NATIONAL UNIVERSITY
COLLEGE OF COMPUTING AND
INFORMATION TECHNOLOGIES

VER 1.0

Course MANUAL

INFORMATION MANAGEMENT

This course provides students with the theoretical knowledge and practical skills in the utilization of databases and database management systems in the ICT applications. The logical design, physical design and implementation of relational databases are all covered in this course.

INFORMATION TECHNOLOGY

JULIE ANNE ANGELES-CRYSTAL
ROGER C. PRIMO JR.
MARK JAMES G. CAYABYAB
JOHN IVAN C. MAURAT

INFORMATION MANAGEMENT

Table of Contents

MODULE

#04

NATIONAL UNIVERSITY

MODULE GOALS

The Entity Relationship Model

Developing an ER Diagram

Advantages and Disadvantages of ER Modelling

FLEX Course Material

Entity Relationship Modeling

WEEK 6-7 MODULE

Education that works.

LESSON #

4.1

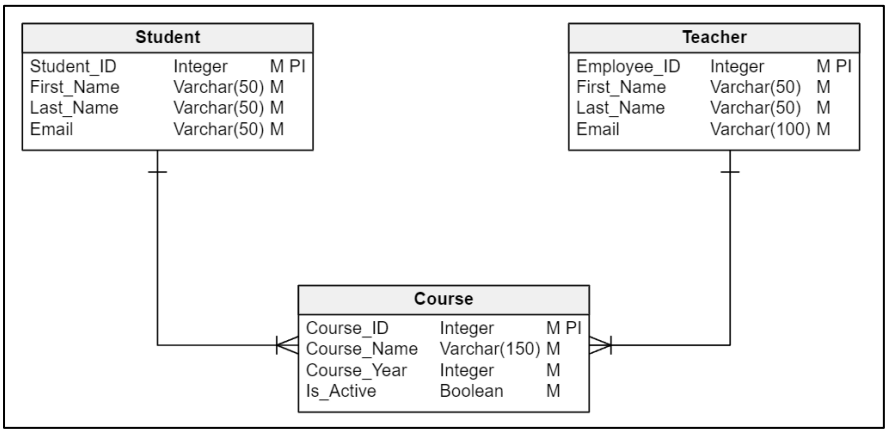
The Entity Relationship Model

Entity-Relationship Modeling (ER Modeling or ERD) is a technique for visualizing and documenting the relationships between different entities in a system.

It is widely used in database design and software engineering to represent the structure of a system and its data in a clear and concise way.

The model is expressed through an Entity-Relationship Diagram (ERD), which consists of entities, attributes, and relationships.

Entity Relationship Model



COMPONENTS OF ERD

Entities:

- Entities are objects or concepts in the real world that have data to be stored.
- In an ERD, entities are typically represented by rectangles. Each entity is labeled with a singular noun, and its instances (individual occurrences) are represented as rows in database tables.
- Example: In a university database, entities could include "Student," "Course," and "Instructor."

Attributes:

- Attributes are the properties or characteristics of entities. They describe the data we want to store about each entity.
- In an ERD, attributes are represented inside ellipses and connected to their respective entities.
- Example: Attributes for the "Student" entity might include "StudentID," "FirstName," "LastName," and "BirthDate."

COMPONENTS OF ERD (continued…)

Relationships:

- Relationships define how entities are connected or associated with each other. They illustrate the connections between different entities in the system.

- Relationships are typically represented by diamond shapes on an ERD. The lines connecting entities to diamonds show the nature and cardinality of the relationship.

- Example: A relationship between "Student" and "Course" entities might be labeled as "Enrolls In," indicating that a student enrolls in courses. The cardinality might specify that a student can enroll in multiple courses, and a course can have multiple enrolled students.

Cardinality and Modality:

- **Cardinality:** Describes the number of instances of one entity that can be associated with a single instance of another entity. Common notations include "1" (one), "M" (many), "0..1" (zero or one), "0..M" (zero to many).

COMPONENTS OF ERD (continued…)

Modality:

- Describes the minimum participation requirements of an entity in a relationship. Common notations include "Optional" or "Mandatory."

Primary Key:

- The primary key is a unique identifier for each instance of an entity. It helps uniquely identify and distinguish records in a database table.
- In an ERD, the primary key is usually underlined in an entity.

- **Foreign Key:**
  - A foreign key is a field in a table that is the primary key in another table. It establishes a link between two tables, typically representing a relationship.

