COMP3311 24T1 Database Systems

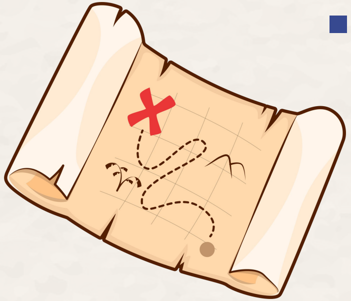
week 1 - 2

Outline

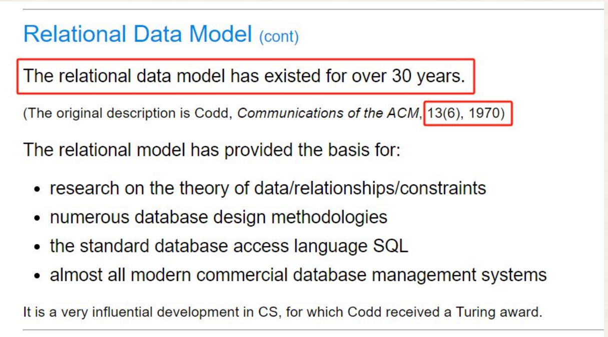
- **Updates**
- ER Modelling
- Relational Modelling

Updates

- New Course Materials Online Now
  - Tutorials
    - https://webcms3.cse.unsw.edu.au/COMP3311/24T1/resources/96321



  - Previous Course Notes
    - much more details; may not be examinable; might be old
    - https://webcms3.cse.unsw.edu.au/COMP3311/24T1/resources/96374



Relational Data Model (cont)

The relational data model has existed for over 30 years.

(The original description is Codd, *Communications of the ACM*, 13(6), 1970)

The relational model has provided the basis for:

- research on the theory of data/relationships/constraints
- numerous database design methodologies
- the standard database access language SQL
- almost all modern commercial database management systems

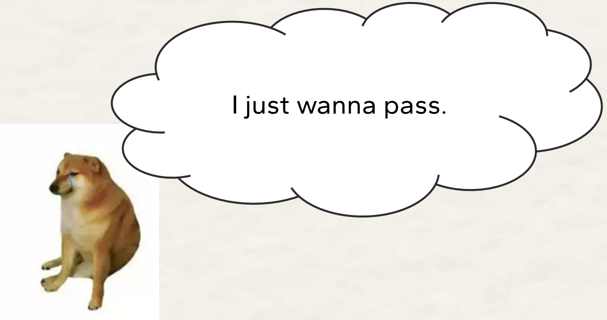
It is a very influential development in CS, for which Codd received a Turing award.

Quick math: how old are the slides?

~20 years old 🙃

Updates

Lecture slides + Tutorials



less info

⚠️ Exam Hurdle: You need to score 40% in the final exam & have a 50% overall score to pass this course.

Previous Course Notes

more info

Textbooks: Course Outline - Course Resources (this is a link)

Updates

Please try to keep quiet during the lecture.

Unless you are asking or answering questions.

What if you really want to talk with your friend? 🤔

A better choice: Build a database:

```
-- create
CREATE TABLE CHAT (
  msg_id INTEGER PRIMARY KEY,
  speaker TEXT NOT NULL,
  msg TEXT NOT NULL,
  time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- insert
INSERT INTO CHAT VALUES (0001, 'Clark', 'haha, the lecture is boring', '2024-02-15 16:20:25-07');
INSERT INTO CHAT VALUES (0002, 'Dave', 'nope, the lecture is interesting', '2024-02-15 16:20:30-07');
-- fetch
SELECT * FROM CHAT WHERE speaker = 'Clark';
```

Try it on: https://onecompiler.com/postgresql/

Outline

- Updates
- **ER Modelling**
- Relational Modelling

Recap

Aims of data modelling:

- describe what **information** is contained in the database
  (e.g., entities: students, courses, accounts, branches, patients, ...)
- describe **relationships** between data items
  (e.g., John is enrolled in COMP3311, Tom's account is held at Coogee)
- describe **constraints** on data
  (e.g., 7-digit IDs, students can enrol in no more than 3 courses per term)

Data modelling is a design process

- converts requirements into a data model
- Input: requirements
- Output: (semi) formal description of the database structure

Entity-Relationship (ER) Modelling

The world is viewed as a collection of inter-related entities.

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)

The ER model is not a standard, so notational variations exist
Lecture notes use notation from SKS and GUW books (simple)

- Database System Concepts , **S**ilberschatz, **K**orth, **S**udarshan, 6th edition, 2010, McGraw-Hill
- Database Systems: The Complete Book , **G**arcia-Molina, **U**llman, **W**idom, 2nd edition, 2008, Prentice-Hall

