MONASH University

MONASH
INFORMATION
TECHNOLOGY

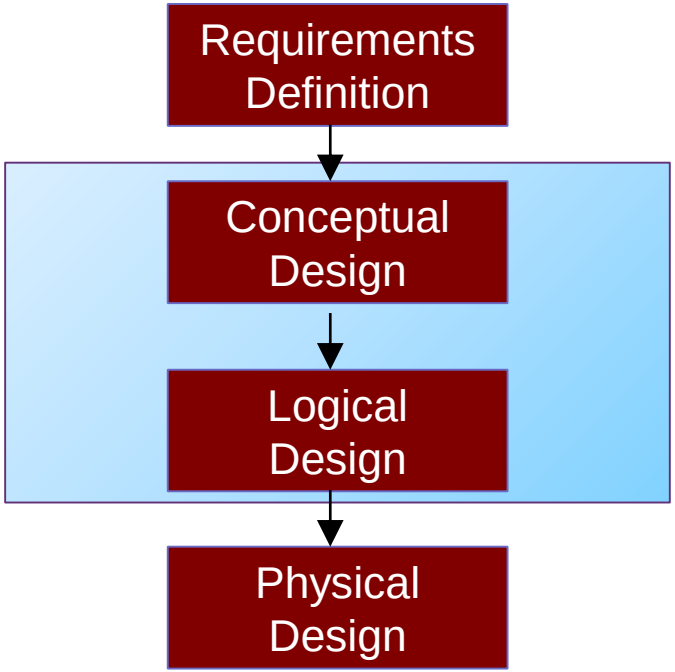
Week 2 – Conceptual Modelling

Please obtain a copy of the case study for this Forum from the week 2 block on Moodle under the "Forum Resources" header

FIT3171 Databases
Semester 1 2022

Malaysia Campus

The Database Design Life Cycle

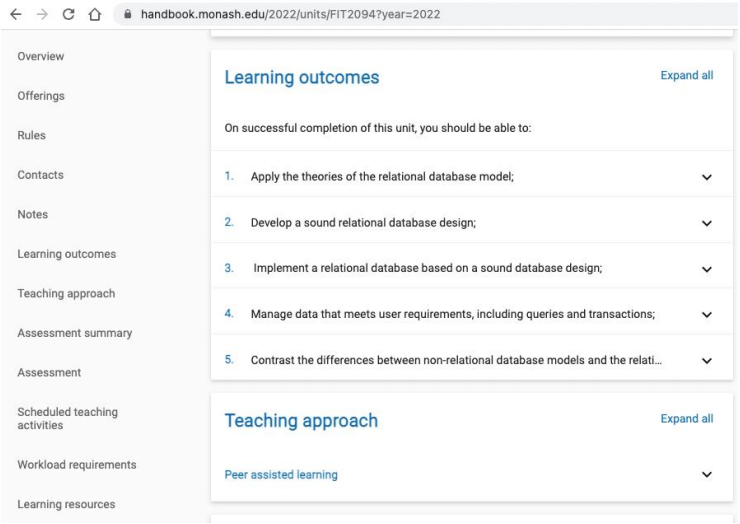


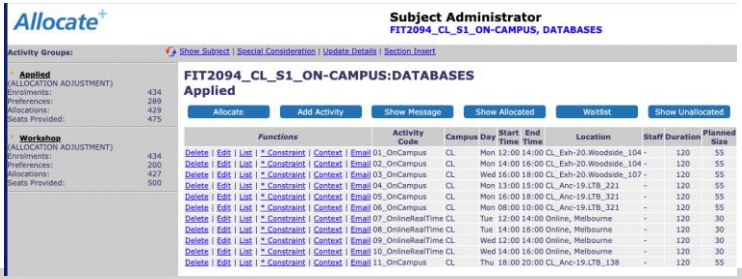
Requirements Definition

- Identify and analyse user views.
- A 'user view' may be a report to be produced or a particular type of transaction that should be supported.
- Corresponds to the external level of the ANSI/SPARC architecture.
- Output is a statement of specifications which describes the user views' particular requirements and constraints.

Different views of the underlying data

Student





Admin

Staff & Student

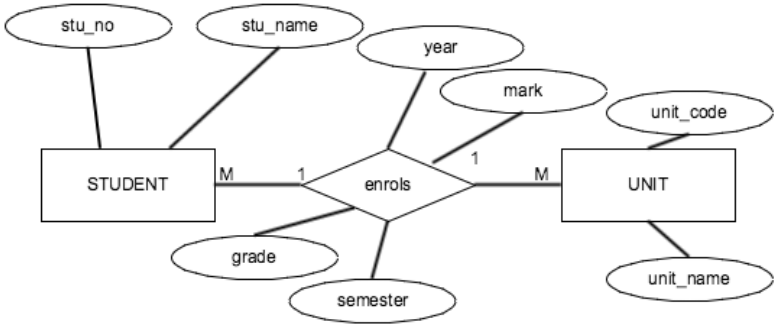
ER Modeling

- ER (Entity-Relationship) model developed by Peter Chen in 1976 to aid database design.
- used for conceptual model (ERD).
- ER diagrams give a visual indication of the design.
- Basic components:
  - Entity
  - Attribute
  - Relationship

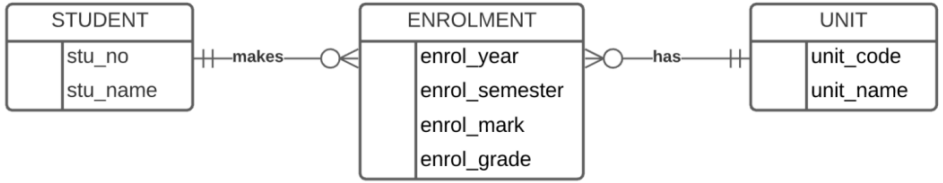
Conceptual Design

- Develop the enterprise data model.
- Corresponds to the conceptual level of the ANSI/SPARC architecture.
- *Independent of all physical implementation considerations (the type of database to be used)*.
- Various design methodologies may be employed such as UML, ER (Entity-Relationship) Modelling and Semantic Modelling.
- ER consists of ENTITIES and RELATIONSHIPS between entities
  - An ENTITY will have attributes (things we wish to record), one or more of which will identify an entity instance (called the KEY)

