COMP3311 Week 2 Monday Lecture

- Week 02
- CSE Environment
- Recap
- ER to Relational Mapping
- SQL Schemas
- SQL Types
- SQL Constraints
- Exercise: Constraints
- Mapping Strong Entities
- Mapping Weak Entities
- Mapping N:M Relationships
- Mapping 1:N Relationships
- Mapping 1:1 Relationships
- Mapping n-way Relationships
- Mapping Composite Attributes
- Mapping Multi-valued Attributes (MVAs)

❖ Week 02

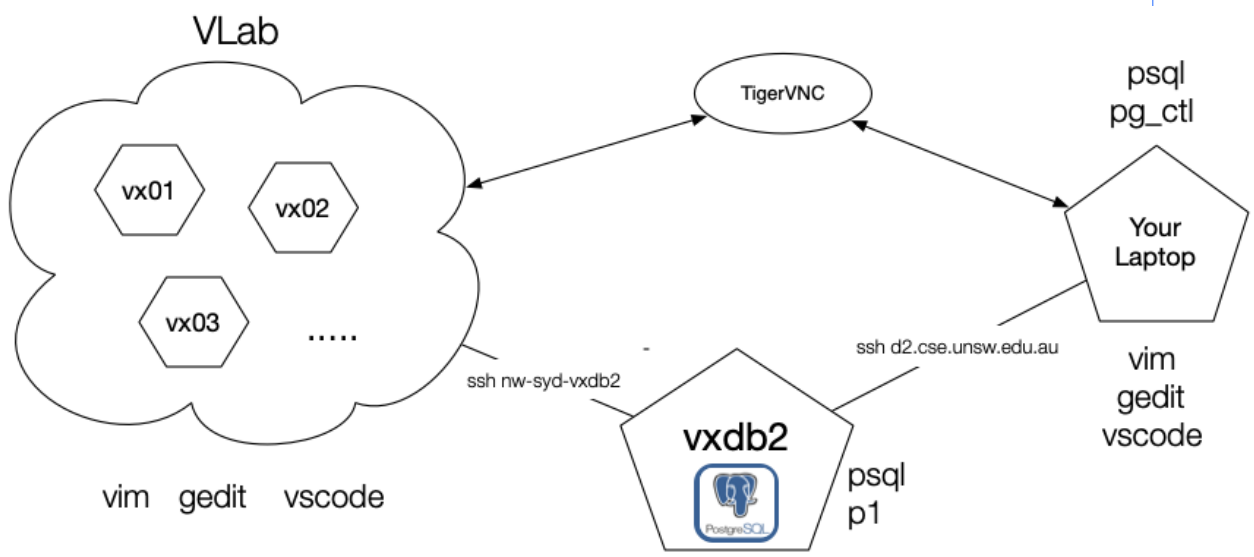
In today's lecture ...

- More SQL Data Definition Language (DDL)
- More Mapping ER → Relational/SQL

Things to do ...

- Quiz before Friday midnight
- SES survey before October 9
- Set up your PostgreSQL server
  (170 students have logged in to db2 and have /localstorage)

❖ CSE Environment



❖ Recap

Entity-relationship data model

- entities, attributes, relationships, subclasses
- relationship variations: total/partial, n:m, 1:n, 1:1

SQL as an implementation of relational data model

- relations → tables, tuples → tuples, attributes → columns/fields

Mapping ER to SQL

- entity sets → tables, entities → tuples, attributes → fields
- relationships → tables or foreign keys
- multi-valued-attributes/weak-entities/subclasses → ?

❖ ER to Relational Mapping

Reminder: a useful strategy for database design:

- perform initial data modelling using ER
  (conceptual-level modelling)
- transform conceptual design into SQL relational model
  (implementation-level modelling)

A formal mapping exists for ER model →SQL/Relational model.

This maps "structures"; but additional info is needed, e.g.

- concrete domains for attributes and other constraints

But can't map some things (e.g. n:m total participation)

❖ SQL Schemas

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 )
);
```

SQL schema = collection of table definitions, including constraints

❖ SQL Types

Built-in types

- numeric: **`integer`**, **`numeric(`*`n`*`)`**, **`real`**
- strings: **`char(`*`n`*`)`**, **`varchar(`*`n`*`)`**, **`text`**
- time: **`date`**, **`time`**, **`timestamp`**, **`interval`**
- **`boolean`**, monetary, geometric, enumerated, ...

Make your own