ER Diagram

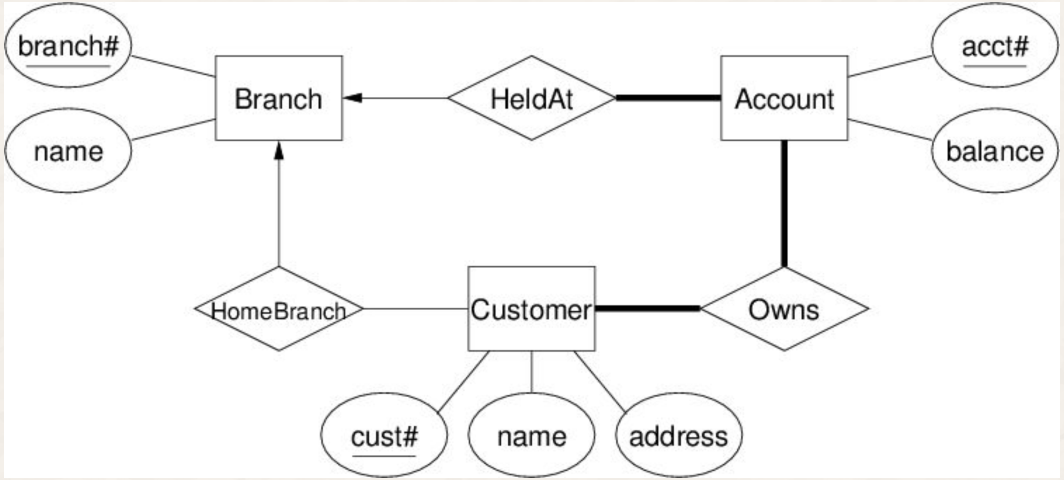
ER diagrams are a graphical tool for data modelling.

An ER diagram consists of:

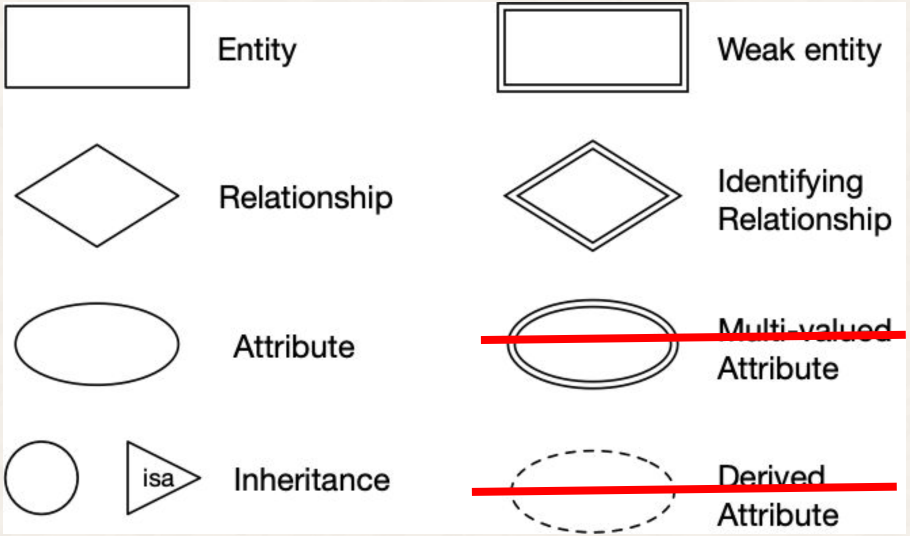
- a collection of **entity set** definitions
- a collection of **relationship set** definitions
- **attributes** associated with entity and relationship sets
- **connections** between entity and relationship sets

Terminology abuse:
we say "entity" when we mean "entity set"
we say "relationship" when we mean "relationship sets"
we say "entity instance" ro refer to a particular entity

Example ER Diagram



ER Diagram (symbols)



Entity Sets - Concepts

An **entity set** can be viewed as either:

- a set of entities with the same set of attributes (extensional)
- an abstract description of a class of entities (intensional)

Key (superkey): any set of attributes whose set of values are **distinct over entity set**

- natural (e.g., name+address+birthday) or artificial (e.g., SSN)

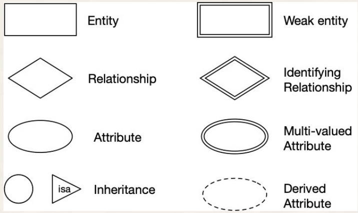
Candidate key = minimal superkey (no subset is a key)

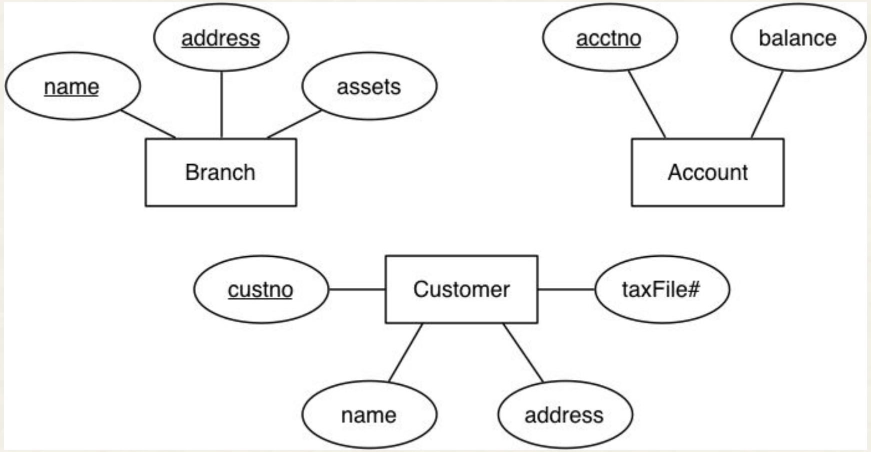
Primary key = candidate key chosen by DB designer

