FIT2094-FIT3171 2019 S1 -- Week 4 eBook

Credits and Copyrights:
Authors: FIT3171 2018 S2 UG Databases
FIT Database Teaching Team
Maria Indrawan, Manoj Kathpalia, Lindsay Smith, et al

Change Log:

- Formatted and Imported from Alexandria March 2019.

4.0. Conceptual Modelling

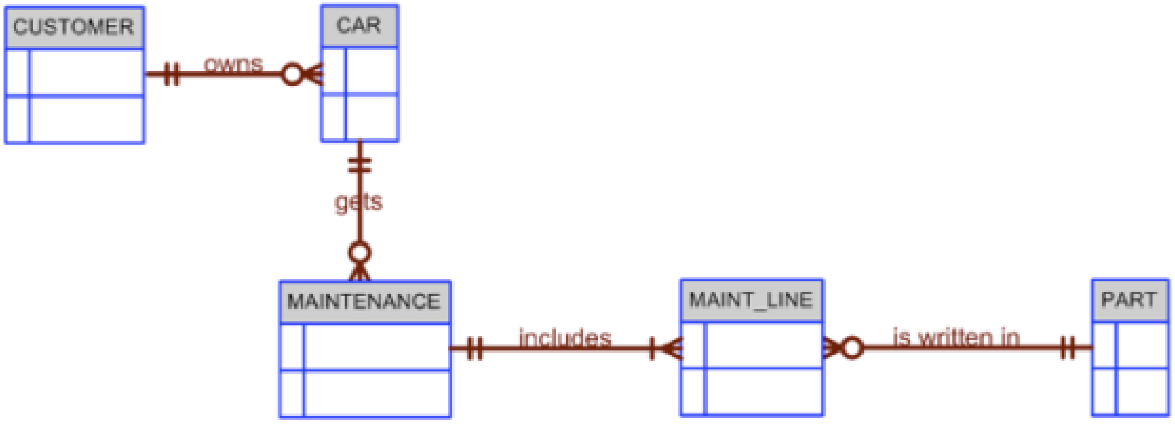
This section will start with some very important theory questions that you should know.

4.0.1. Conceptual Design – Theory Questions

Reference: Coronel, C and Morris, Database Systems: Design, Implementation & Management, Chapter 4, Selected Review Questions and Problems.

1. What two conditions must be met before an entity can be classified as a weak entity? Give an example of a weak entity.
2. What is a identifying relationship, and how is it depicted in a Crow's Foot ERD?
3. Given the business rule "an employee may have many degrees," discuss its effect on attributes, entities, and relationships. (Hint: Remember what a multivalued attribute is and how it might be implemented.)
4. What is a composite entity, and when is it used?
5. Suppose you are working within the framework of the conceptual model in the Figure below:

The Conceptual Model for Question 5



- Write the business rules that are reflected in it.
- Identify all the cardinalities.

6. What is a recursive relationship? Give an example.
7. How would you (graphically) identify each of the following ER model components?
    - an entity
    - the cardinality (0,N)
    - a weak relationship, and
    - a strong relationship
8. Discuss the difference between a composite key and a composite attribute. How would each be indicated in an ERD?
9. What two courses of action are available to a designer when he or she encounters a multivalued attribute?

10. What is a derived attribute? Give an example.
11. How is a relationship between entities indicated in an ERD? Give an example, using the Crow’s Foot notation.
12. Discuss two ways in which the 1:M relationship between COURSE and CLASS can be implemented. (Hint: Think about relationship strength.)
13. How is a composite entity represented in an ERD, and what is its function? Illustrate using the Crow’s Foot model.
14. Briefly, but precisely, explain the difference between single-valued attributes and simple attributes. Give an example of each.
15. What are multivalued attributes, and how can they be handled within the database design?

4.1. Using Tools to draw ERD

4.1.1. Tools to draw ER diagrams

There are several tools available to draw ER diagrams. Some of them are available to be used within a web browser. Some examples of these are:

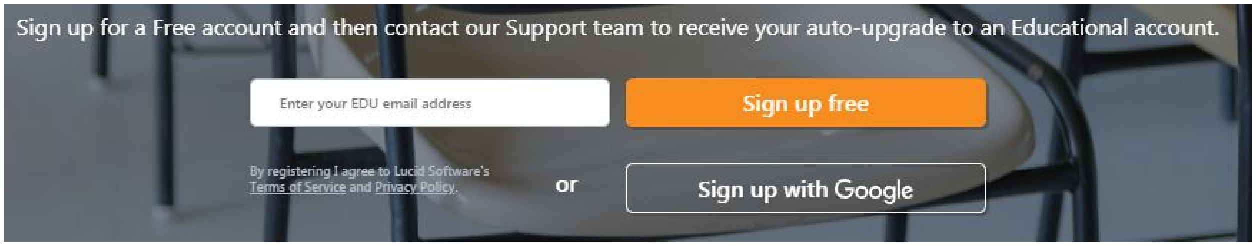
- Lucidchart – this product is a browser-based diagramming tool; it is able to draw a wide range of different diagrams, including ER Diagrams, or
- any other drawing package you wish. One excellent alternative is Gliffy (https://www.gliffy.com/)
- Another one is https://creately.com/, which does not require registration to quickly draw diagrams; however it still uses Flash and does not allow for diagrams to be saved unless one registers.

