THE UNIVERSITY
of ADELAIDE

School of Computer Science

COMP SCI 2207/7207 Web and Database Computing
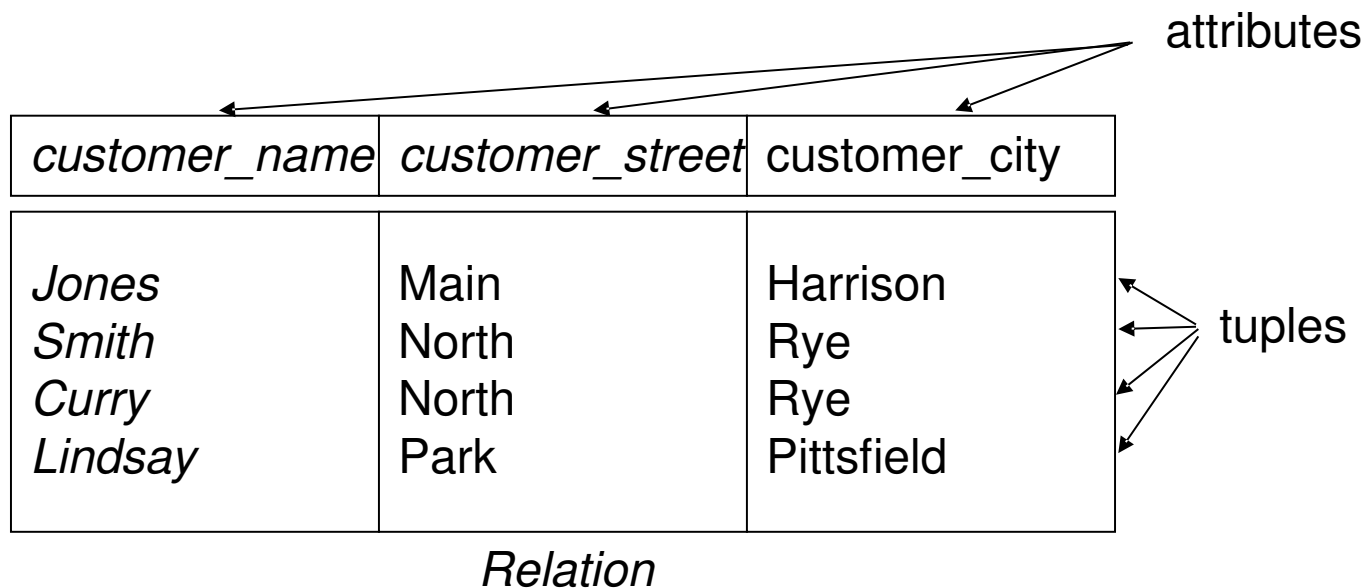## Lecture 27 – E-R Model to Relation Schemas

adelaide.edu.au

seek LIGHT

Design Database

- Step 1 – Specify requirements
- Step 2 – Model data using Entity-Relationship (ER) model
- **Step 3 – Derive ER model to a relation schema**

Relational Databases

- A relational database stores information in 2 dimensional tables
  - Relation, i.e., a table
  - Attributes, i.e., column headers
  - Tuples, i.e., rows in a table

attributes

customer_name	*customer_street*	customer_city
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

tuples

Relation

Relation Schema

- Before create the database and its tables, we should have **a relation schema** for each table.

- A relation schema is a named relation defined by a tuple of attributes

  RELATION = (Attribute$_1$, Attribute$_2$, …….., Attribute$_N$)

- Properties of a relation schema:
  - A relation has a name that is distinct from all other relation names in the relational database schema
  - The name of the relation is written with capital letters.
  - Each attribute in one relation has a distinct name
  - The order of the attributes has no significance, however, the primary key is often listed as the first attribute

Design Database

- Step 1 – Specify requirements
- Step 2 – Model data using Entity-Relationship (ER) model
- **Step 3 – Derive ER model to a relation schema**
  - Step 3a: Mapping strong entities & attributes
  - Step 3b: Mapping weak entities
  - Step 3c: Mapping relationships