Weak Entities:

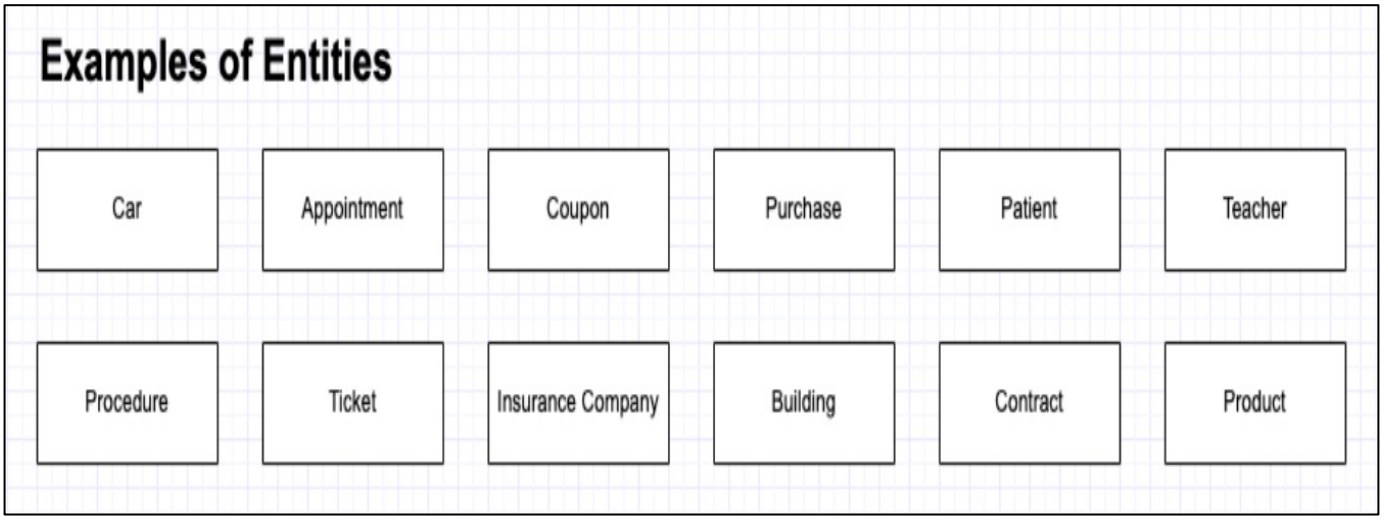
- Weak entities are entities that do not have a primary key attribute of their own. They depend on another entity (owner) for their existence.
- They are represented by a double-bordered rectangle in an ERD.

Entity Relationship Model

HOW TO READ ENTITY RELATIONSHIP DIAGRAM

What is an Entity?

- The entity of an ERD is simply a specific noun that carries data to describe it. In an ERD, you'll use rectangles to designate entities.
- Entities can also be described in types, sets, or categories.
- For example, a single customer would be an entity, while the broader noun "customers" would be an entity type.
- Entity sets are like an entity type, but also include an element of time, so an example would be "customers who came in last week."



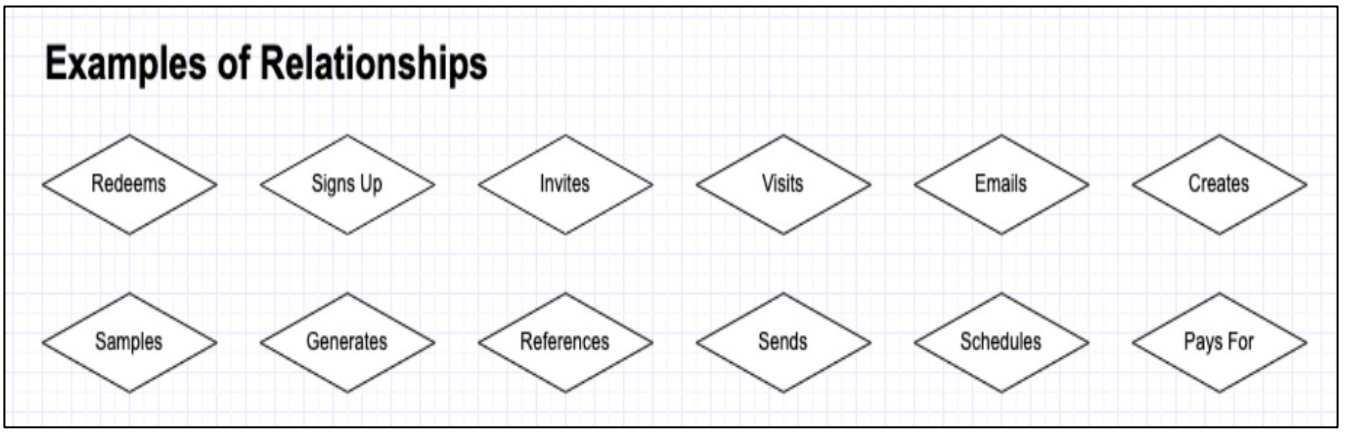
Entity Relationship Model

HOW TO READ ENTITY RELATIONSHIP DIAGRAM

What is a Relationship in ER Models?