NOTE: At this stage of our study, we do not wish to use a CASE tool – it is important that we first establish a clear understanding of Entity Relationship modelling.

4.1.2. Setting up Lucidchart

The Lucidchart Education account details and sign up are here.

You will see a screen similar to the following (source: Lucidchart)



Sign up for a Free account and then contact our Support team to receive your auto-upgrade to an Educational account.

Enter your EDU email address

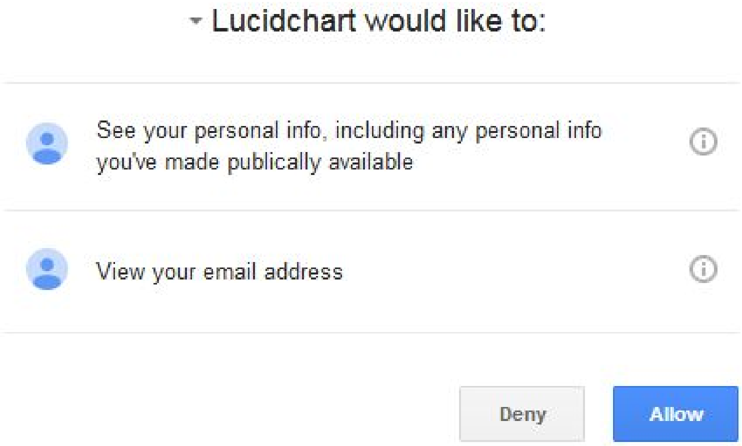
By registering I agree to Lucid Software's Terms of Service and Privacy Policy.

There are two ways of doing this.

4.1.2.1. METHOD ONE: GOOGLE SINGLE SIGN-ON LOGIN - EASIEST, MOST STRAIGHTFORWARD

As Monash uses Google Apps (GMail) as their email provider, you can simply click on “**Sign up with Google**” - i.e. the SECOND link (not the one in orange). Then, enter your **Monash GMail account** (NOT your personal account) -- e.g. abcd1234@student.monash.edu -- then complete your login with Okta.

Finally, allow Google to integrate with Lucidchart by selecting **Allow** after reading the security prompts.



Lucidchart would like to:

See your personal info, including any personal info you've made publically available

View your email address

4.1.2.2. METHOD TWO: WITHOUT USING GOOGLE SINGLE SIGN-ON

Students can also *create their own account* by signing up for a free account with your Monash educational email address at the URL listed above (i.e. the “**Sign up free**” orange button). This will result in an email being sent to your student account confirming your account details and providing a link to set your password. Alternatively, you can set your password when you first login … (see the top left of your first screen).

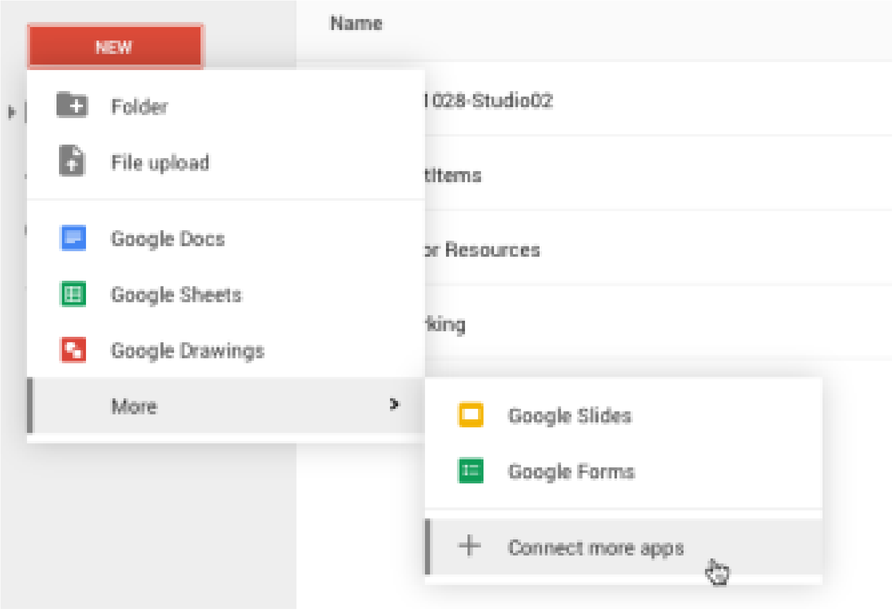
4.1.3. First Steps in Lucidchart

As a first step, you should look at the provided tutorials, in particular, "Entity Relationship Diagrams" (begin with "Manual ERD Creation", note that the model we are asking you to build should not include the "Type" column)

After you have set up your account, Lucidchart can be logged into directly at the Lucidchart site: https://www.lucidchart.com/ or alternatively accessed from your student Google Drive (from my.monash select the "Drive" link).