ERD - Notation



Chen



Information Engineering/James Martin/Crows foot

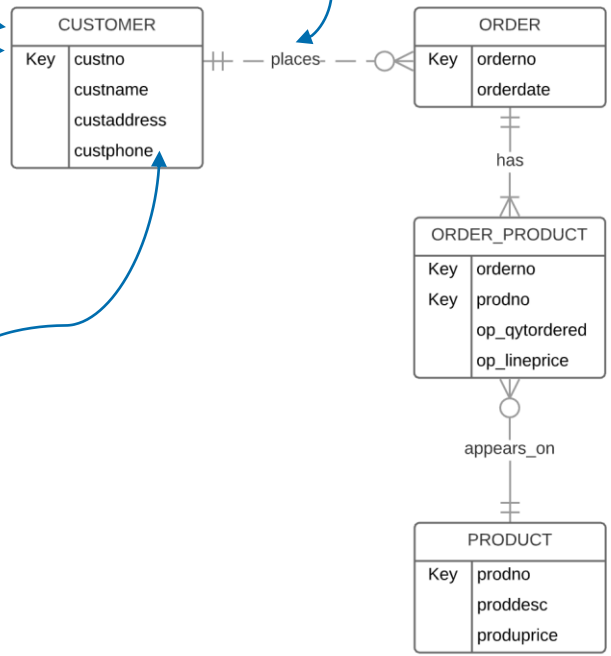
**** This is what we will be using***

Conceptual Level (ER Model)

ENTITY
Collection of "Customer(s)"

KEY ATTRIBUTE(S)
Instance identifier

NON KEY ATTRIBUTE
Other non-key attributes



RELATIONSHIP
*Connects entities - on a conceptual model this is the **ONLY** manner in which entities are connected*

Assignment 1A

Q1. In your group, discuss your pre Forum identification of the Monash Software Entities. How many entities did your group identify:

A. 2
B. 4
C. 5
D. 6

Conceptual Level (Monash Software Entities)

TRAINING	
Key	training_code

TEAM	
Key	team_no

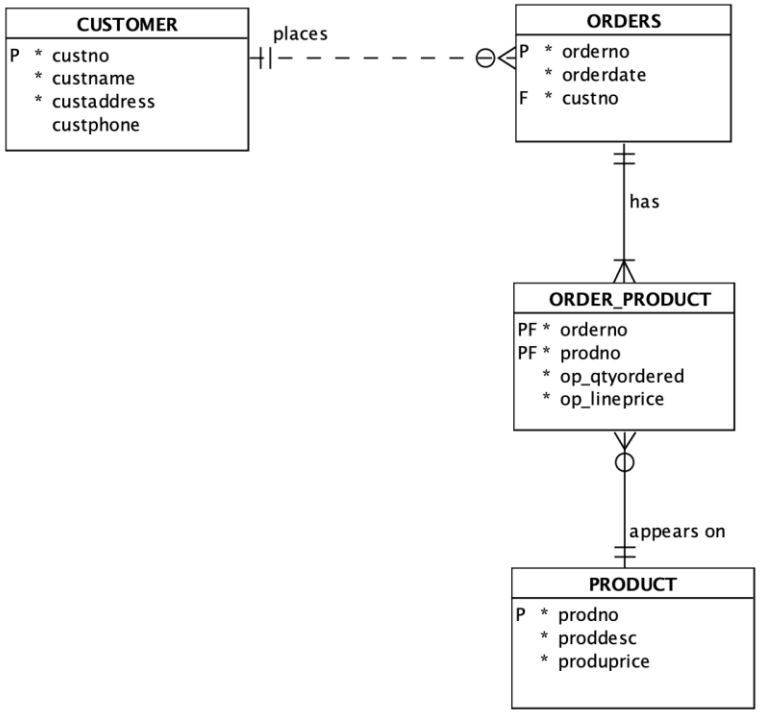
EMPLOYEE	
Key	emp_no

FAMILY_MEMBER	
Key	

Logical Design

- Develop a data model which targets a particular database type (e.g. relational, hierarchical, network, object-oriented, noSQL).
- Independent of any implementation details which are specific to any particular vendors DBMS package.
- Normalisation technique (see session 4) is used to test the correctness of a relational logical model.

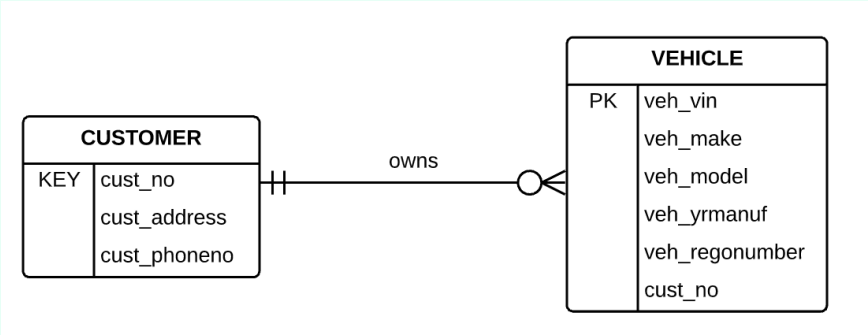
Logical Level (Logical Model - Relational)



Assignment 1B

Q2. Is the diagram shown below a valid *Conceptual Model*?

Be prepared to justify your answer with why you chose this option



A. Yes
B. No
C. Depends on how it is implemented in the database

Physical Design

- Develop a strategy for the physical implementation of the logical data model.
- Choose appropriate storage structures, indexes, file organisations and access methods which will most efficiently support the user requirements (not part of unit).
- Physical design phase is dependent on the particular DBMS in use.
- ANSI/SPARC internal level.