Design Database

- Step 1 – Specify requirements

- Step 2 – Model data using Entity-Relationship (ER) model

- **Step 3 – Derive ER model to a relation schema**
  - **Step 3a: Mapping strong entities & attributes**
  - Step 3b: Mapping weak entities
  - Step 3c: Mapping relationships

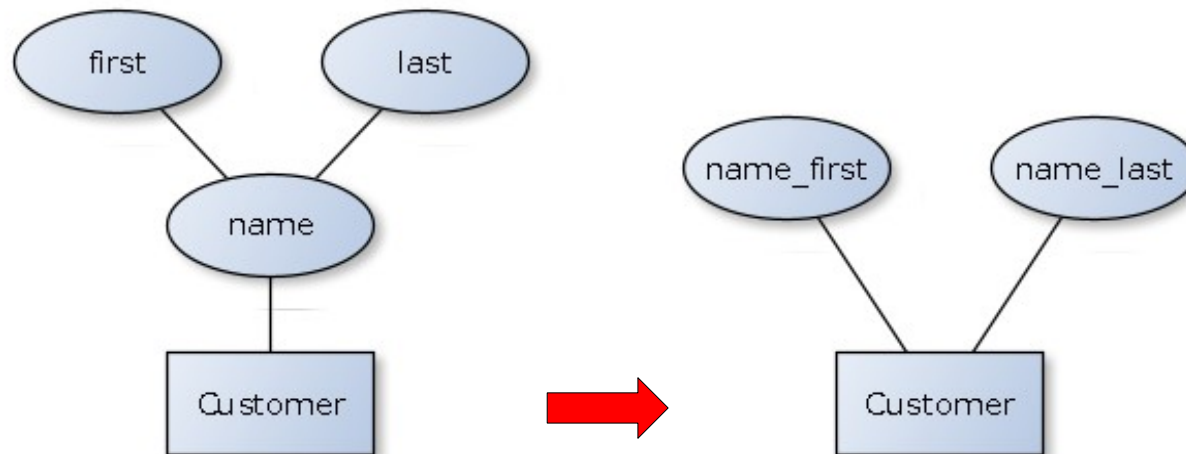# Mapping Strong Entities & Attributes

- Entity set is derived to a relation
- Entity attributes are derived to attributes of relation
- The primary key is underlined
- Example: Customer entity set

  CUSTOMER=(cust_id, cust_name, cust_street, cust_suburb)
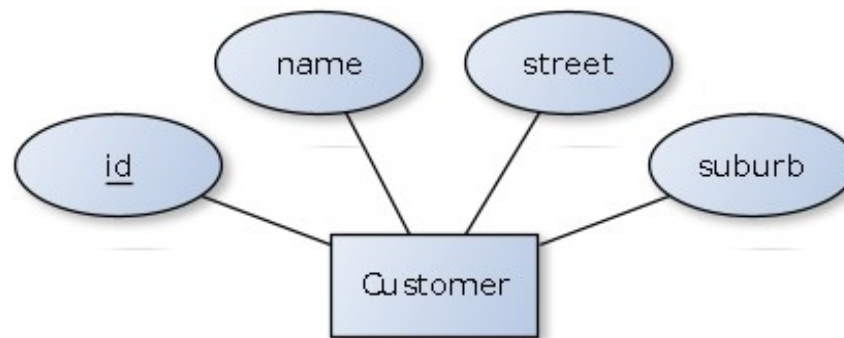
Mapping Strong Entities & Attributes