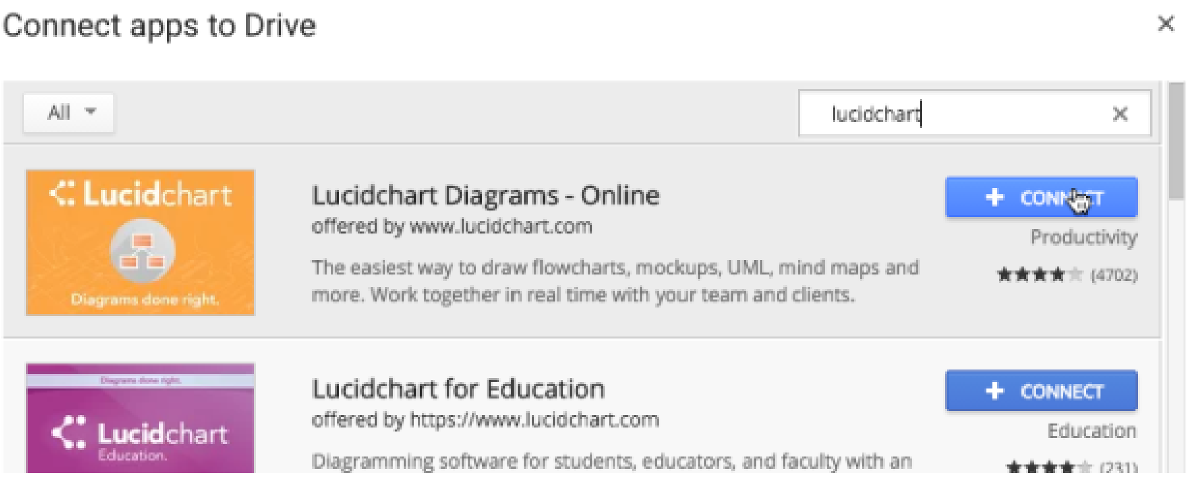
4.1.2.1. Connect LucidChart to your Google Drive [OPTIONAL]

Connect to Google Drive, click "New" and then more, and "Connect more apps"



This will open the app picker which lists a wide range of applications that can be added to your Google drive.

You can browse through and find Lucidchart or alternatively type “Lucidchart” in the top right search box:



Connect apps to Drive

Lucidchart Diagrams - Online
offered by www.lucidchart.com
The easiest way to draw flowcharts, mockups, UML, mind maps and more. Work together in real time with your team and clients.

Lucidchart for Education
offered by https://www.lucidchart.com
Diagramming software for students, educators, and faculty with an

Select the “**+ CONNECT**” button for Lucidchart Diagrams – Online

After this process has been completed Lucidchart charts created from within your Google Drive will be stored in your Google Drive and on the Lucidchart server. Full details of the integration are available here.

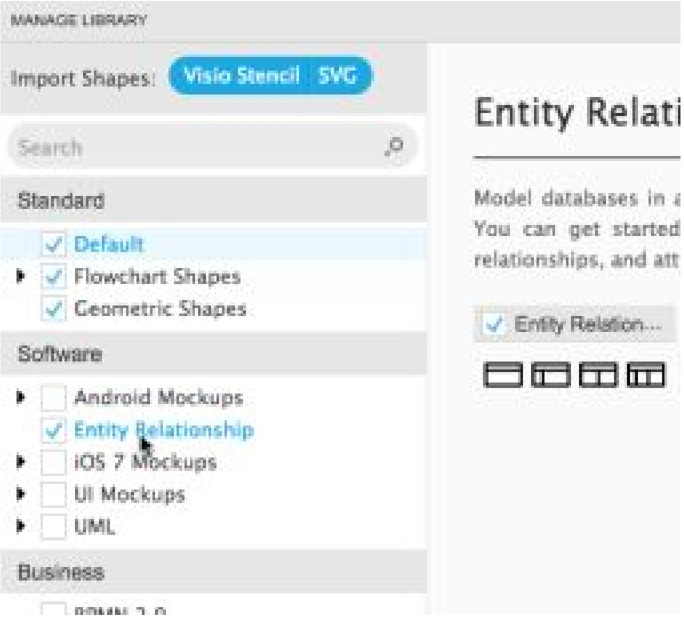
To create a new Lucidchart – simply select New in Google Drive, and then select “Lucidchart” (you may need to expand “More” to see the Lucidchart option).

If this is the first document you have created as a Lucid Chart from your Google Drive you will be asked to approve Lucidchart’s access to your Google Drive – please ensure you select “Accept”. This will then transfer you into the Lucidchart workspace with a new document open.