Keys are indicated in ER diagrams by underlining

Primary Keys - Example

ref:



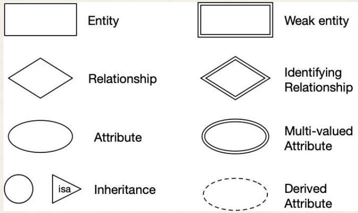


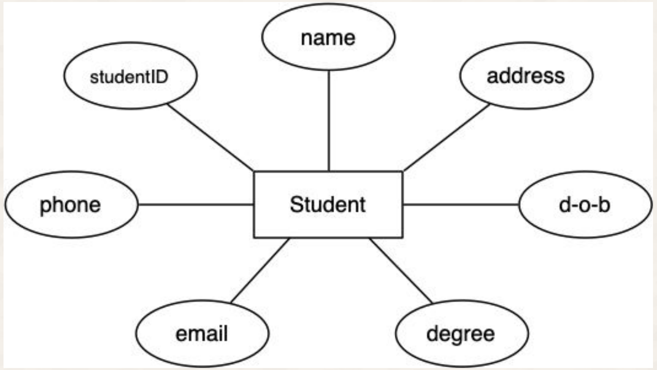
Does anyone have the same name?

E

Exercise: Candidate Key

ref:





Possibilities: {studentID}, {phone}, {email}, {name,address,d-o-b}

Relationship Sets – Concepts

Relationship: an association among several entities

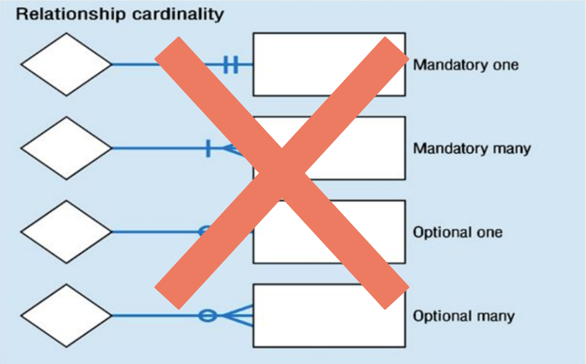
- e.g., Customer(9876) is the owner of Account(12345)

Relationship set: collection of relationships of the same type

Degree = # entities involved in reln (in ER model, ≥ 2)

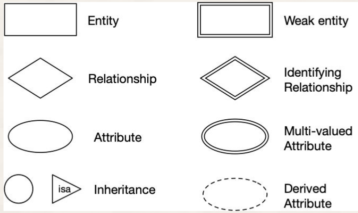
Cardinality = # associated entities on each side of reln

Participation = must every entity be in the relationship

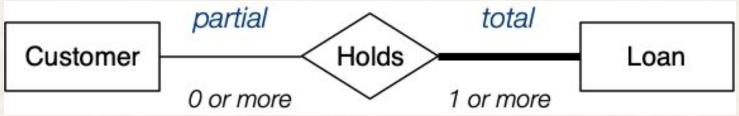


alternative notions

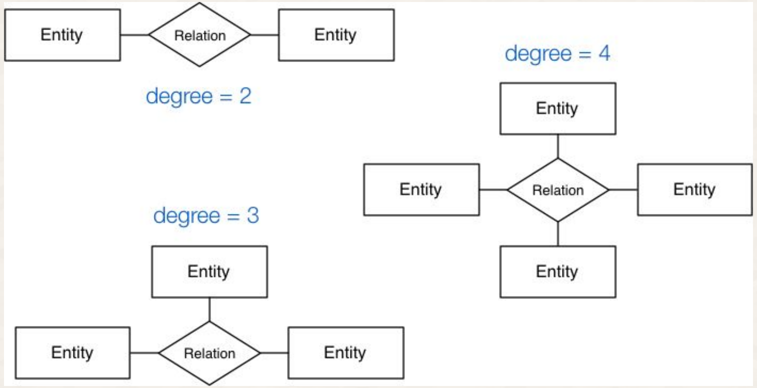
ref:



Example



Relationship - Degree



Relationship – Cardinality

one-to-one

- each a is associated with at most one b
- each b is associated with at most one a

one-to-many

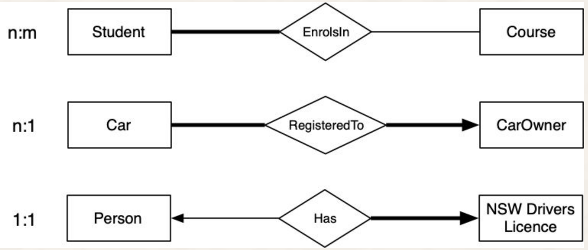
- each a is associated with zero or more b
- each b is associated with at most one a

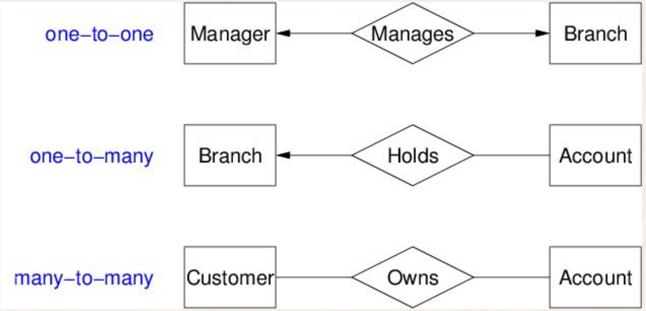
many-to-one

- each a is associated with at most one b
- each b is associated with zero or more a

many-to-many

- each a is associated with zero or more b
- each b is associated with zero or more a