- **Composite attributes**
  - Flatten composite attributes by creating an attribute for each component attribute
- Example: CUSTOMER=(cust_name_first, cust_name_last)
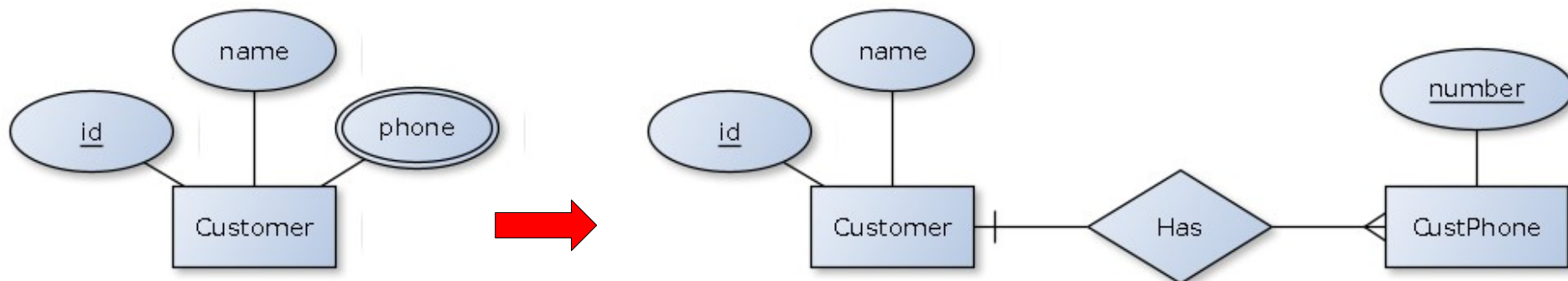
Mapping Strong Entities & Attributes

- **Multi-value attributes**
  - Create a new relation where attributes are primary key of the entity set and the multi-value attribute
  - Each value of the multi-value attribute maps to a separate tuple of the relation
- Example:
  CUSTOMER=(cust_id, name)
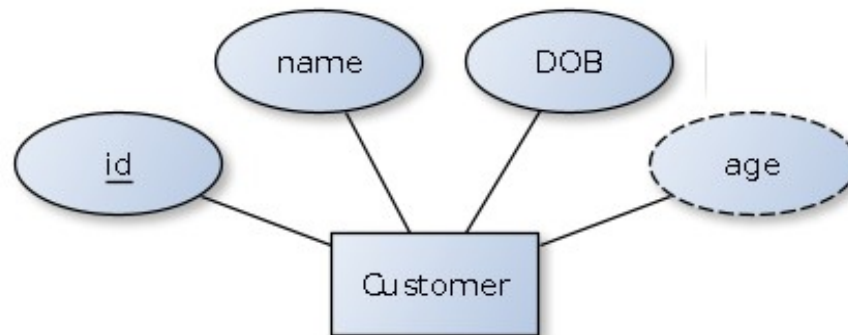 CUST_PHONE = (cust_id, phone_number)

Mapping Strong Entities & Attributes

- **Derived attributes**
  - The value of derived attributes are not store in DB since it can be computed from another attributes
  - Derived attributes are not included in a relation schema
- Example:
  CUSTOMER=(cust_id, cust_name, cust_DOB)
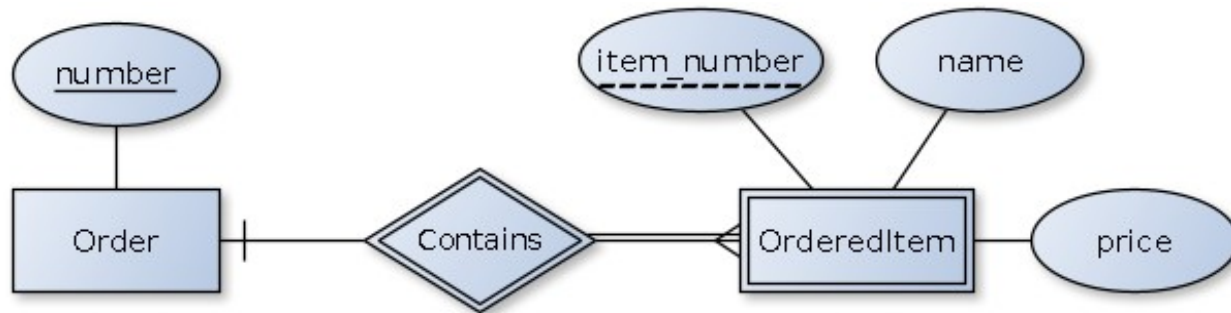
Design Database