4.1.2.2. Creating a new LucidChart Diagram

Either use Google Drive, as above, or login directly to LucidChart. If you log in directly you will need t0 select to create a new “Entity Relationship (ERD)”, then Blank ERD and then “Start Drawing”.

If the ERD shapes are not listed in the left panel, add the ERD shapes by selecting “More Shapes” (bottom left) and then checking “Entity Relationship” and then “Save”:



Standard
- ☑ Default
- ☑ Flowchart Shapes
- ☑ Geometric Shapes

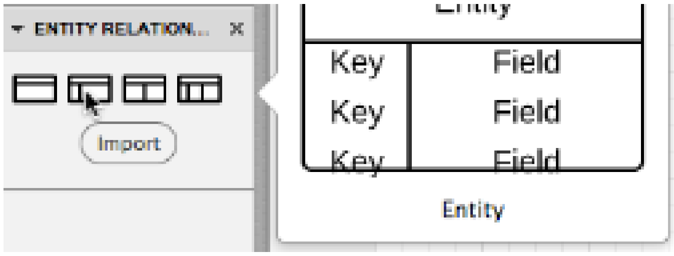
Software
- ☐ Android Mockups
- ☑ Entity Relationship
- ☐ iOS 7 Mockups
- ☐ UI Mockups
- ☐ UML

Business
- ☐ BPMN 2.0

Model databases in
You can get started
relationships, and att

☑ Entity Relation...

The symbol we will use to represent an entity is the second symbol from the left:

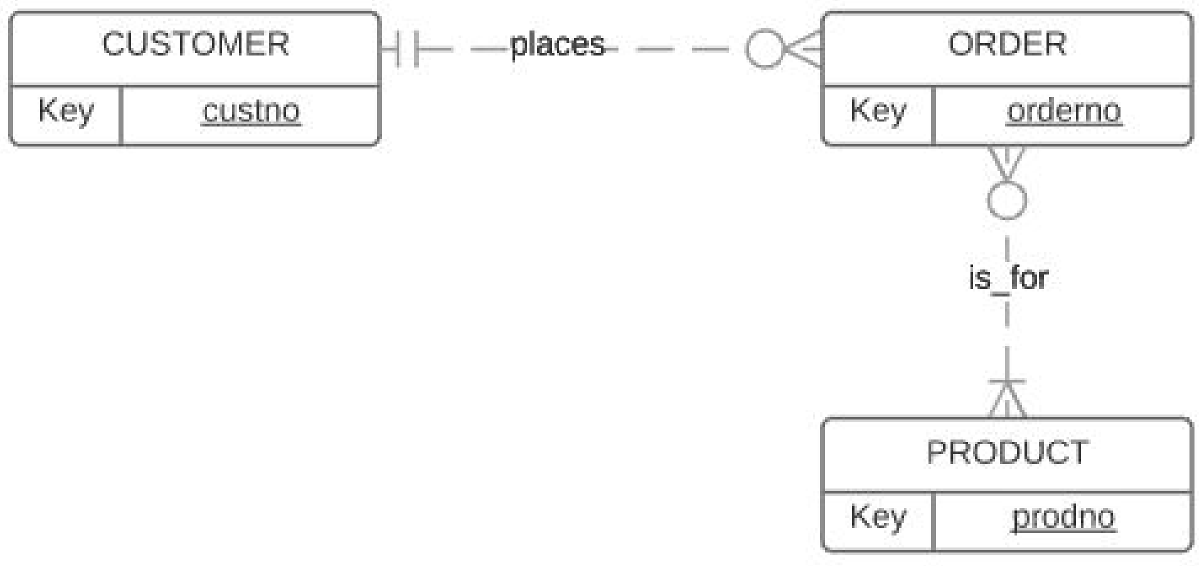


4.1.3. Using Lucidchart

Given a scenario represented by the following entities, where customers place orders for products:

- CUSTOMER – customer number, name, address, phone number
- ORDER – order number, order date and for each product ordered the quantity ordered and the total line price
- PRODUCT – product number, product description and product unit price

An initial ERD using Lucidchart for this scenario would be:



This CONCEPTUAL ERD only shows the keys of each of the entities, sometimes an ERD is drawn such that it will show all of the attributes for the scenario. In such a complete (or full) ERD **you must not show or label foreign keys, the use of foreign keys indicates that you are looking at a logical model** where a choice has been made to use a relational database, **rather than a conceptual model.**