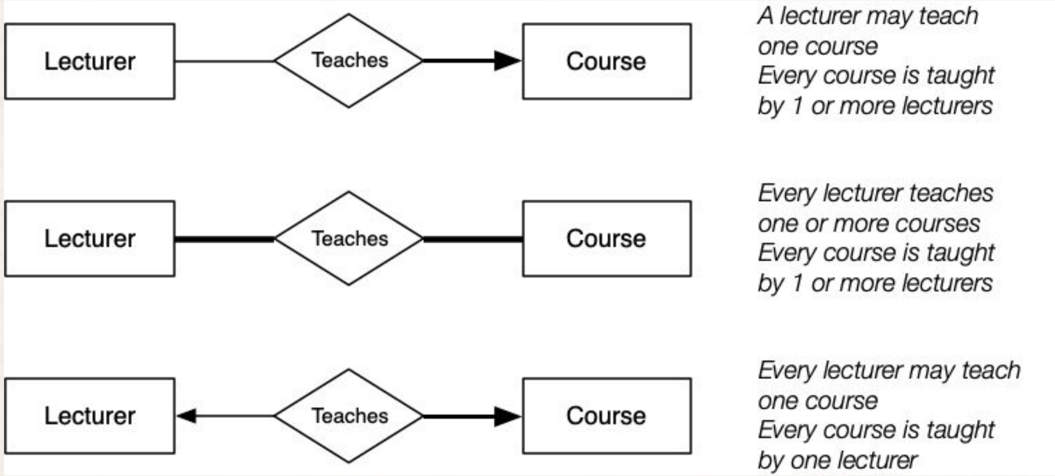




with arrow: one

without arrow: many

Relationship - Participation



A lecturer may teach one course Every course is taught by 1 or more lecturers

Every lecturer teaches one or more courses Every course is taught by 1 or more lecturers

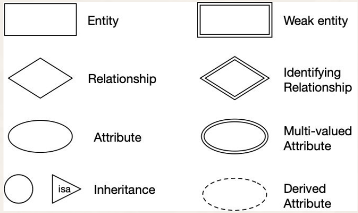
Every lecturer may teach one course Every course is taught by one lecturer

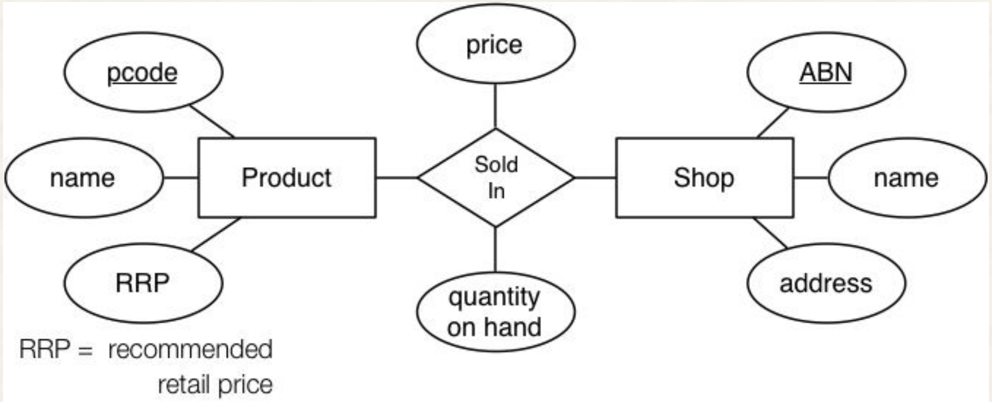
thick: total participation

thin: partial participation (may)

Relationship - Attributes

ref:





RRP = recommended retail price

Weak Entity Set - Concept

Weak entities exist only because of association with other entities.

Examples:

- family of employees in a company
  - (would not be interested in the family once the employee leaves)
- payments on bank loans
  - (if there were no loans, no need to keep information about payments)

Weak entities

- do **not** have a primary key (or any superkey)
- have a subset of attributes that form a **discriminator**
- need to be considered in conjunction with **strong entities**

Weak Entity Set - Concept

- While weak entities do **not** have a primary key

  they have a subset of attributes that form a **discriminator**

- We can can form a primary key by taking a combination of
  - the set of values for the discriminator
  - the primary key of the associated strong entity

Example:

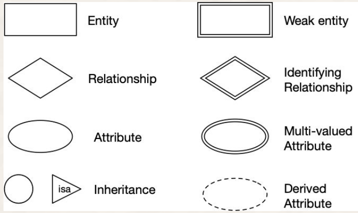
Ian's son called Tim is different to Paul's son called Tim

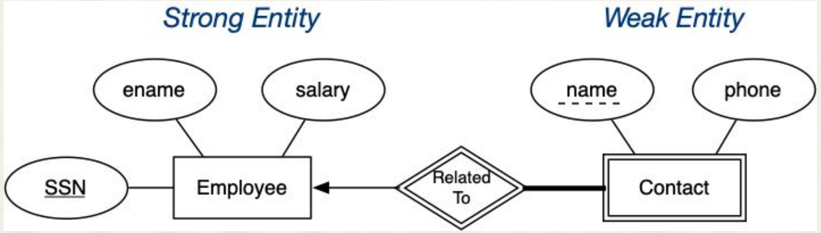
Weak Entity Set – Notion

In ER diagrams:

- weak entities are denoted by double-boxes
- strong/weak entity relationships are denoted by double-diamonds
- discriminators are denoted by dotted underline

ref:





Subclasses and Inheritance

Extensions to the "standard" ER model include inheritance.

