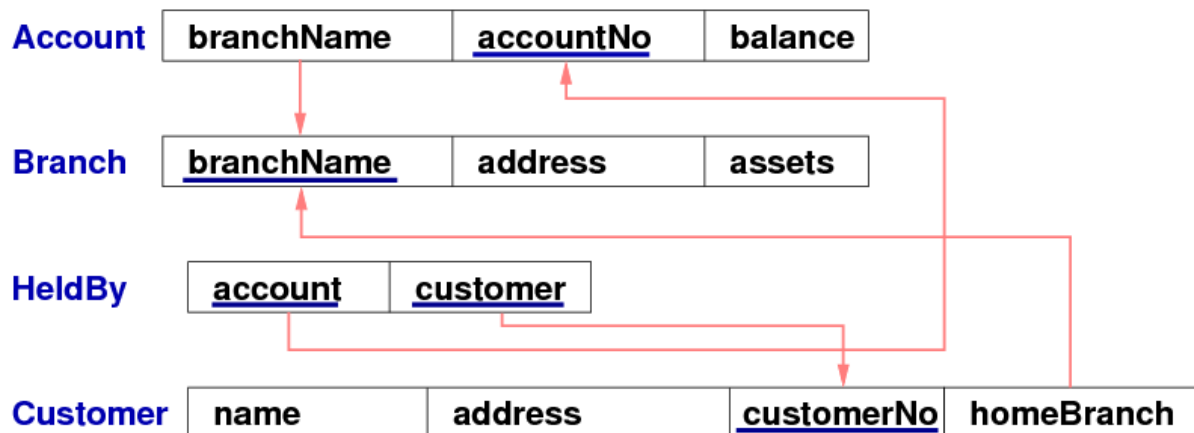
A relation: **Account(branchName, accountNo, balance)**. An *instance* of this relation:

```
{  
  (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 )  
);
```

Let's apply what we've learnt!



11, 15