- Step 1 – Specify requirements

- Step 2 – Model data using Entity-Relationship (ER) model

- **Step 3 – Derive ER model to a relation schema**
  - Step 3a: Mapping strong entities & attributes
  - **Step 3b: Mapping weak entities**
  - Step 3c: Mapping relationships

Mapping Weak Entities

- An existence of an entity in a weak entity set is dependent to another strong entity set
- **Partial key (Discriminator)** - Attributes in a weak entity set that distinguish weak entities related to the same strong entity
  - ER Diagram: Dashed underline an attribute
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's partial key.

Mapping Weak Entities

- Example:
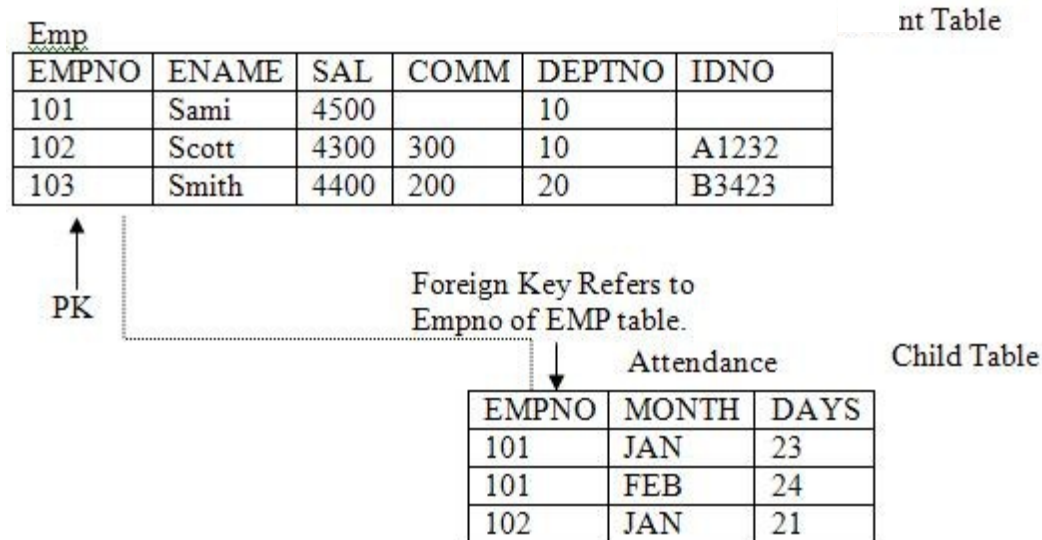 ORDERED_ITEM=(<u>order_number, item_number</u>, name, price)

Design Database

- Step 1 – Specify requirements
- Step 2 – Model data using Entity-Relationship (ER) model
- **Step 3 – Derive ER model to a relation schema**
  - Step 3a: Mapping strong entities & attributes
  - Step 3b: Mapping weak entities
 - **Step 3c: Mapping relationships**

Foreign key

- In rational database, foreign keys are used to establish a relationship between two tables
- Foreign key is a set of attributes that refer to the a primary key of another relation that the relation has a relationship with
- In relation schema, dashed underline indicates a foreign key

Emp

EMPNO	ENAME	SAL	COMM	DEPTNO	IDNO
101	Sami	4500		10	
102	Scott	4300	300	10	A1232
103	Smith	4400	200	20	B3423

nt Table

PK

Foreign Key Refers to
Empno of EMP table.