A subclass of an entity set A is a set of entities:

- with all attributes of A, plus (usually) it own attributes
- that is involved in all of A's relationships, plus its own

In other words, the subclass inherits the attributes and relationships of A.

Subclasses and Inheritance

If an entity set has multiple subclasses, they may be:

- disjoint - an entity belongs to at most one subclass
- overlapping - an entity may belong to several subclasses

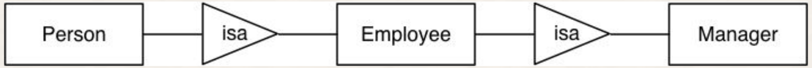
An orthogonal property is the completeness contraint:

- total - all entities must belong to at least one subclass
- partial - some entities may belong to no subclass

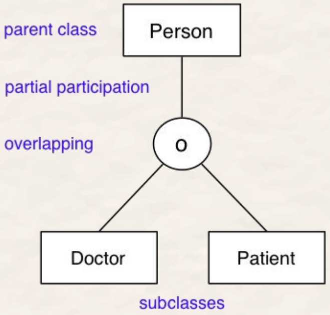
Subclasses and Inheritance

ER diagrams use the following notation:

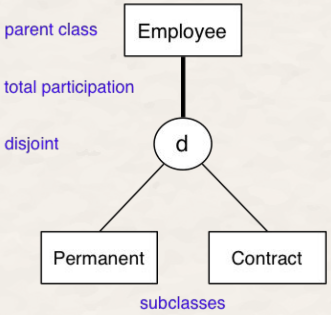
- subclass denoted by **ISA**
  - entity has one subclass ("B is-a A" specialisation)
- disjoint/overlapping = the letter **'d'/'o'** in a circle
- total/partial completeness = **double(thick)/normal** line



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



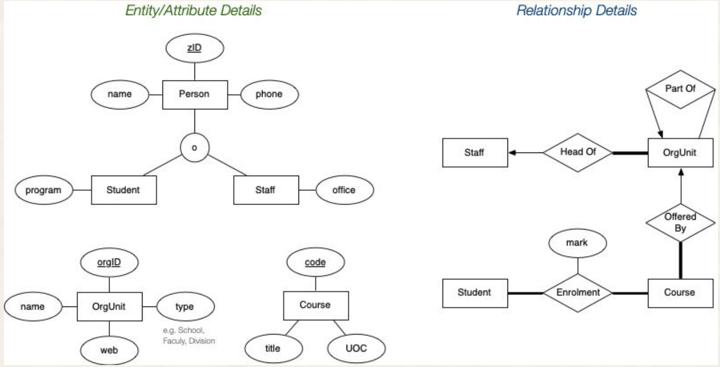
Every employee is either a permanent employee or works under a contract



Handling Large ER Diagrams

One commonly used strategy:

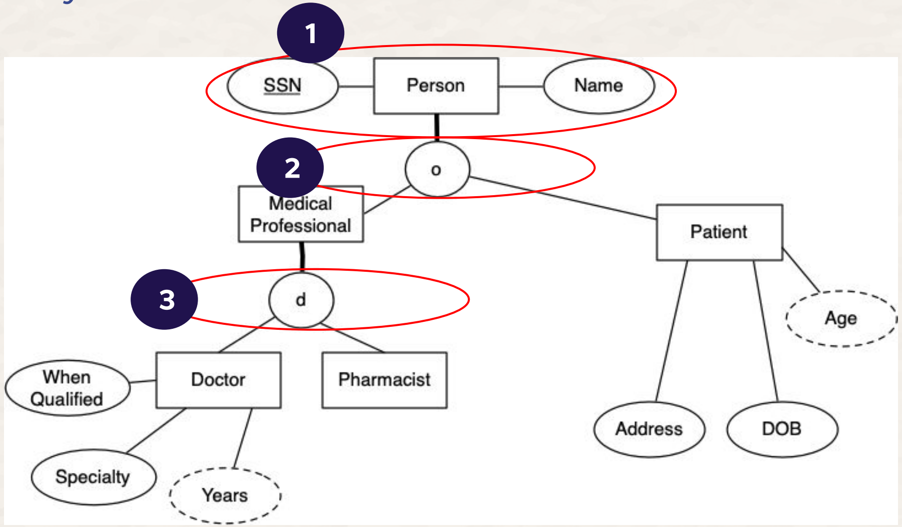
- define entity sets separately, showing attributes
- combine entities and relationships on a single diagram (but without showing entity attributes)
- if very large design, may use several linked diagrams