Physical Level – Starting point

Oracle Database 12c | Relational_1 | Generate

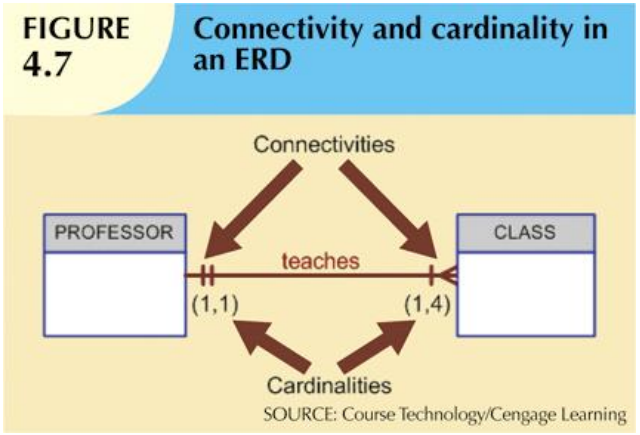
```
 8  CREATE TABLE customer (
 9      custno        NUMBER(7) NOT NULL,
10      custname      VARCHAR2(50) NOT NULL,
11      custaddress   VARCHAR2(50) NOT NULL,
12      custphone     CHAR(10)
13  );
14
15  COMMENT ON COLUMN customer.custno IS
16      'Customer number';
17
18  COMMENT ON COLUMN customer.custname IS
19      'Customer name';
20
21  COMMENT ON COLUMN customer.custaddress IS
22      'Customer address';
23
24  COMMENT ON COLUMN customer.custphone IS
25      'Customer phone number';
26
27  ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY ( custno );
28
29  CREATE TABLE order_product (
30      orderno         NUMBER(7) NOT NULL,
31      prodno          NUMBER(7) NOT NULL,
32      op_qtyordered   NUMBER(3) NOT NULL,
33      op_lineprice    NUMBER(8, 2) NOT NULL
34  );
35
```

The database *schema*

Important rule for Conceptual Modelling

- ***All that is described in the brief has been included and all that has been included was described in the brief***
  - Every entity, attribute and relationship described in the brief **has been included**, and
  - **Must not add** entities, attributes and relationships which are not included as part of the brief, and
- In a real life scenario if there are concerns about features of the brief, discuss with client
  - For assignments:
    - your client will be the ed forum
    - may make assumptions provided they do not violate this rule

CONNECTIVITY/CARDINALITY



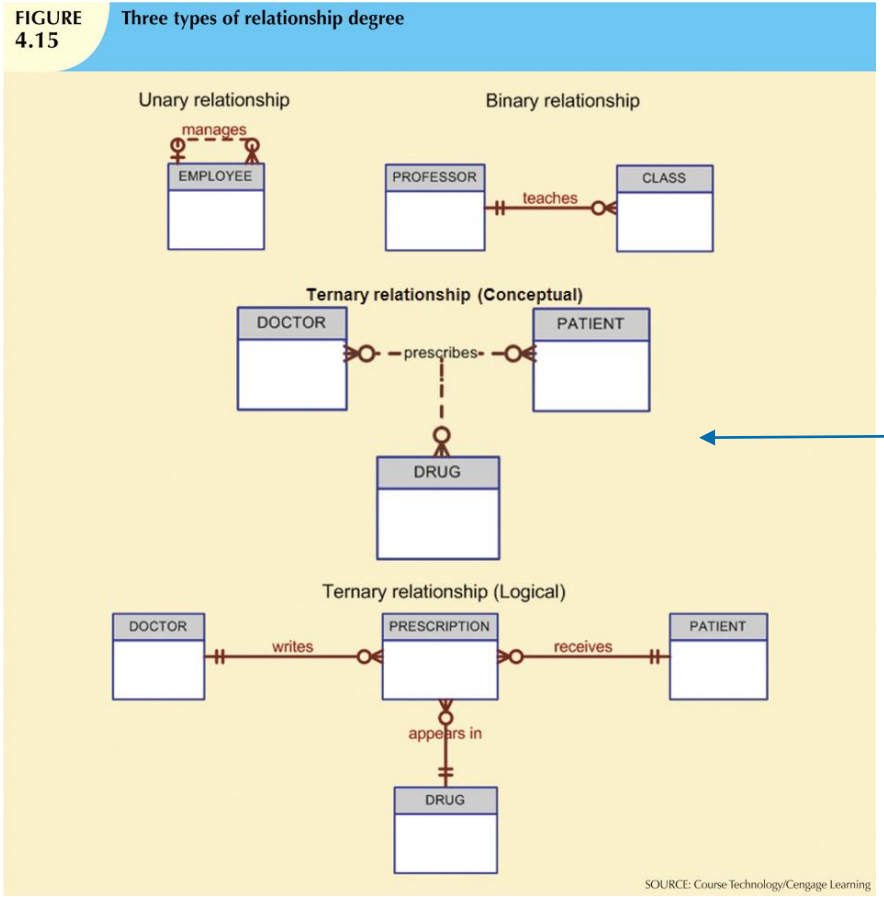
CONNECTIVITY

one to one

one to many

many to many

In general for Crows Foot notation specific cardinalities are not shown as above eg. (1,4), instead cardinality is depicted via min and max using standard symbols (Inside symbol = min, outside symbol = max)

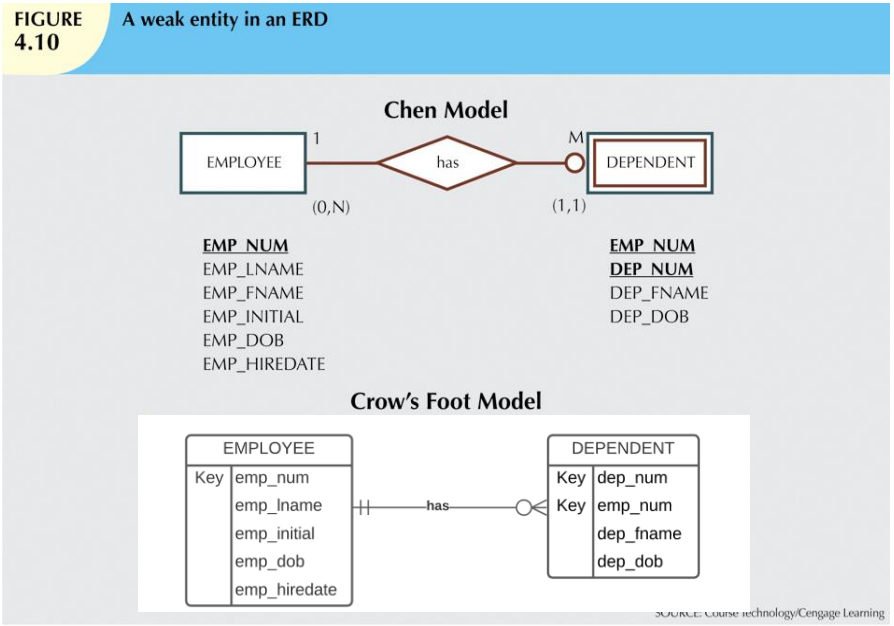


Note this is not an acceptable form of a conceptual model in Crow's Foot notation (*relationship lines cannot join*)