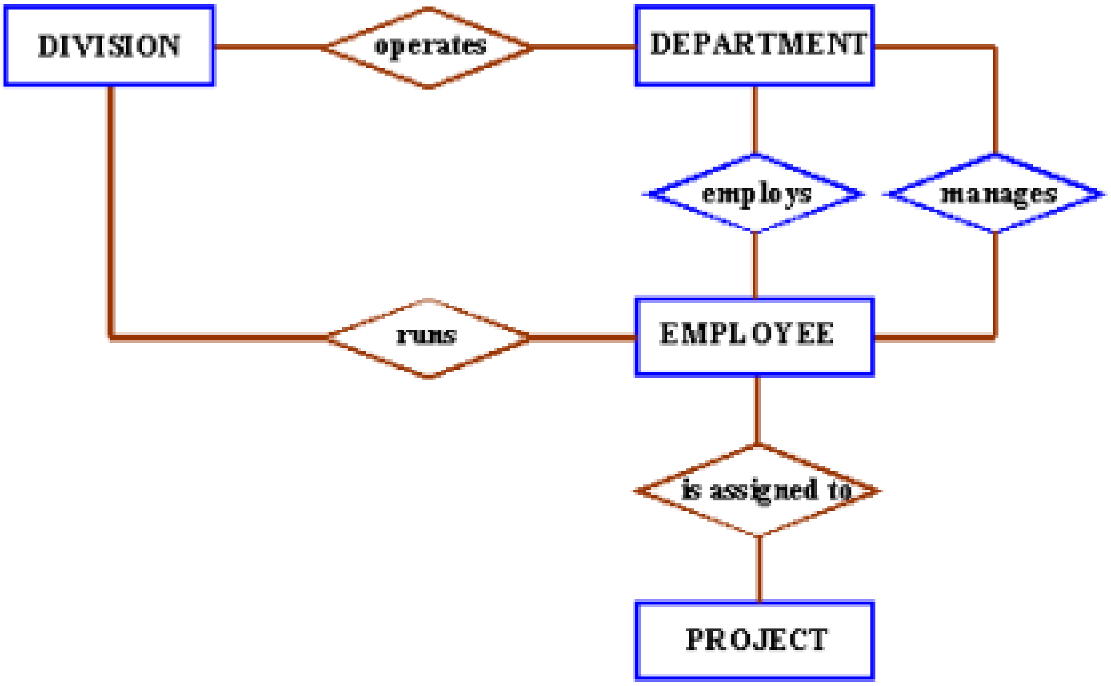
Prepare the ERD shown above using your choice of drawing tool and then create a copy of this initial model and add all attributes listed in the scenario above to your copy. In Lucidchart, you can easily create a copy by selecting the bottom tab of the diagram, click the down arrow and then select “Duplicate”.

4.2. Building Conceptual Models

4.2.1. Conceptual Modelling – Practical Work

For this week’s lab exercises, we are going to prepare conceptual models (Entity Relationship Diagrams – ERDs) for a number of scenarios.

At this point we are not interested in the database implementation of these models, our aim will be to model without any consideration of the database system in which the model may ultimately be implemented - hence CONCEPTUAL MODELS.



1. Use the above diagram and the business rules below, create an Entity Relationship Diagram using crows-foot notations.

   - Include:
     - all appropriate connectivities,
     - all cardinalities and
     - at least the minimum number of attributes required to implement the model
   - Business Rules:
     - A department employs many employees, but each employee is employed by one department.
     - Some employees, known as “rovers,” are not assigned to any department.
     - A division operates many departments, but each department is operated by one division

    - An employee may be assigned to many projects, and a project may have many employees assigned to it.
    - A project must have at least one employee assigned to it.
    - One of the employees manages each department, and each department is managed by one employee
    - One of the employees runs each division, and each division is run by one employee.

2. Prepare an Entity Relationship Diagram (ERD) showing all attributes and the identifier of each entity for the following description of a Property Rental System:
    - Properties are rented by tenants. Each tenant is assigned a unique number by the Agency. Data held about tenants include family name, given name, property rented, contact address – street, city, state, postcode & telephone number. A tenant may rent more than one property and many tenants may rent parts of the same property (eg. a large shopping complex).
    - Properties are owned by owners. Each property is assigned a unique building number. The agency only recognises a single owner for any of the properties it handles. The owner, address, and value are recorded for each property. Also, the lease period and bond are recorded for each property or sub-property rented. An owner may own several properties. For each owner an owner number is assigned, the owner name is also recorded.