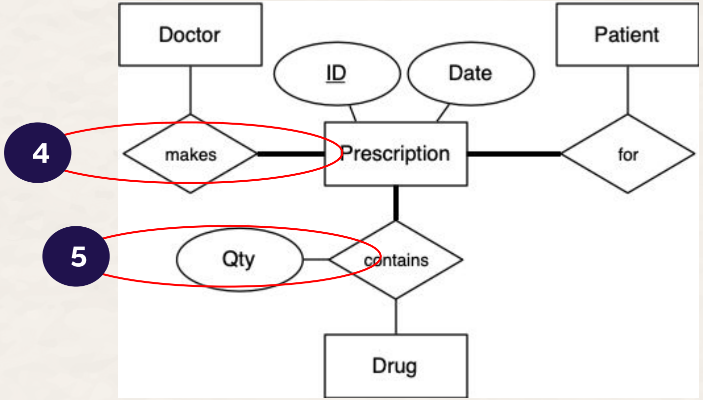


E

Exercise: A medical ER diagram

Try to describe some entities/attributes/relations





Limitations of ER Models

There are some design aspects that ER does not deal with:

- **attribute domains**
  - e.g. should phone "number" be represented by number or string?
- **computational dependencies**
  - e.g. employee's salary is determined by department and level
- **general constraints**
  - e.g. manager's budget is less than 10% of the combined budget of all departments they manage

Some of these are handled later in the relational model.

Data Models

- **Entity-relationship (ER) model**
    - world is modelled via entities, relationships, attributes

- **Relational model**
    - world is modelled via tuples, relations, constraints

- **SQL schemas**
    - a good approximation of the relational model

Also ODL, UML, and a variety of others ... but not in this course.

Outline

- Updates
- ER Modelling
- **Relational Modelling**

Relational Model

The relational data model describes the world as:

- a collection of inter-related relations (or tables)

Goal of relational model:

- a simple, general data modelling formalism
  which maps easily to file structures (i.e. implementable)

Can be viewed as an attempt to formalise the file organisations that were in common use at the time the model was developed.

Relational Model

The relational data model has existed for **over 50 years**.

(The original description is **Codd**, Communications of the ACM, 13(6), 1970)

The relational model has provided the basis for:

- research on the theory of data/relationships/constraints
- numerous database design methodologies
- the standard database access language SQL
- almost all modern commercial database management systems

It is a very influential development in CS, for which Codd received **a Turing award**.

Relational Model

The relational data model describes the world as

- a collection of **inter-connected relations (or tables)**

The relational model has one structuring mechanism: **relations**

- relations are used to model both entities and relationships

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 a1,a2,...) has:

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

Relational Model – Terminologies

Consider relation R with attributes $a_1, a_2, \ldots a_n$
Relation schema of R : $R(a_1{:}D_1, a_2{:}D_2, \ldots a_n{:}D_n)$
Tuple of R : an element of $D_1 \times D_2 \times \ldots \times D_n$ (i.e. list of values)
Instance of R : subset of $D_1 \times D_2 \times \ldots \times D_n$ (i.e. set of tuples)

- Note: tuples: $(2,3) \neq (3,2)$ relation: $\{ (a,b), (c,d) \} = \{ (c,d), (a,b) \}$

- Domains are comprised of **atomic** values (e.g. integer, string, date) (no composite or multi-valued attributes)
- A distinguished value **NULL** belongs to all domains
- Each relation has a **key** (subset of attributes unique for each tuple)
- A **database** is a collection of associated relations.

Relational Model - Example

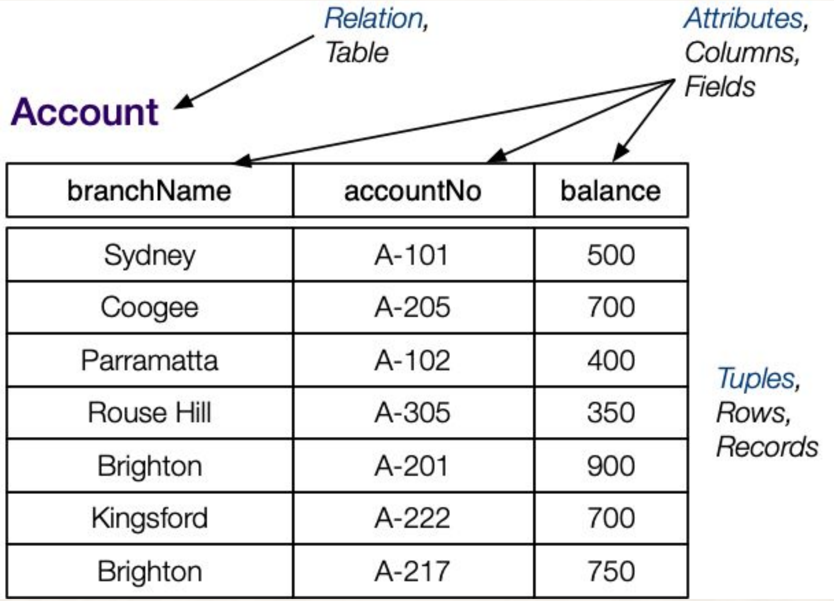
A relation: **Account(branchName, accountNo, balance)**

And 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)
}

Note: **accountNo** is a primary key.

