>>

COMP3311 Week 2 Monday Lecture

- Week 02
- Email and Context
- n:m:p Relationships
- Recap
- Relational Data Model
- Exercise: ER-to-relational Mapping
- Mapping ER to SQL
- SQL Schemas
- SQL Types
- Exercise: ER-to-SQL Mapping
- SQL Constraints
- Exercise: Constraints
- Exercise: ER-to-SQL Mappings
- Mapping Strong Entities
- Mapping Weak Entities
- Mapping N:M Relationships
- Mapping 1:N Relationships
- Mapping 1:1 Relationships
- Mapping n-way Relationships
- Exercise: Mapping n-way Relationships

∧ >>

❖ Week 02

In today's lecture ...

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

Things to do ...

- Tutorials start this week
- Quiz due Friday midnight
- Set up your PostgreSQL server
  - ~170/690 students have logged in to vxdb2 and have /localstorage
  - come to CSE Help! if you're having trouble installing PostgreSQL

<< ∧ >>

❖ Email and Context

When sending email to us (cs3311), please include your zID

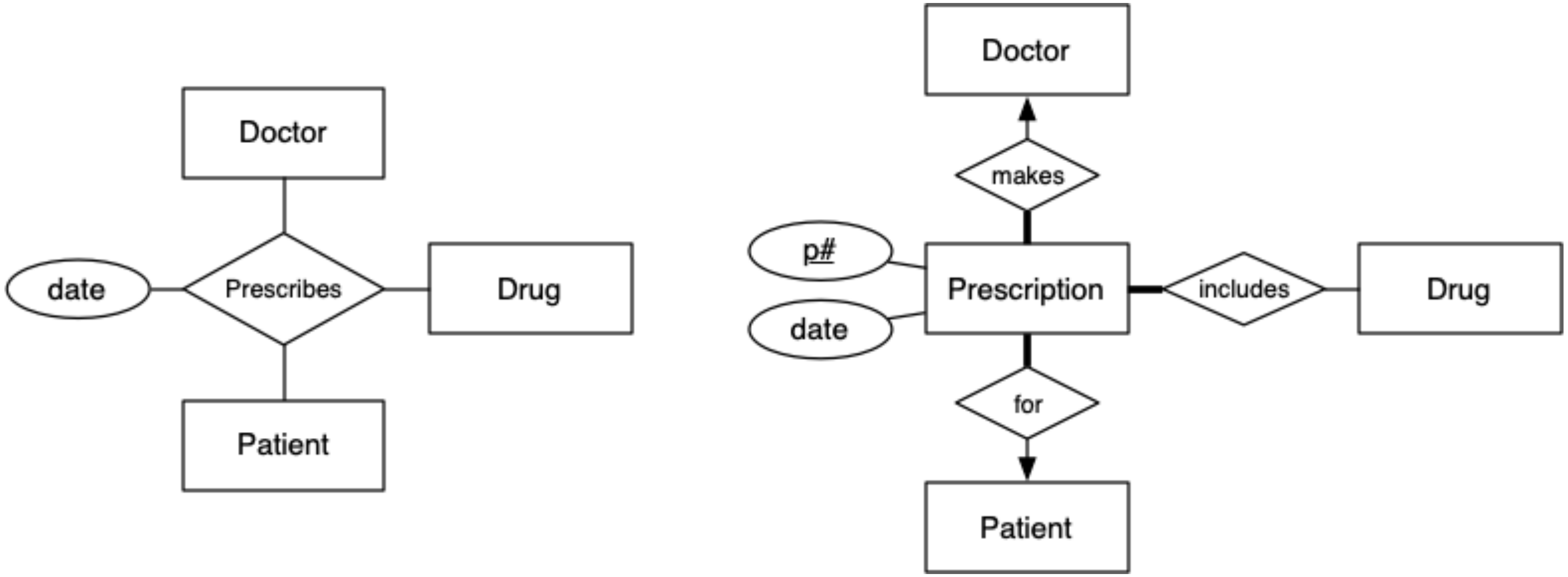
Before typing commands, think about the context:

Prompt	Context
`$`	in Linux shell run Unix commands, e.g. **`ls`**, **`cd`**
`db=#`	in **`psql`** run SQL commands, e.g. **`select`**, **`update`**
`db>`	in **`sqlite3`** run SQL commands, e.g. **`select`**, **`update`**
none	in **`vim`**, you are doomed but try **`:q`** or **`ZZ`**

<< ∧ >>

❖ n:m:p Relationships

Two possible ER models for "prescribes" in the medical scenario



Could be done as a 3-way relationship, or using a new entity

<< ^ >>

❖ Recap

Entity-relationship data model

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

Relational data model

- attributes, tuples, relations
- attribute = name + domain/type
- tuple = list of attributes, based on attribute definitions
- relation = set of tuples, based on a tuple definition

E.g. **`Student(zID:integer, name:string, WAM:float, ...)`**

❖ Relational Data Model

A relational schema consists of

- a collection of relation definitions + constraints

Different kinds of constraints

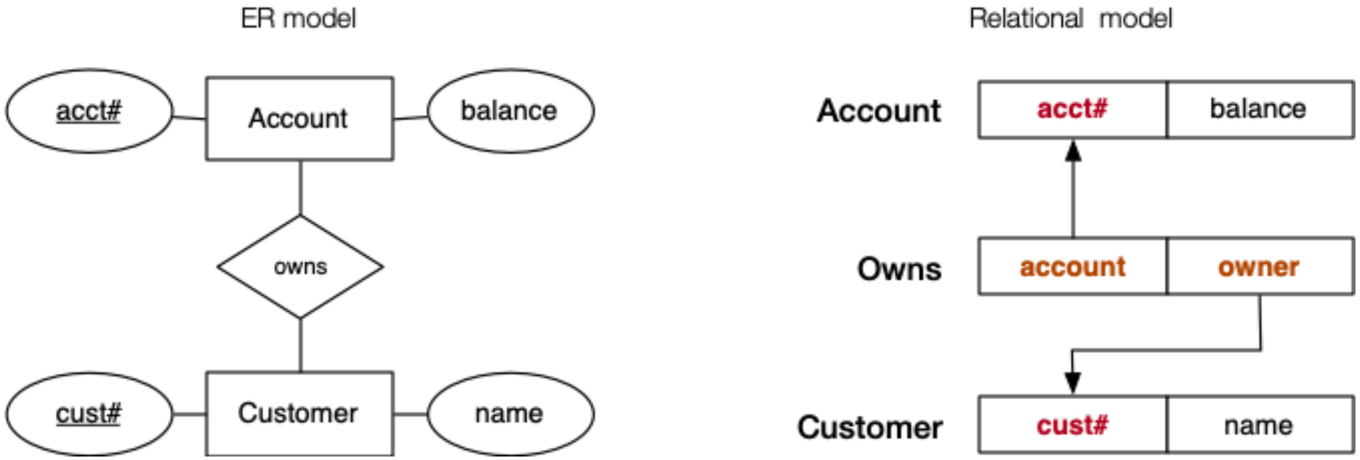
- unique = value of attribute is unique in relation
- key = chosen unique attribute to distinguish tuples
- domain = type of attribute, restrictions within type
- referential integrity = foreign key
    - tuple in relation R has attribute F
    - whose value corresponds to key attribute K in relation S

❖ Relational Data Model (cont)