Attendance   Child Table

| EMPNO	MONTH	DAYS
101	JAN	23
101	FEB	24
102	JAN	21

Mapping One to One Relationship

- Add a foreign key to one of the two relations
- Example:
  CUSTOMER=(cust_id, cust_name)
 ADDRESS=(cust_id, adr_number, adr_street, adr_country)
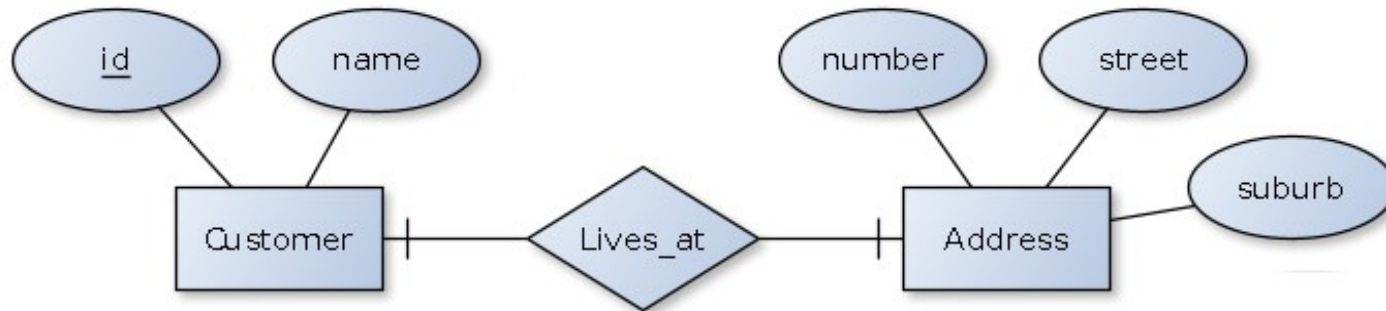
Mapping One to Many Relationship

- Add a foreign key (referring to the 'one' relation) into the 'many' relation, plus descriptive attributes (attributes of the relationship)

- Example:
CUSTOMER=(<u>cust_id</u>, cust_name)
ACCOUNT=(<u>cust_id</u>, <u>acc_number</u>, acc_name, acc_open_date)
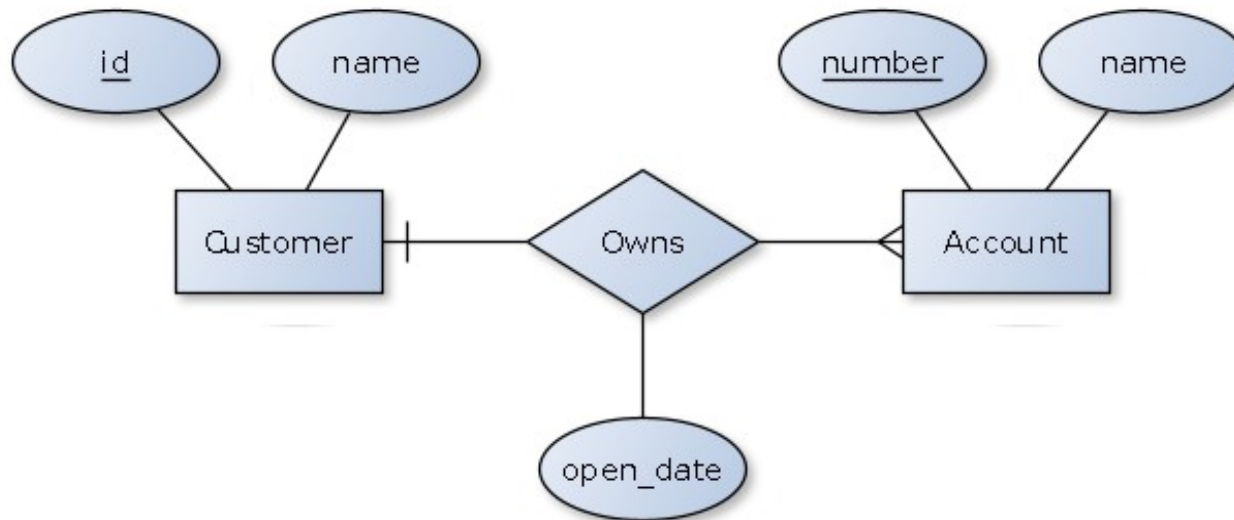
Mapping Many to Many Relationship