Weak vs Strong Entity

- Strong entity
  - Has a key which may be defined without reference to other entities.
  - For example EMPLOYEE entity.
- Weak entity
  - Has a key which requires the existence of one or more other entities.
  - For example FAMILY entity - need to include the key of employee to create a suitable key for family
- Database designer often determines whether an entity can be described as weak based on business rules
  - customer pays monthly account
    - Key: cust_no, date_paid, or
    - Key: payment_no (surrogate? – not at conceptual level)

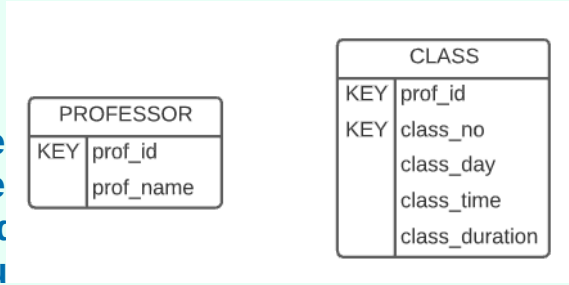
Weak vs Strong Entity



Note the Crow's Foot model shown here has been modified from the text version

Q3.The client indicates that a CLASS is identified by a combination of the the prof_id and the assigned class number for the professor (1st class, 2nd class, 3rd class etc):

prof_id,	class_no,	class_day, ...
1,	1,	Tue
1,	2,	Tue
1,	3,	Wed
2,	1,	Thu
2,	2,	Tue
....		



This business rule is captured in the provided diagram. Pick the correct statement for this diagram.

A. Both entities are strong entities
B. PROFESSOR is a strong entity, CLASS is a weak entity
C. CLASS is a strong entity, PROFESSOR is a weak entity
D. Both entities are weak entities

Identifying vs Non-Identifying Relationship

- **Identifying**

- Identifier of A is part of identifier of B.

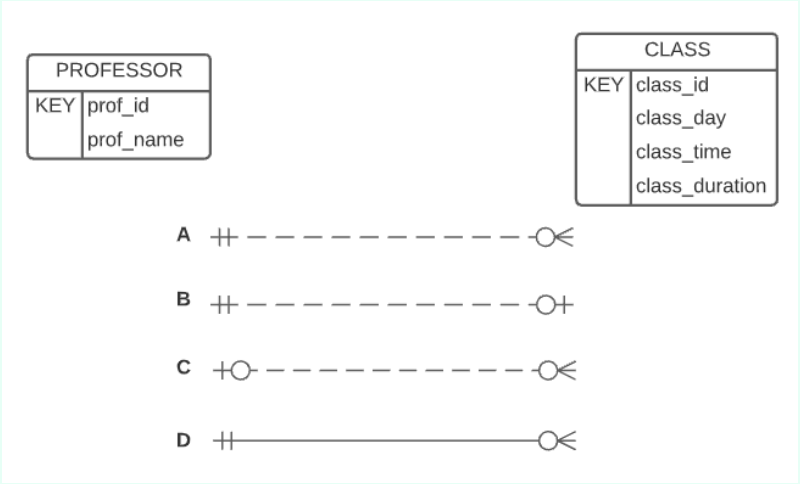
- Shown with solid line
- ENROLMENT - STUDENT
  Enrolment key includes student id, which is an identifier of student.

- **Non-identifying**

- Identifier of A is NOT part of identifier of B.

- Shown with broken line
- Department no (identifier of department) is not part of Employee's identifier.

Q4. The client indicates that a professor may teach several classes, but some professors do not have any assigned classes. Each class is taken by only one professor. Note that in this diagram, each class has a unique class id (class_id). Pick the most appropriate relationship for this business rule.



Types of Attributes

- Simple
  - Cannot be subdivided
  - Age, sex, marital status
- Composite
  - Can be subdivided into additional attributes
  - Address into street, city, zip
- Single-valued
  - Can have only a single value
  - Person has one social security number
- Multi-valued
  - Can have many values
  - Person may have several college degrees
- Derived
  - Can be derived with algorithm
  - Age can be derived from date of birth
- Attribute classification is driven by Client requirements
  - Phone Number?

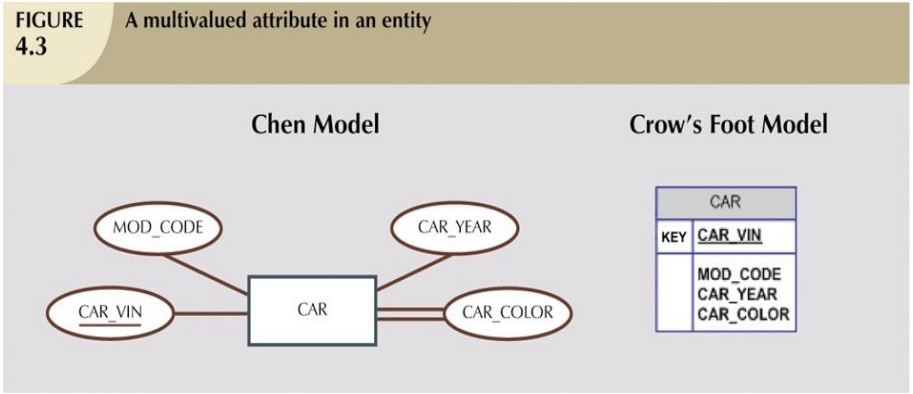
Q5. The HiFlying case study indicates "*HiFlying establishes a drone hire rate as a cost per hour for customers to rent this particular drone (rates per hour are often changed over the life of the drone, as it ages, although they are only interested in recording the current cost per hour for the drone).* " Note that although the hire rate may change over the life of the drone, it is not directly related to the hours flown.

What type of attribute is the drone hire rate?