Mapping an ER model to a relational schema

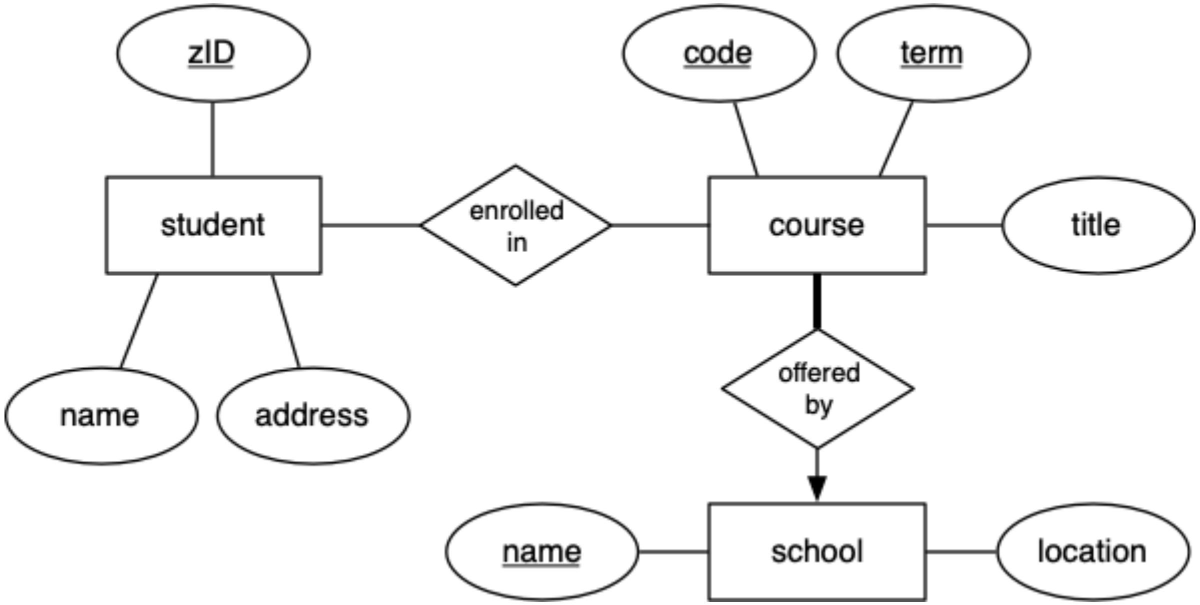
- attributes → attributes, plus domains
- entities → tuples, entity sets → relations
- relationships → relations, plus constraints

Example:



❖ Exercise: ER-to-relational Mapping

Describe this ER model as an (informal) relational schema



<< ∧ >>

❖ Mapping ER to SQL

Above example gives informal description of relational schema

Need a more formal way of describing relational schemas

SQL data definition language (DDL) provides this

SQL is an implementation of relational data model

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

Mapping ER to SQL

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

<< ∧ >>

❖ Mapping ER to SQL (cont)

Useful strategy for database design:

- perform initial data modelling using ER
  (conceptual-level modelling, gives a "map" of the DB)
- 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

Note: cannot map some things (e.g. n:m total participation)

<< ∧ >>

❖ SQL Schemas

Primary SQL DDL construct is table creation:

```
create table TableName (
   attr_1Name  type [constraints],
   attr_2Name  type [constraints],
   attr_3Name  type [constraints],
   ...
   primary key (attr_xName ),
   foreign key (attr_yName)
               references OtherTable (attr_zName )
);
```