Relational Model – Example



Relation, Table

Attributes, Columns, Fields

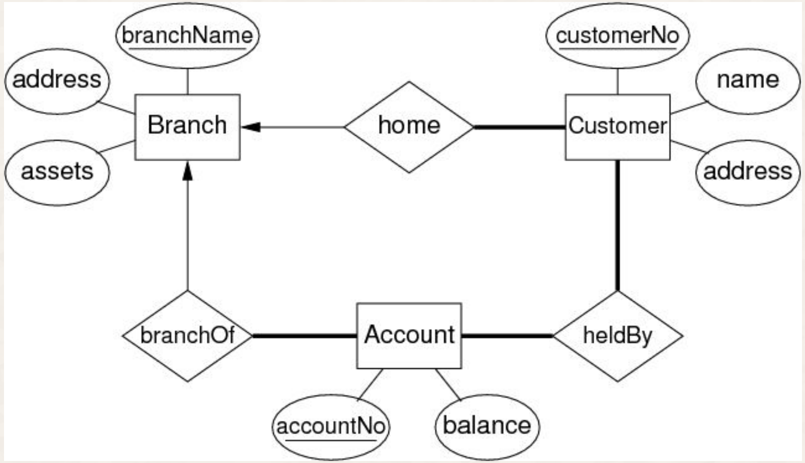
Tuples, Rows, Records

Account

branchName	accountNo	balance
Sydney	A-101	500
Coogee	A-205	700
Parramatta	A-102	400
Rouse Hill	A-305	350
Brighton	A-201	900
Kingsford	A-222	700
Brighton	A-217	750

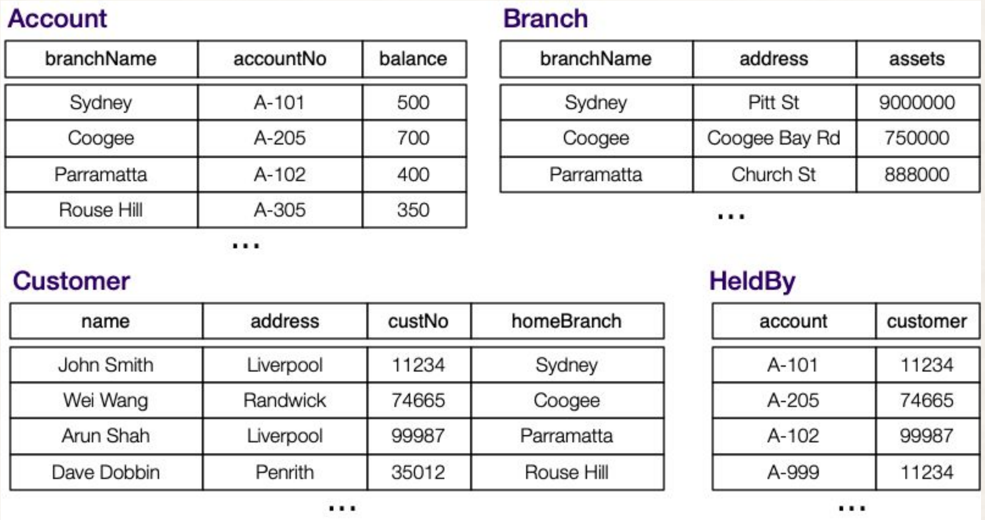
Relational DB - Example (ER)

Consider this ER data model for a bank:



Relational DB – Example (instance)

Instances of the relations:



Account

branchName	accountNo	balance
Sydney	A-101	500
Coogee	A-205	700
Parramatta	A-102	400
Rouse Hill	A-305	350

...

Branch

branchName	address	assets
Sydney	Pitt St	9000000
Coogee	Coogee Bay Rd	750000
Parramatta	Church St	888000

...

Customer

name	address	custNo	homeBranch
John Smith	Liverpool	11234	Sydney
Wei Wang	Randwick	74665	Coogee
Arun Shah	Liverpool	99987	Parramatta
Dave Dobbin	Penrith	35012	Rouse Hill

...

HeldBy

account	customer
A-101	11234
A-205	74665
A-102	99987
A-999	11234

...

Relational Model – Constraints

To represent real-world problems, need to describe

- what **values** are/are not allowed
- what **combinations of values** are/are not allowed

Constraints are logical statements that do this:

- **domain constraints**: limit the set of values that attributes can take
- **key constraints**: identify attributes that uniquely identify tuples
- **entity integrity constraints**: require keys to be fully-defined
- **referential integrity constraints**: require references to other tables to be valid

Relational Model – Constraints

Domain constraints example:

- **Employee.age** attribute is typically defined as **integer**
- better modelled by adding extra constraint (15<age<66)
- Note: NULL satisfies all domain constraints (except (**NOT NULL**))

Key constraints example:

- Student(id, ...) is guaranteed unique
- Class(...,day,time,location,...) is unique