```
create domain Dom as Type Constraint;
create type Name as enum (val1, val2,...);
```

❖ SQL Constraints

Constraints in SQL DDL

- on attributes e.g. **`integer`**, **`check (x > 0)`**, **`not null`**
- on table e.g. **`unique`**, **`primary key (a,b,c)`**
- between tables e.g. **`foreign key (x) references T(y)`**

Tuples which do not satisfy constraints cannot be added to DB

Gives strong guarantee that the data is valid (internally consistent)

But does not guarantee that it reflects reality

❖ Exercise: Constraints

Constraint = SQL expression limiting possible values

Define type + constraints for

- positive integers
- marks (range 0..100)
- unsw course codes (COMP3311)
- person's name (alpha + space + - + ')

❖ Mapping Strong Entities

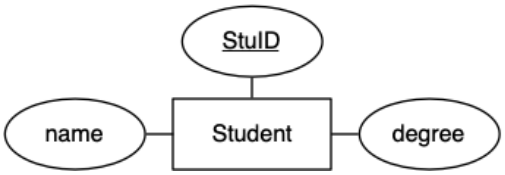
An entity set *E* with atomic attributes $a_1, a_2, \ldots a_n$

maps to

A relation *R* with attributes (columns) $a_1, a_2, \ldots a_n$

Example:

ER Model



Relational Version

Student

StuID	name	degree

Note: the key is preserved in the mapping.

❖ Mapping Weak Entities

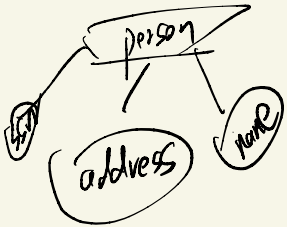
Example:

ER Model

Relational Version

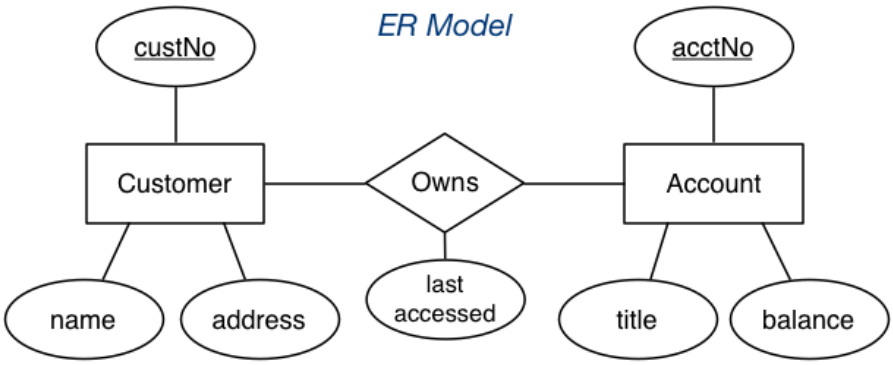
Employee	**SSN**	ename	salary
Contact	**SSN**	**name**	phone


person
address


person
address
name

❖ Mapping N:M Relationships

Example:



key

Relational Version

Customer	**custNo**	name	address

key

Account	**acctNo**	title	balance

Owns	**acctNo**	**custNo**	lastAccessed

acctNo custNo

These are created foreign keys
- acctNo should relate to acctNo
- custNo should relate to custNo

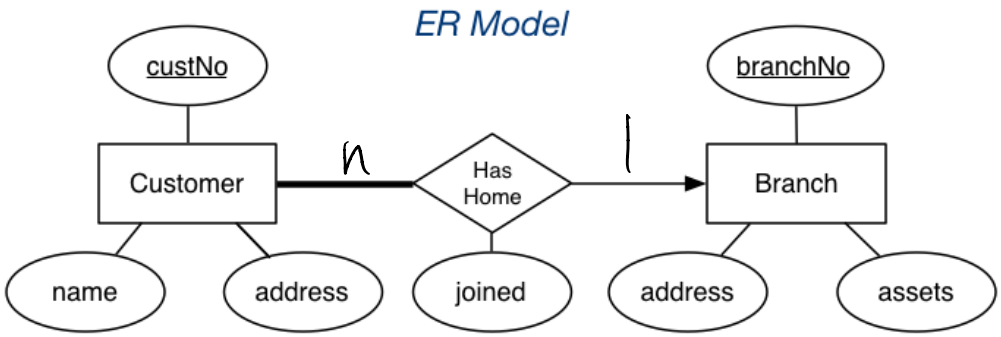
Together, they are composite primary keys

The entity at N side includes the primary key of the entity at 1 side as foreign key