SQL schema = collection of table definitions, including constraints

<< ^ >>

❖ SQL Types

Built-in types

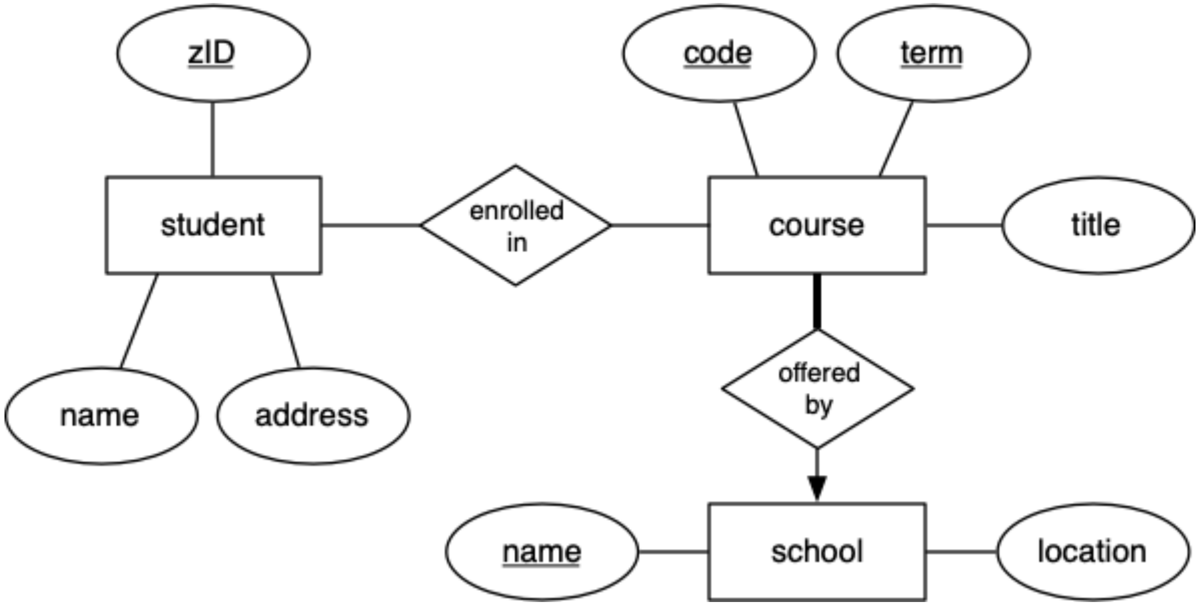
- numeric: **integer**, **numeric(*n*)**, **real**
- strings: **char(*n*)**, **varchar(*n*)**, **text**
- time: **date**, **time**, **timestamp**, **interval**
- **boolean**, monetary, geometric, enumerated, ...

Make your own

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

❖ Exercise: ER-to-SQL Mapping

Describe this ER model as an SQL schema



<< ∧ >>

❖ 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 + - + ')

<< ∧ >>

❖ Exercise: ER-to-SQL Mappings

For each of the following mappings:

- strong entity
- n:m relationship
- 1:n relationship
- 1:1 relationship
- n-way relationships
- multi-valued attributes

give an SQL schema including relevant constraints

<< ∧ >>

❖ Mapping Strong Entities

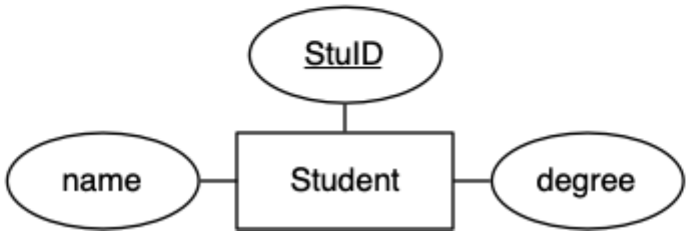
An entity set *E* with atomic attributes $a_1, a_2, \ldots a_n$

maps to

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

Example:

ER Model



Relational Version

Student

StuID	name	degree

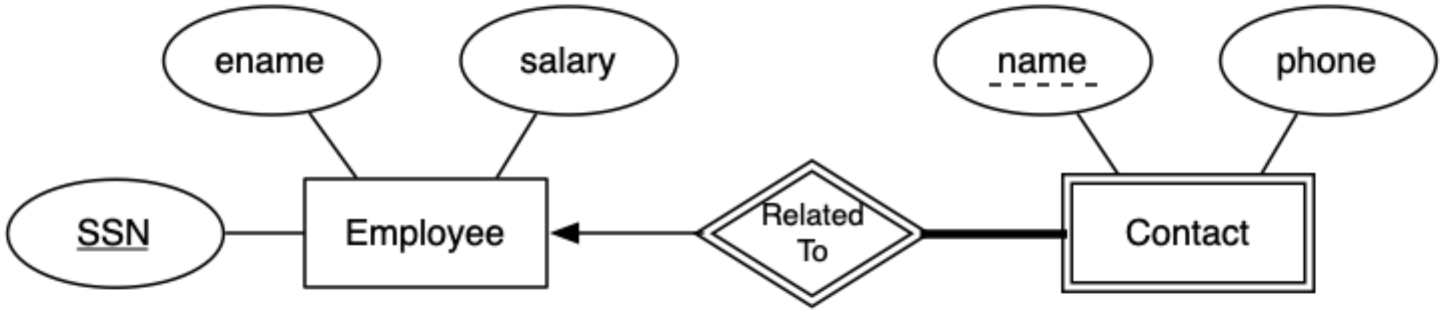
Note: the key is preserved in the mapping.