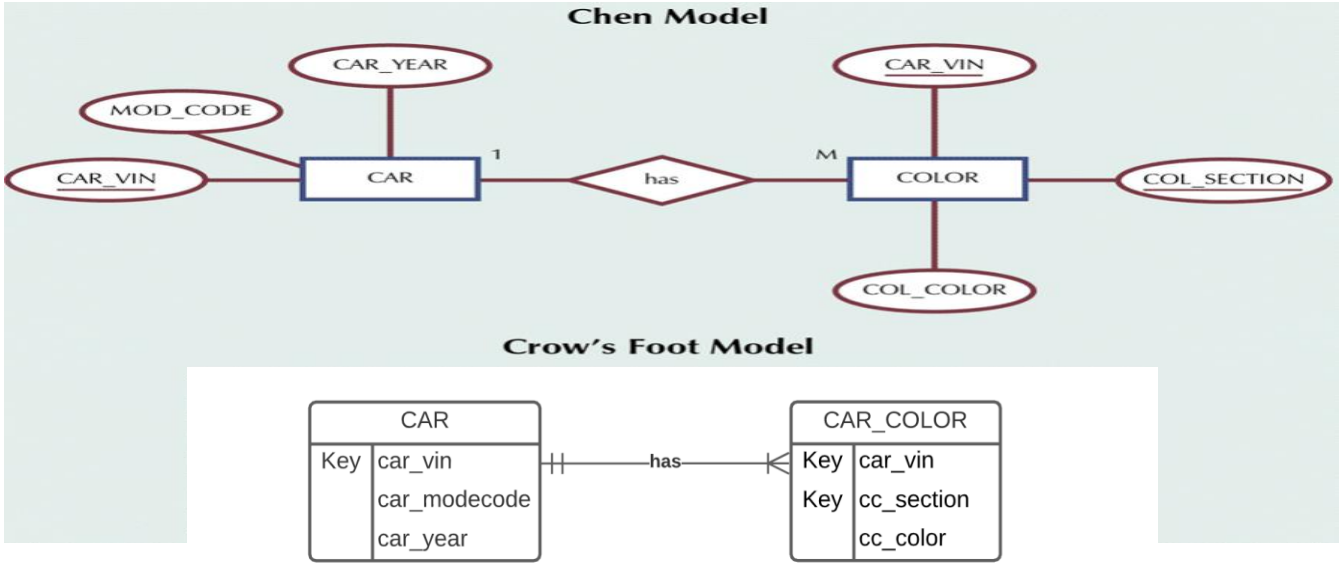
A. Simple
B. Composite
C. Single-valued
D. Multi-valued
E. Derived

Multivalued Attribute

- An attribute that has a list of values.
- For example:
  - Car colour may consist of body colour, trim colour, bumper colour.
- Crow's foot notation does not support multivalued attributes. Values are listed as a separate attribute.

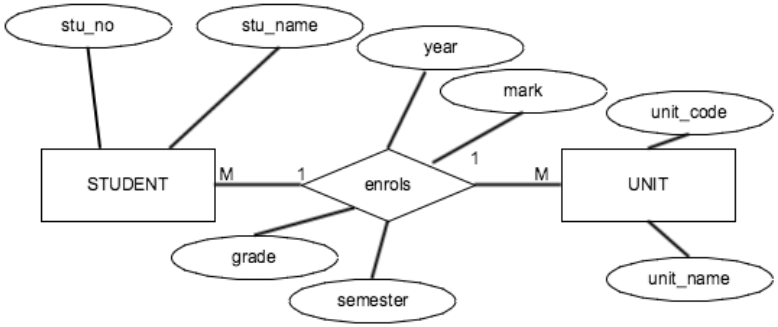


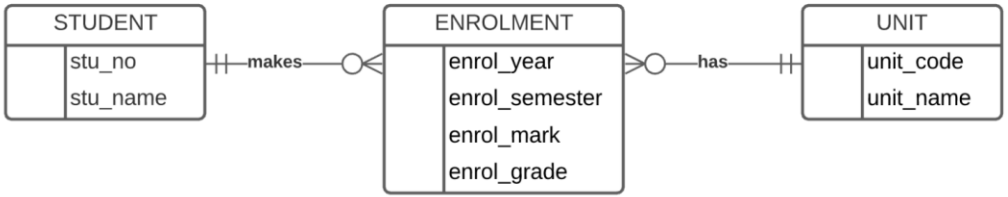
Resolving Multivalued Attributes



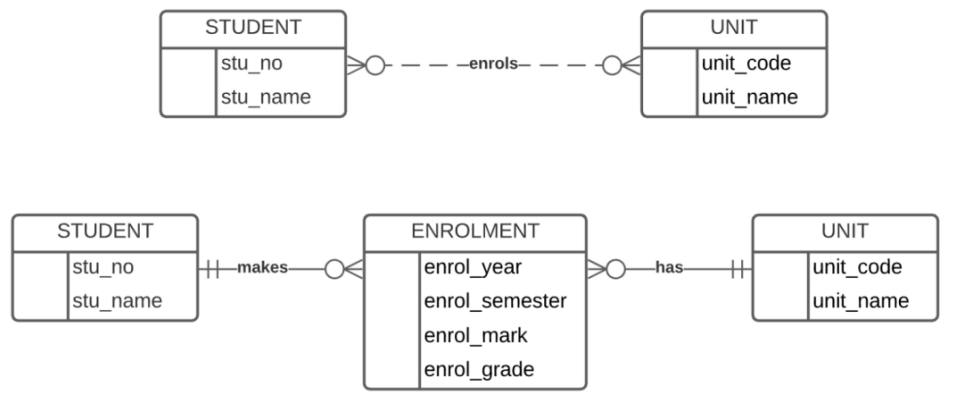
Note the Crow's Foot model shown here has been modified from the text version