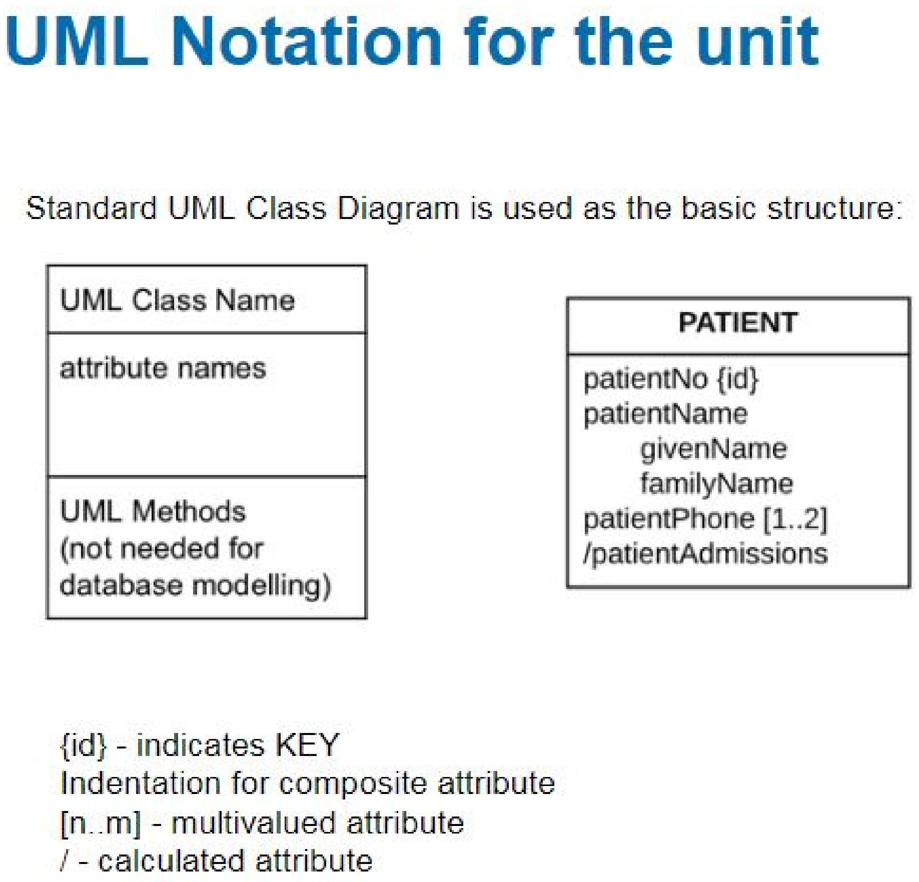
- Properties are subject to damage and the agency records all instance of damage to its properties – property, date, type of damage and repair cost are recorded. Repair costs are charged directly to tenants
- Normal property maintenance is also noted – property, date, type of maintenance and cost are recorded. Maintenance costs are charged to the property owner.
- Tenants pay accounts to the Agency – these consist of weekly rental payments, bond payments (for new properties) and damage bills. The date of payment, tenant, property, type of account (Rental, Bond, Damage) and amount are recorded. Each payment is assigned a payment number.

4.3. Pre-Lecture Notes

These are notes you may find useful to read through before the lecture, adapted from Lindsay Smith's lecture material. NOTE: THESE ARE NOT THE FINAL LECTURE SLIDES.

Read up the pre-lecture notes below.

IMPORTANT: AS THE LECTURE FOCUSES ON THE 'FLIPPED CLASSROOM' APPROACH - I.E. MORE INTERACTIVE DISCUSSION AND LESS ON DISCOVERING NEW MATERIAL - THE THEORY SLIDES BELOW ARE PROVIDED FOR YOUR READING CONVENIENCE BEFORE THE LECTURE.



UML Notation for the unit

Standard UML Class Diagram is used as the basic structure:

{id} - indicates KEY

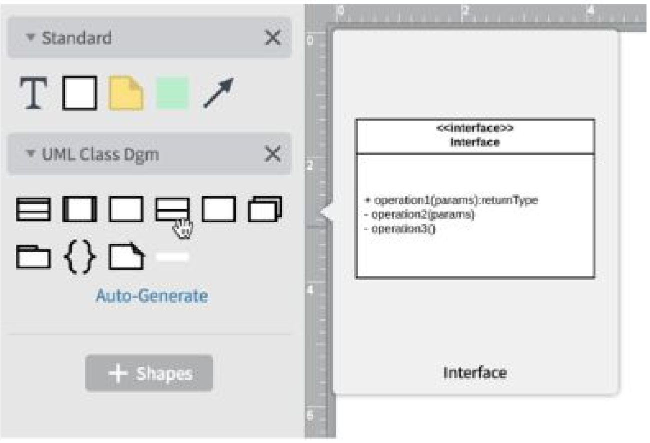
Indentation for composite attribute

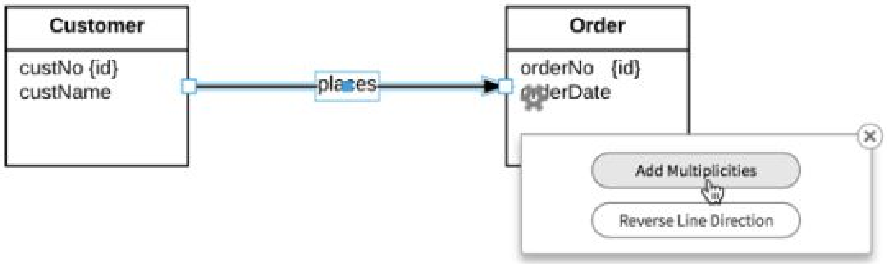
[n..m] - multivalued attribute

/ - calculated attribute

UML - LucidChart

UML Class Diagram - use Interface shape as no methods being added





Relationship lines - directed line, arrow head at M end, add Multiplicities

Summary of Terminologies at Different Levels

Conceptual	Logical	Physical
Entity	Relation	Table
Attribute	Attribute	Column
Instance	Tuple	Row
Identifier	Primary Key	Primary Key
Relationship	---	---
---	Foreign Key	Foreign Key

Properties of Relations

- Some properties to be considered:
  - Each relation has a unique name in the database.
  - Each row is unique - i.e. duplicate tuples are not allowed.
  - Each column has a (meaningful) name.
  - The order of attributes is immaterial.
  - The order of tuples is immaterial.
  - The entries are single-valued **(atomic)** - each cell contains a single entry.
    - **Multi-valued** and **composite attributes**???