<< ∧ >>

❖ Mapping Weak Entities

Example:

ER Model



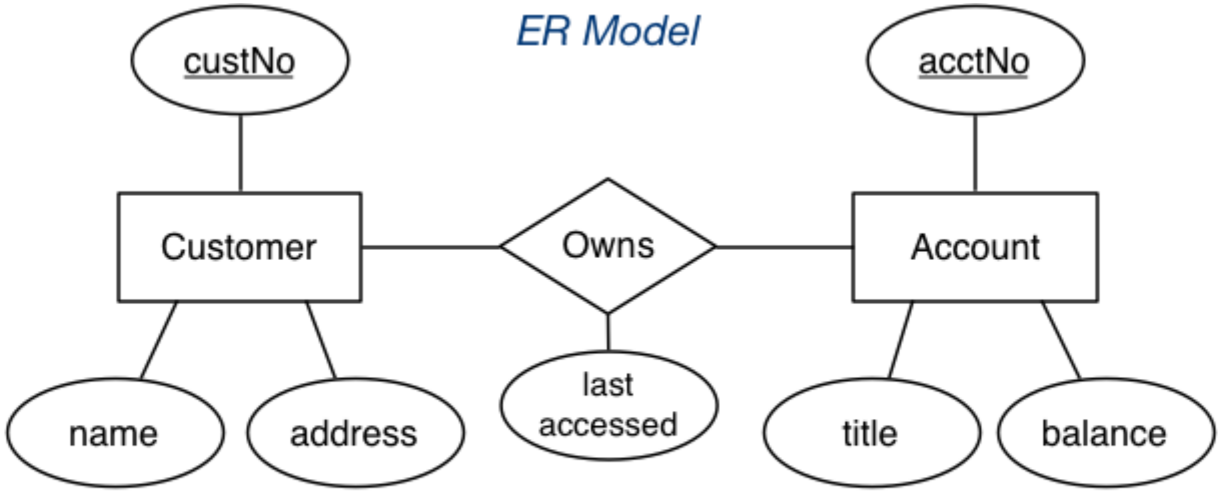
Relational Version

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

❖ Mapping N:M Relationships

Example:

ER Model



Relational Version

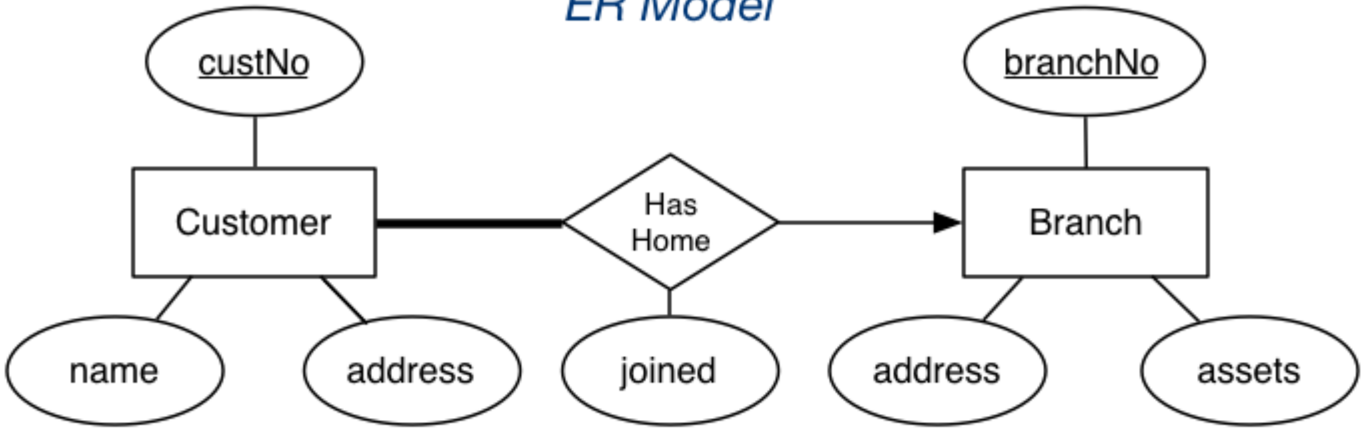
Customer	**custNo**	name	address
Account	**acctNo**	title	balance
Owns	**acctNo**	**custNo**	lastAccessed