- In an ERD or ER model, relationships are the verbs that describe how your entities interact with one another. Relationships are shown with a diamond.

- An example of a relationship within an ER model would be the interaction of two entities. Using our “customer” example from above, another entity could be “coupons.” The relationship between these two would be the verb, that the customer redeems a coupon.



Entity Relationship Model

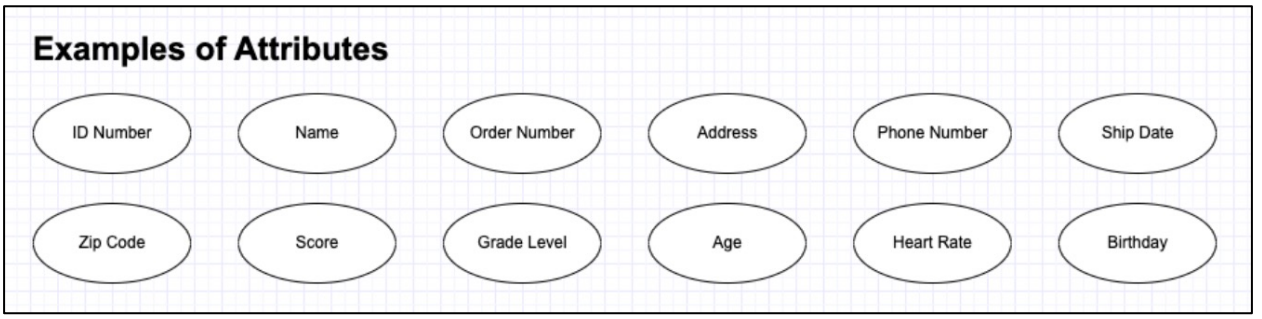
HOW TO READ ENTITY RELATIONSHIP DIAGRAM

What is an Attribute for an ERD?

- Attributes are the adjectives or pieces of data that you capture about an entity. When you make an entity relationship diagram of your own, you'll use an oval to designate these attributes.

- So if we've identified customers and coupons as entities for our diagram, attributes would be the data we collect around either the customers or the coupons.

- These attributes can be customer names, coupon codes, payment methods, the amount of money saved with the coupon, or even whether that customer chose paper or plastic at the checkout lane. Whatever data you're collecting can show up in your diagram as an attribute.

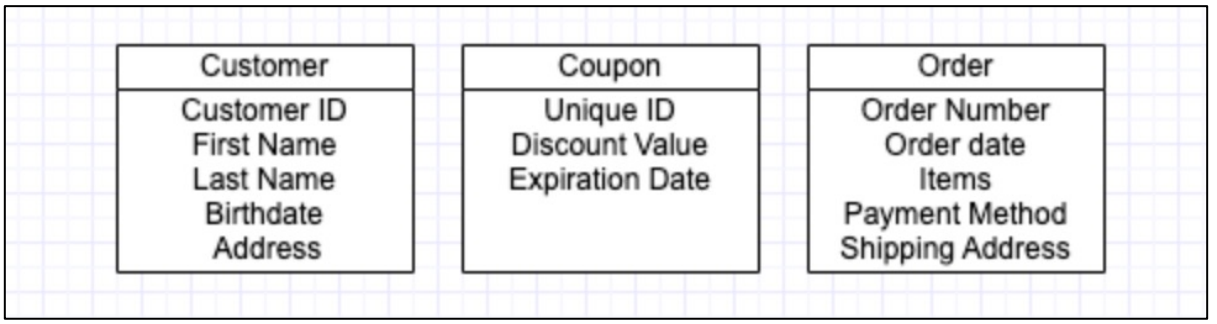


Entity Relationship Model

HOW TO READ ENTITY RELATIONSHIP DIAGRAM

Combining Entities and Attributes

- The above notations work great for a conceptual model of your architecture, but you can be more specific by using a different set of notations.
- Without complicating things too much, simply think of this step as collapsing your attributes under each entity.