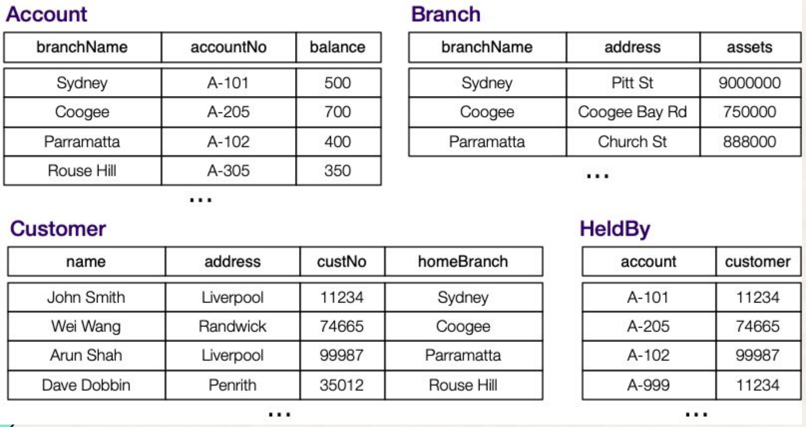
Entity integrity example:

- Class(...,Mon,2pm,Lyre,...) is well-defined
- Class(...,NULL,2pm,Lyre,...) is not well-defined

Relational Model – Constraints

Referential integrity constraints

- describe references between relations (tables)
- are related to notion of a **foreign key** (FK)



Account

branchName	accountNo	balance
Sydney	A-101	500
Coogee	A-205	700
Parramatta	A-102	400
Rouse Hill	A-305	350

…

Branch

branchName	address	assets
Sydney	Pitt St	9000000
Coogee	Coogee Bay Rd	750000
Parramatta	Church St	888000

…

Customer

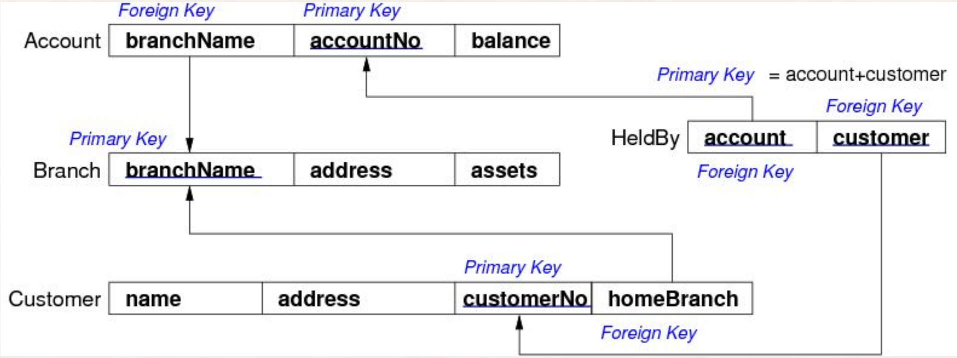
name	address	custNo	homeBranch
John Smith	Liverpool	11234	Sydney
Wei Wang	Randwick	74665	Coogee
Arun Shah	Liverpool	99987	Parramatta
Dave Dobbin	Penrith	35012	Rouse Hill

…

HeldBy

account	customer
A-101	11234
A-205	74665
A-102	99987
A-999	11234

…



Relational Model - Constraints

A set of attributes F in relation R1 is a foreign key for R2 if:

- the attributes in F correspond to the primary key of R2
- the value for F in each tuple of R1
  - either occurs as a primary key in R2
  - or is entirely NULL

Foreign keys are critical in relational DBs; they provide ...

- the "glue" that links individual relations (tables)
- the way to assemble query answers from multiple tables
- the relational representation of ER relationships

Relational Model – Concepts (recap)

Attribute = data item with a name and a type/domain

e.g. account_balance has domain non-negative integer

Tuple = list of values (cf. Python tuples, C structs)

e.g. (1234567, John Smith, BE, SENG, 75.2)

Relation Instance = set of tuples

e.g. { (1,2,3), (3,2,1), (1,3,5), (2,4,6) }

Constraint = logical statements on valid data

e.g. zID is unique and 0 ≤ WAM ≤ 100

ER vs Relational Models

Correspondence between ER and Relational models:

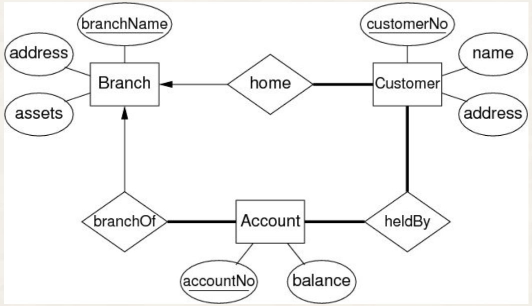
Relational attributes correspond to **ER attributes**

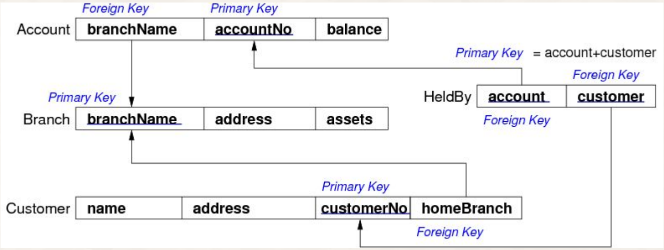
- although ER attributes generally don't have explicit domains

Relational tuples correspond to **ER entities**

Relations correspond to **sets of ER entities**

Relations also correspond **ER relationships**





Thank you!