Associative (or Composite) Entity

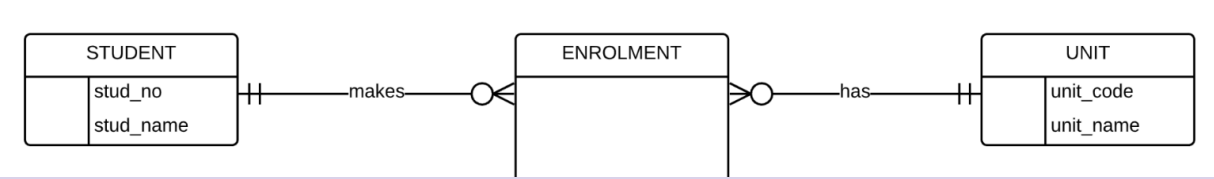




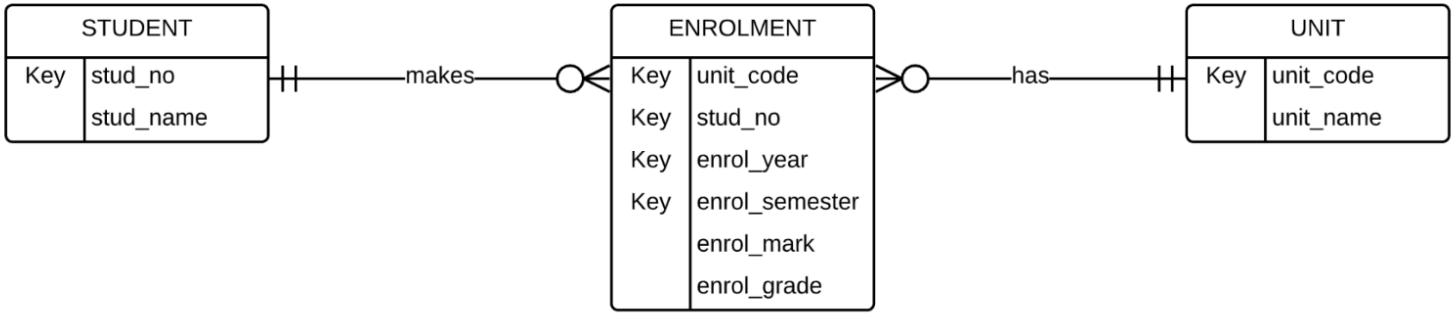
Associative (or Composite) Entity



Q6. Show all attributes for the three entities and add KEYS:



Associative or Composite Entities



Q7. STEP 1: List ALL entities and their key attribute/s which exist in the case study.

For example:

HiFlying Drones

HiFlying Drones

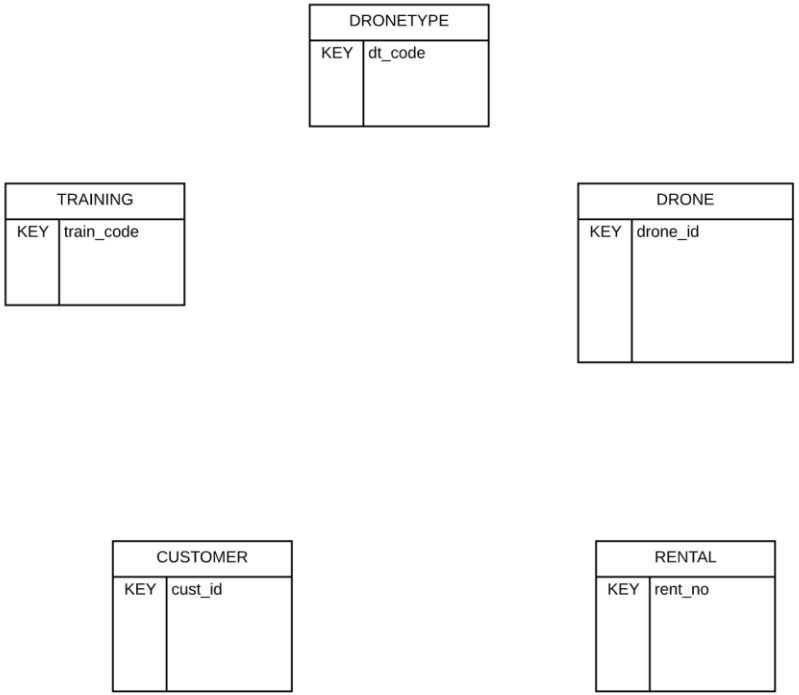
HiFlying Drones is a company which rents drones out to customers.

The company purchases a range of different types of drones in order to meet their customers' requirements. Each type of drone they purchase is assigned a drone type code (e.g. RH4) as the identifier for this type.

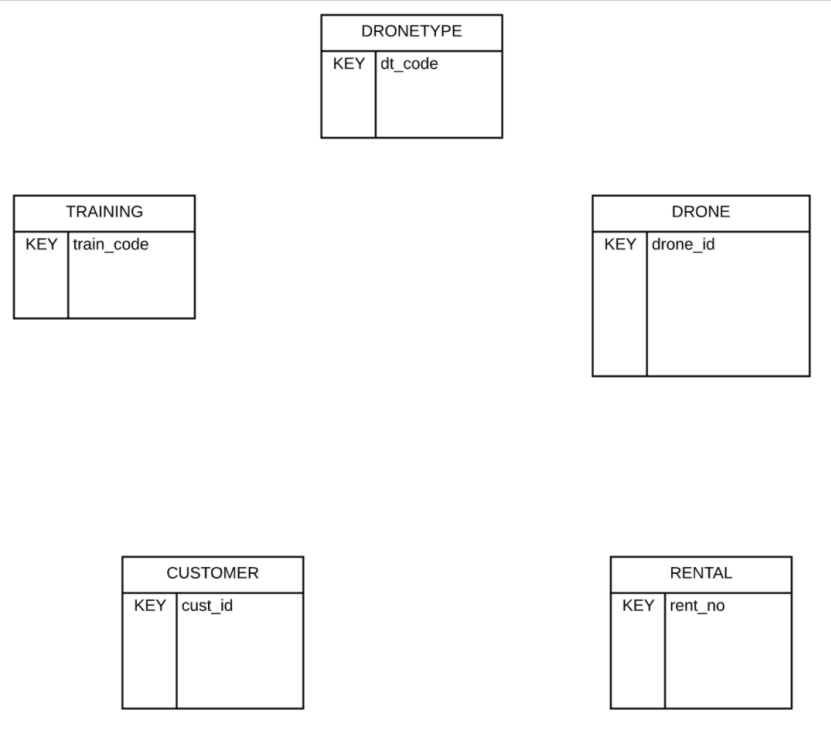
To keep track of the drones they purchase, HiFlying identifies each drone with a drone id. When a new drone is added to the system the type of the drone, the date it was purchased and the purchase price are recorded. In addition, HiFlying establishes a rate as a cost per hour for each drone type. This rate has changed over the life of the company.