Transforming ER diagrams into relations (mapping conceptual level to logical level)

- **The steps are:**
  - Map strong (regular) entities
  - Map weak entities
  - Map binary relationships
  - Map associative entities
  - Map unary relationships
  - Map ternary relationships
  - Map supertype/subtype relationships (is not part of this unit).

Map Regular Entities

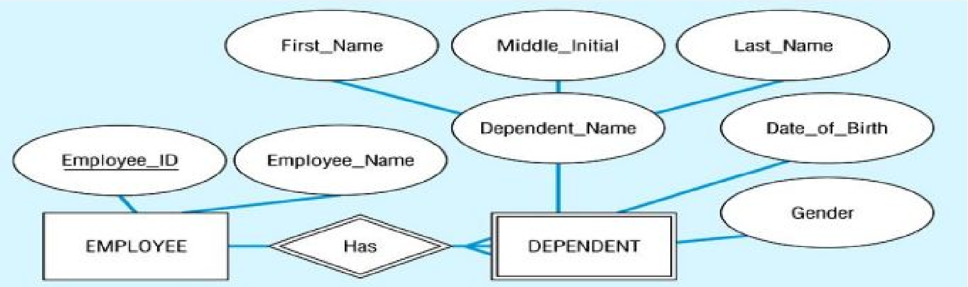
- **Composite Attributes**
  - When the regular entity type contains a **composite attribute**, only the simple component attributes of the composite attribute are included in the new relation.
  - Compared to composite attributes, simple attributes not only improve data accessibility but also help in maintaining data quality

Map Regular Entities

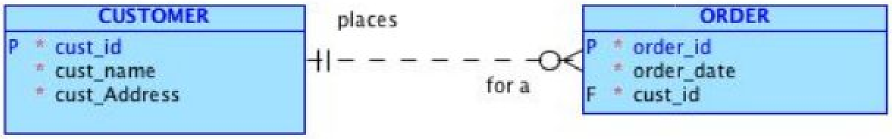
- **Multivalued Attribute**
  - When the regular entity type contains a **multivalued attribute,** two new relations are created.
  - The first relation contains all the attributes of the entity type except the multivalued attribute itself.
  - The second relation contains two attributes that form the PK. One of the attributes is the PK from the first relation, which becomes the FK in the second relation and the other is the multivalued attribute.
  - There can also be non key attributes in the second relation depending upon the data requirements.

Mapping a Weak Entity

- For each weak entity type, create a new relation and include all of the simple attributes as attributes of this relation. The PK of the identifying relation is also included as the FK in this new relation.



Map Binary Relationships (1:M)



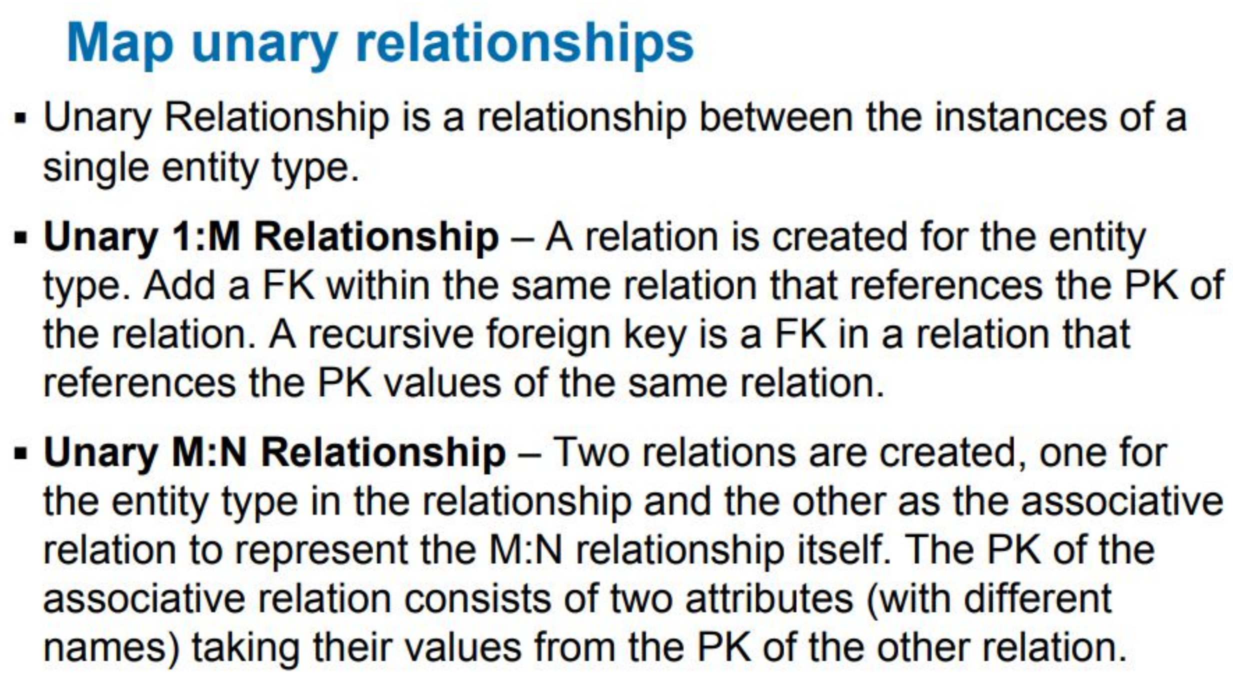
For each 1:M binary relationship, first create a relation for each of the two entity types participating in the relationship. Then include the PK attribute (or attributes) of the entity on the one-side of the relationship as the FK on the many-side of the relationship.