primary key == foreign key

❖ Mapping 1:N Relationships

Example:



Relational Version

fk

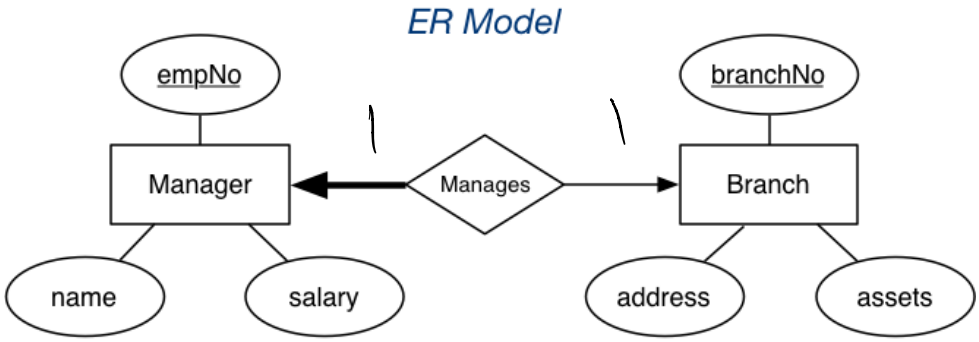
Customer	custNo	name	address	branchNo	joined

Branch	branchNo	address	assets

choose one entity to include the primary key of the other entity as foreign key

❖ Mapping 1:1 Relationships

Example:



Relational Version

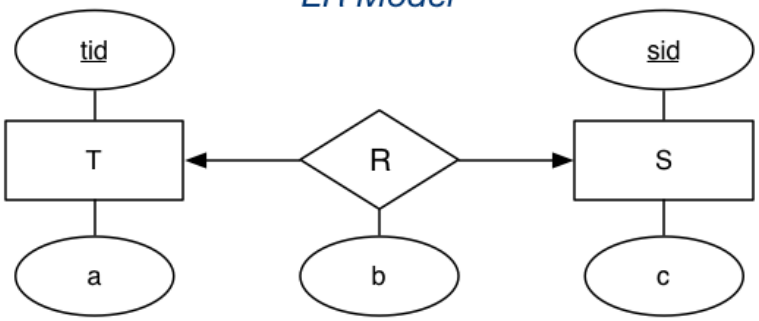
foreign key

Manager	**empNo**	name	salary	branchNo
Branch	**branchNo**	address	assets	

❖ Mapping 1:1 Relationships (cont)

If there is no reason to favour one side of the relationship ...

ER Model



Relational Version #1

T | tid | a | sid | b

S | sid | c

Relational Version #2

T | tid | a

S | sid | c | tid | b

❖ Mapping n-way Relationships

Relationship mappings above assume binary relationship.

If multiple entities are involved:

- *n:m* generalises naturally to *n:m:p:q*
  - include foreign key for each participating entity
  - include any other attributes of the relationship
- other multiplicities (e.g. *1:n:m*) ...
  - need to be mapped the same as *n:m:p:q*
  - so not quite an accurate mapping of the ER

Some people advocate converting n-way relationships into:

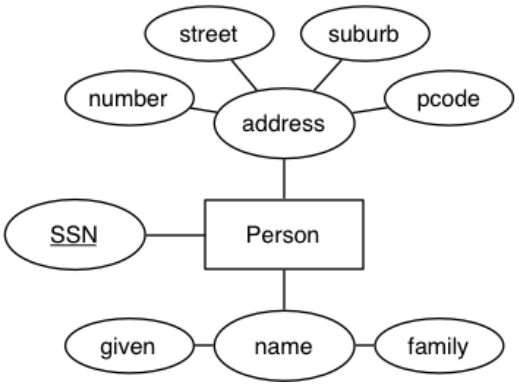
- a new entity, and a set of *n* binary relationships

❖ Mapping Composite Attributes

Composite attributes are mapped by concatenation or flattening.

Example:

ER Model



Relational Version #1

Person

SSN	name	address

Relational Version #2

Person

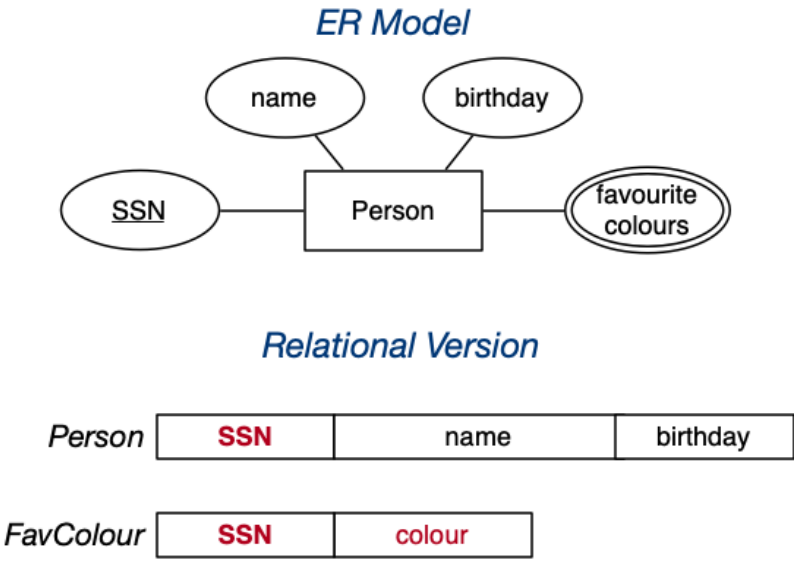
SSN	given	family	.....

.....	number	street	suburb	pcode

❖ Mapping Multi-valued Attributes (MVAs)

MVAs are mapped by a new table linking values to their entity.

Example:



Produced: 21 Feb 2023