DRONE	
KEY	drone_id

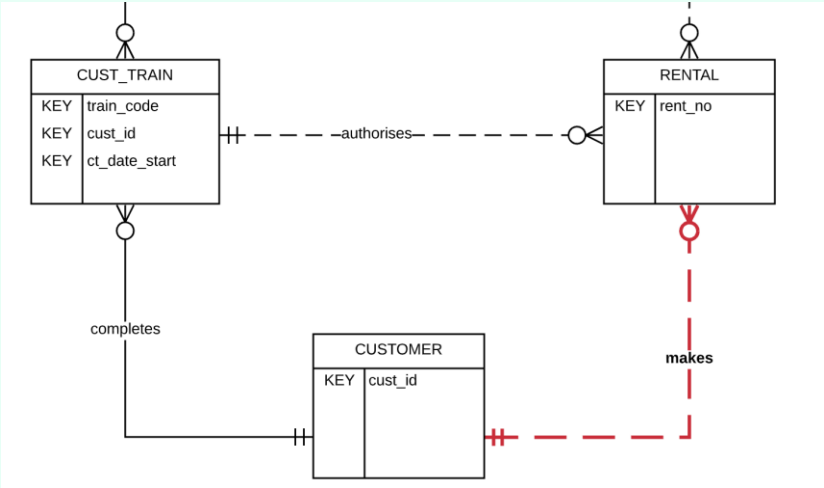
HiFlying Drones - Step 1 Identify Main Entities



Q8. STEP 2: Identify the relationships which exist between these entities (remember to add an appropriate verb):

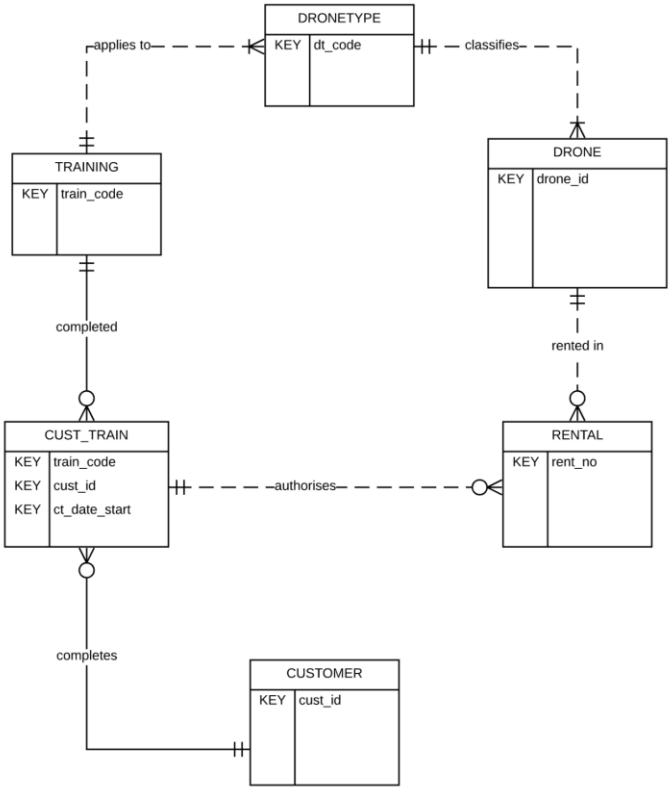


Q9. Since a customer makes a rental, should the database designer include a relationship between RENTAL and CUSTOMER?

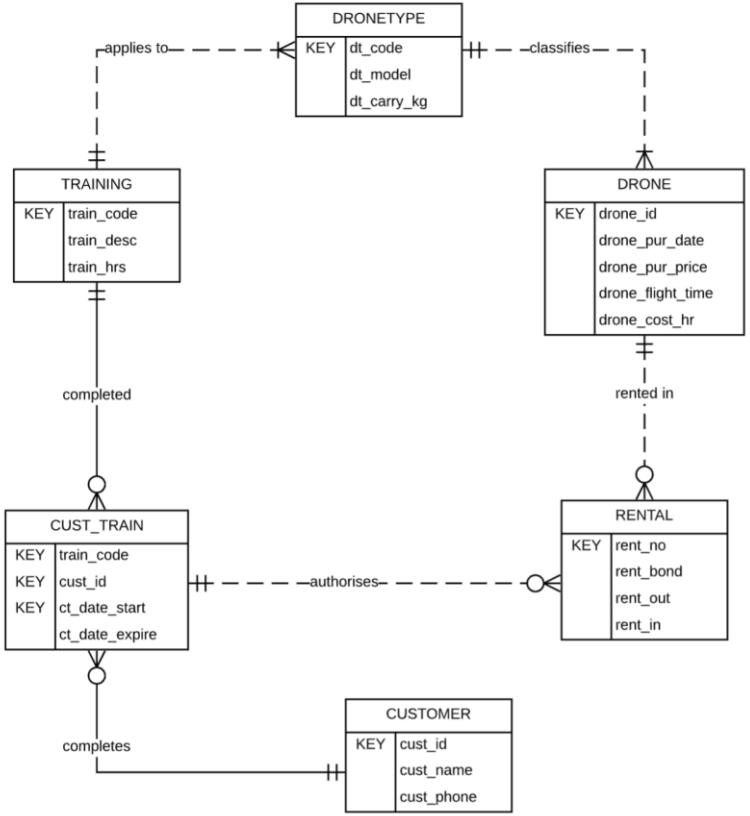


A. Yes, it is an important relationship to capture
B. No, it is redundant information
C. It depends on the client's requirements

HiFlying Drones - Step 2 Identify Relationships



HiFlying Drones - Step 3 Add Non-Key Attributes - Final Model



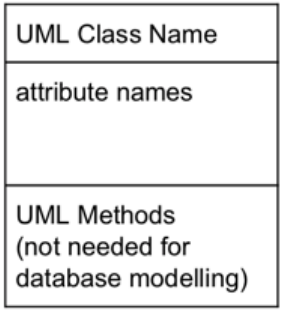
UML Notation
for Conceptual Modelling

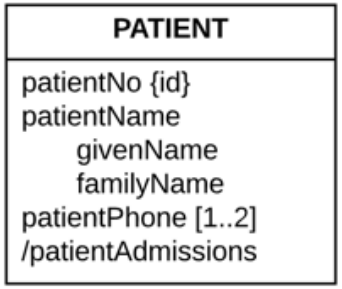
Unified Modeling Language (UML)