Map Binary Relationship (M:N)

- For a M:N binary relationship
  - First create a relation for each of the two entity types participating in the relationship.
  - Then create a new relation and include as foreign key attributes, the PK attribute (or attributes) for each of the two participating entity types. These attributes become the PK of the new relation.
  - If there are any nonkey attributes associated with the M:N relationship, they are also included in the new relation.

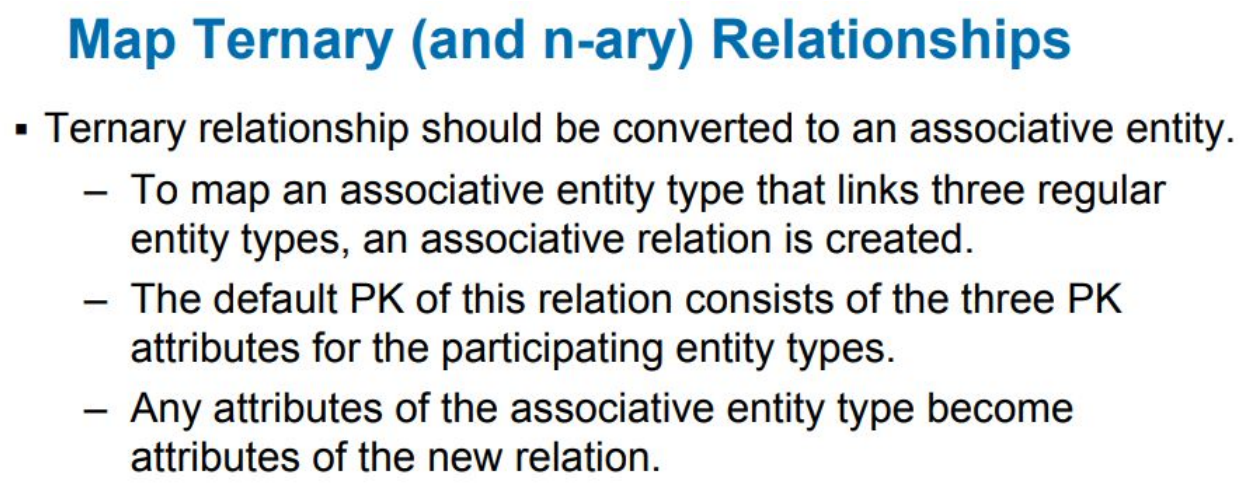
Map Binary Relationship (1:1)

- Create two relations, one for each of the participating entity types.
  - The primary key (PK) on the mandatory side of the relationship becomes the foreign key (FK) on the optional side of the relationship.
  - where both are optional place the FK on the side which causes the fewest nulls
  - Special case: *1:1 total* relationship (mandatory participation on both sides)
    - Consider consolidating the two entity types into one relation



Map unary relationships

- Unary Relationship is a relationship between the instances of a single entity type.
- **Unary 1:M Relationship** – A relation is created for the entity type. Add a FK within the same relation that references the PK of the relation. A recursive foreign key is a FK in a relation that references the PK values of the same relation.
- **Unary M:N Relationship** – Two relations are created, one for the entity type in the relationship and the other as the associative relation to represent the M:N relationship itself. The PK of the associative relation consists of two attributes (with different names) taking their values from the PK of the other relation.



Map Ternary (and n-ary) Relationships

- Ternary relationship should be converted to an associative entity.
  - To map an associative entity type that links three regular entity types, an associative relation is created.
  - The default PK of this relation consists of the three PK attributes for the participating entity types.
  - Any attributes of the associative entity type become attributes of the new relation.

EOF.