- Create a relation that contains foreign keys referring to the primary keys of the participating relationship, plus descriptive attributes

- Example:
  STUDENT=(<u>std_id</u>, std_name)
  COURSE=(<u>course_code</u>, course_name, course_description)
  ENROL=(<u>std_id,course_code</u>, enrol_date)
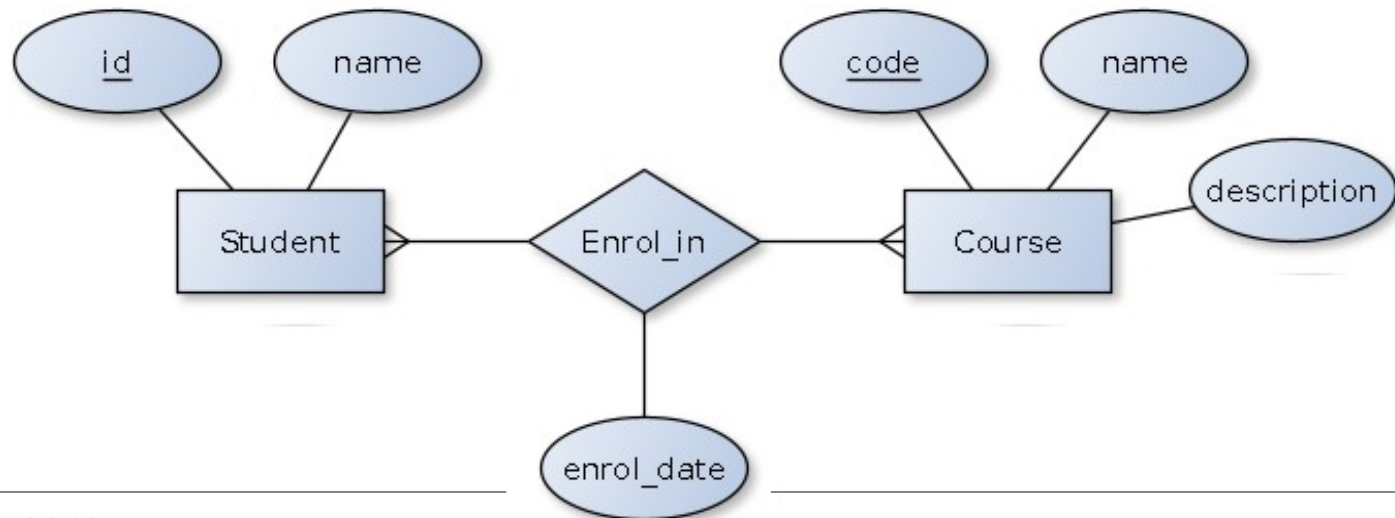
ER Modelling Exercise: Phone Company

- A customer has a name (first and last), customer number (unique) and address.

- Each customer hires one or more phone lines.

- Each line has a unique phone number (the start date for the hire should be recorded).

- A phone line is of a certain line type. There are two line types: the primary line has a higher monthly hire than the additional line type.

- Local calls are free and are included in the hire fee.

- History of domestic and international calls should be stored in DB.

- For each call made from a certain phone line, the database should store its timing (the date and start time), end time and the length of the call.

- A call is classified as international call or domestic.

- The cost per minute for international & domestic calls should be stored in DB.

- The cost per minute for domestic calls depends on the area code.

- The cost for international calls depends on the country code.

- The total cost of a call should also be stored, and is the product of the length of the call and the price per minute for the corresponding type of call.