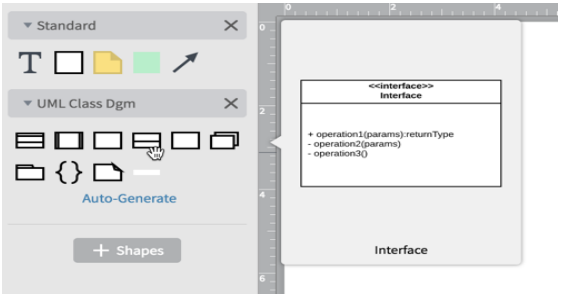
- The way that data is organised in a database is very different to the way it is organised in an OO program
  - eg. inheritance
- Use a **subset** of UML notation for database modelling
  - Conceptual
  - Logical
- Several vendors support Database Modelling via UML, some examples:
  - Star UML
  - Altova UModel
- Variety of standards adopted, not widely used in practice

UML Notation for the unit

Standard UML Diagram is used as the basic structure:







{id} - indicates KEY

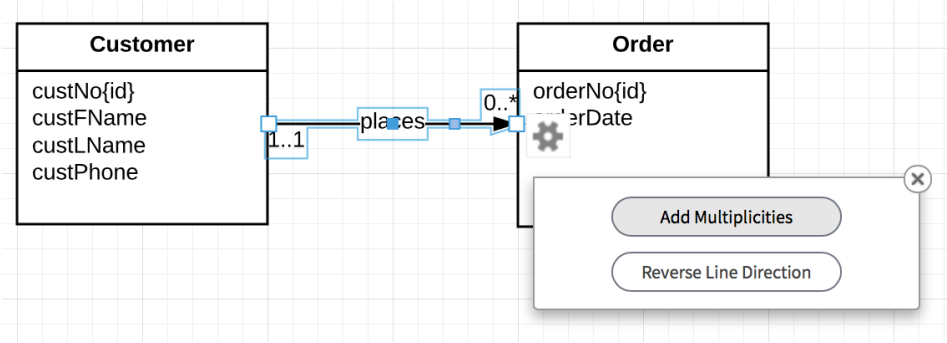
Indentation for composite attribute

[n..m] - multivalued attribute

/ - calculated attribute

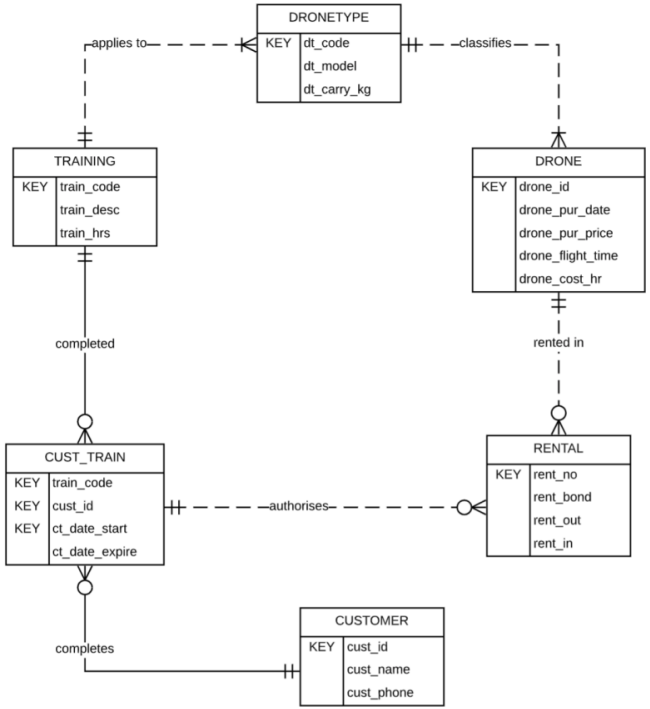
use Interface shape as no methods being added

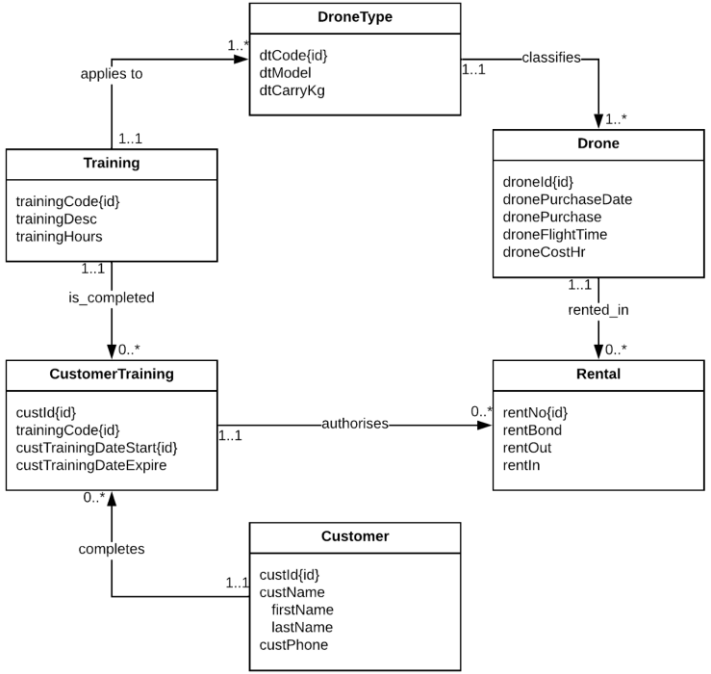
UML - Relationship



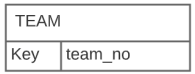
Relationship lines - directed line, arrowhead at M end, add Multiplicities (minimal and maximal on each side)

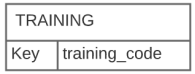
HiFlying Drones - UML

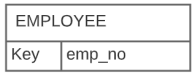




Conceptual Model (Monash Software)







You have completed

- **Step 1 identify entities and keys**

of the modelling process for Monash Software

After the Forum please proceed and complete:

- **Step 2 Identify Relationships**, and
- **Step 3 Add all non key attributes**
- **Step 4 Map your ERD to UML**

Videos will be provided showing the full process (available from Friday 5pm).