<< ∧ >>

❖ Mapping 1:N Relationships

Example:

ER Model



Relational Version

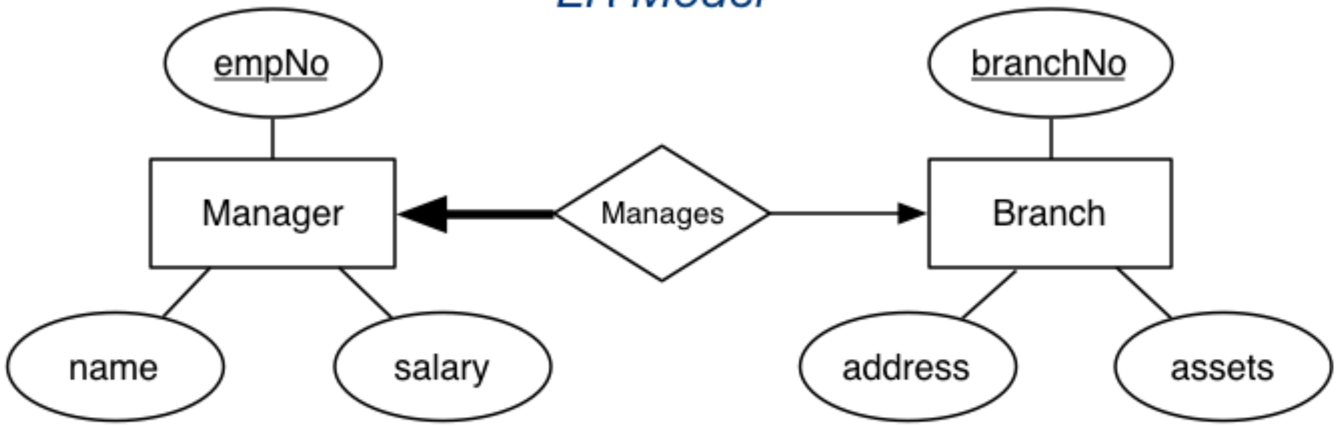
Customer	**custNo**	name	address	branchNo	joined
Branch	**branchNo**	address	assets		

<< ∧ >>

❖ Mapping 1:1 Relationships

Example:

ER Model



Relational Version

Manager	**empNo**	name	salary	branchNo

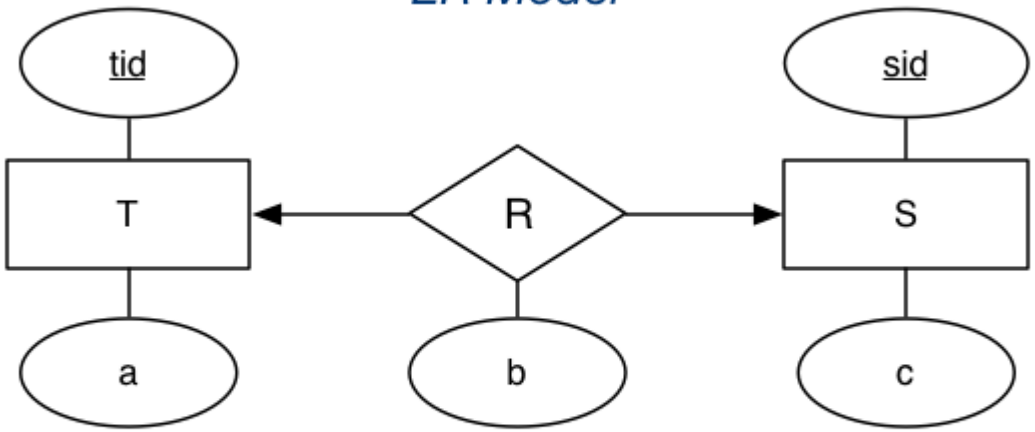
Branch	**branchNo**	address	assets

<< ^ >>

❖ 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

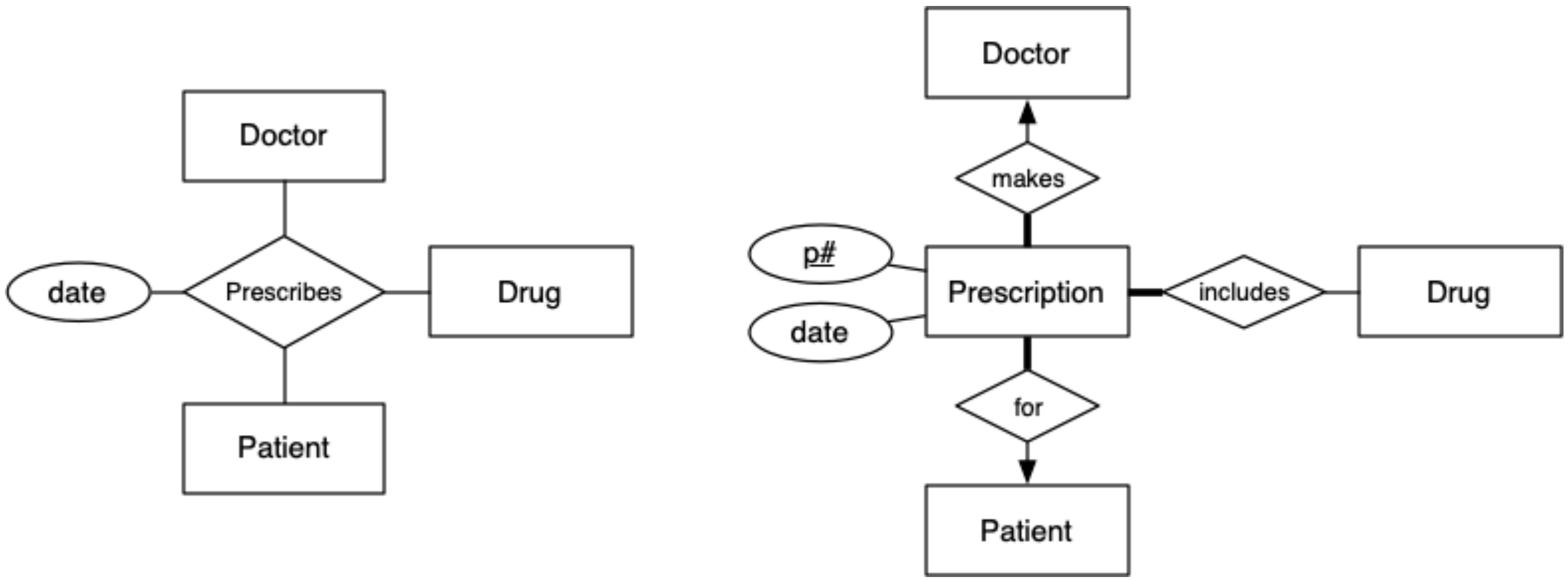
Relationship mappings above assume binary relationship.

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

❖ Exercise: Mapping n-way Relationships

Convert the following ER models into an SQL schema:



Produced: 18 Sep 2023