ER Modelling Exercise: Phone Company
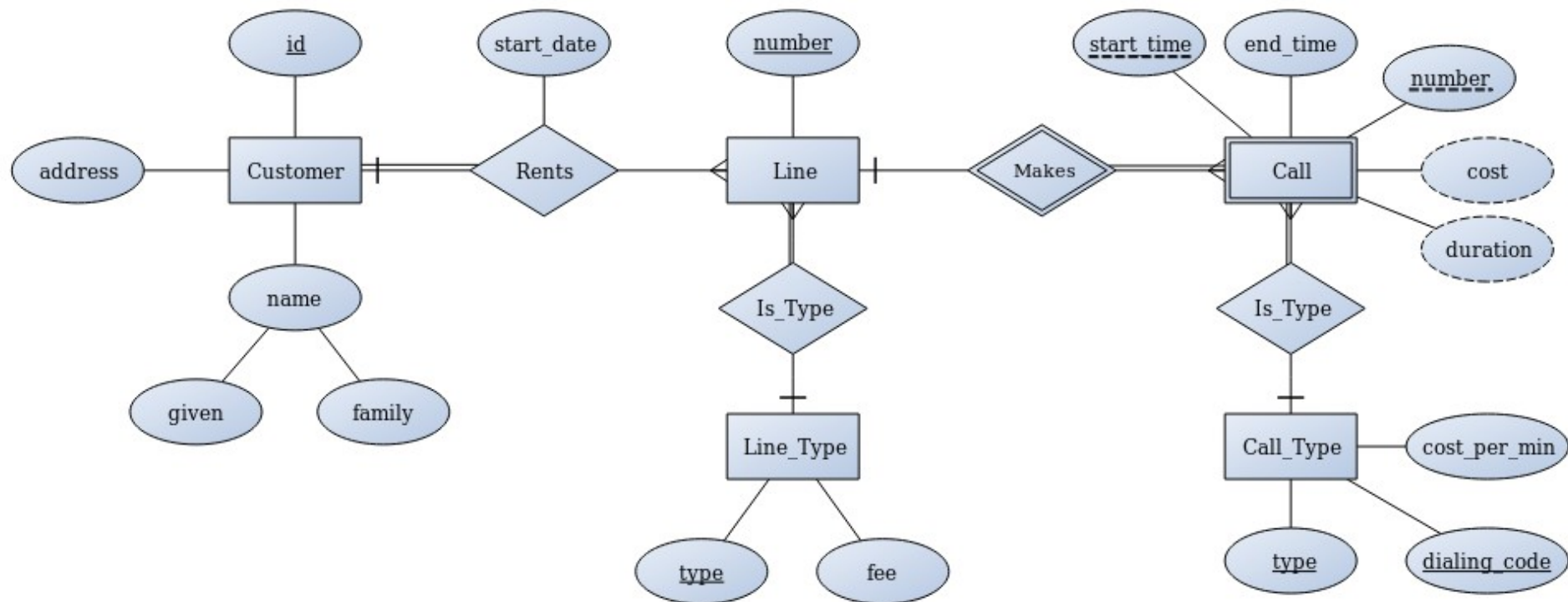
ER Modelling Exercise: Phone Company

- Mapping strong entities & attributes
  - CUSTOMER=(<u>cust_id</u>, name_given, name_family, address)
  - LINE = (<u>line_number</u>)
  - CALL_TYPE = (<u>call_type</u>, dialling_code, cost_per_min)
  - LINE_TYPE = (<u>line_type</u>, fee)
- Mapping weak entities
  - CALL = (<u>line_number, number, start_time, end_time</u>)

ER Modelling Exercise: Phone Company

- Mapping relationships
  - CUSTOMER=(<u>cust_id</u>, name_given, name_family, address)
  - LINE = (<u>number,</u> <u>cust_id</u>, start_date, <u>line_type</u>)
  - CALL_TYPE = (<u>call_type</u>, dialling_code, cost_per_min)
  - LINE_TYPE = (<u>line_type</u>, fee)
  - CALL = (<u>line_number, number, start_time, end_time</u>, <u>call_type</u>)

What's Happening?

- **Milestone 1 DUE tonight**
- Prac Exercise 8 Available, Due Monday Week
  - Websub to come
- Q&A Livestream Wednesday 8:30pm
- Keep working on group projects