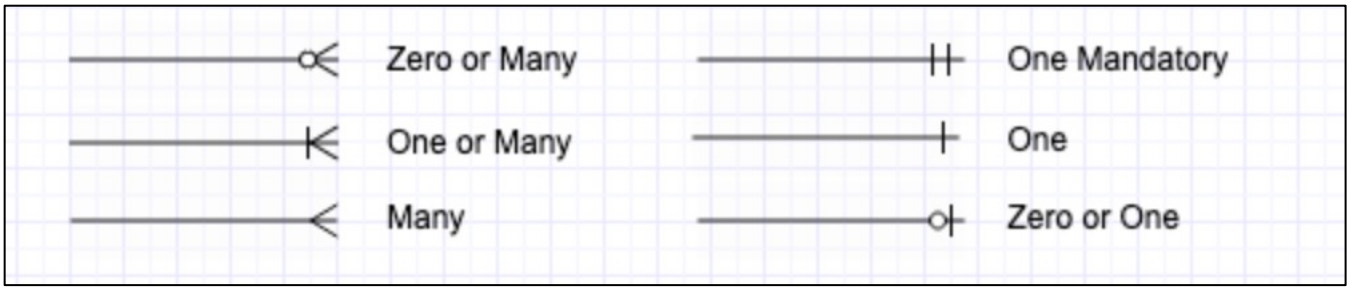
Using Lines in ERDs

- Lines in entity relationship diagrams are important because they further explain the relationships and attributes of the data you're describing. These line types are called the cardinality in your diagram. These lines tell you whether the connection between an entity and attribute can have zero, one, or many points of data.

CARDINAL RELATIONSHIP

- Zero or one
- One
- One and only one
- Zero or many
- Many
- One or many

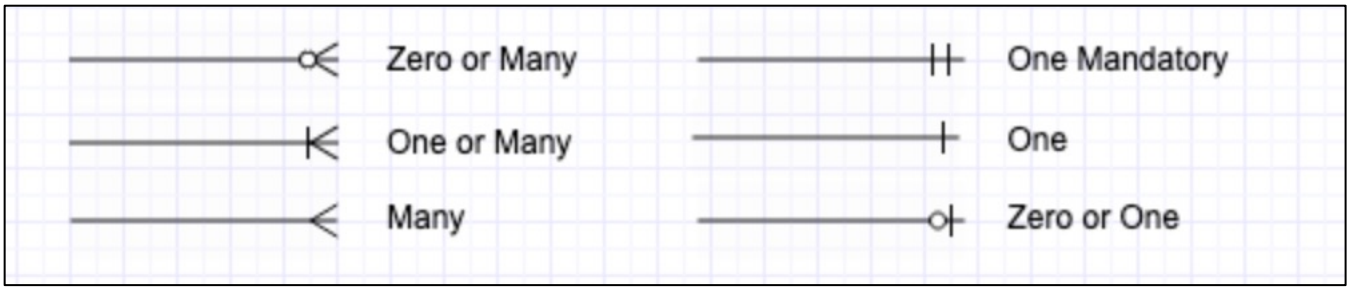
- For example, **one and only one** adult should have **zero or one** driver's license numbers. An adult may not have their license yet, so there may not be any data here (zero), but they also should never have more than one driver's license number.
- Likewise, the same license number should never be given to more than one adult. When it comes to phone numbers, many adults might have multiple numbers on record while some might not have any phone numbers recorded, so you'd use **zero or many** in that case.



CARDINAL RELATIONSHIP

- Zero or one
- One
- One and only one
- Zero or many
- Many
- One or many

- For example, **one and only one** adult should have **zero or one** driver's license numbers. An adult may not have their license yet, so there may not be any data here (zero), but they also should never have more than one driver's license number.
- Likewise, the same license number should never be given to more than one adult. When it comes to phone numbers, many adults might have multiple numbers on record while some might not have any phone numbers recorded, so you'd use **zero or many** in that case.



Entity Relationship Model

STEPS IN CREATING AN ERD:

Entity-Relationship Modeling provides a visual representation of the data structure and relationships in a system, helping in database design, communication among stakeholders, and understanding the overall structure of a system.

STEPS:

1. **Identify Entities:** Identify the main entities in the system.
2. **Identify Relationships:** Determine how entities are related to each other.
3. **Define Attributes:** Specify the attributes for each entity.
4. **Determine Primary Keys:** Identify the primary key for each entity.
5. **Draw the ERD:** Create the diagram, including entities, attributes, relationships, and cardinalities.

LESSON #

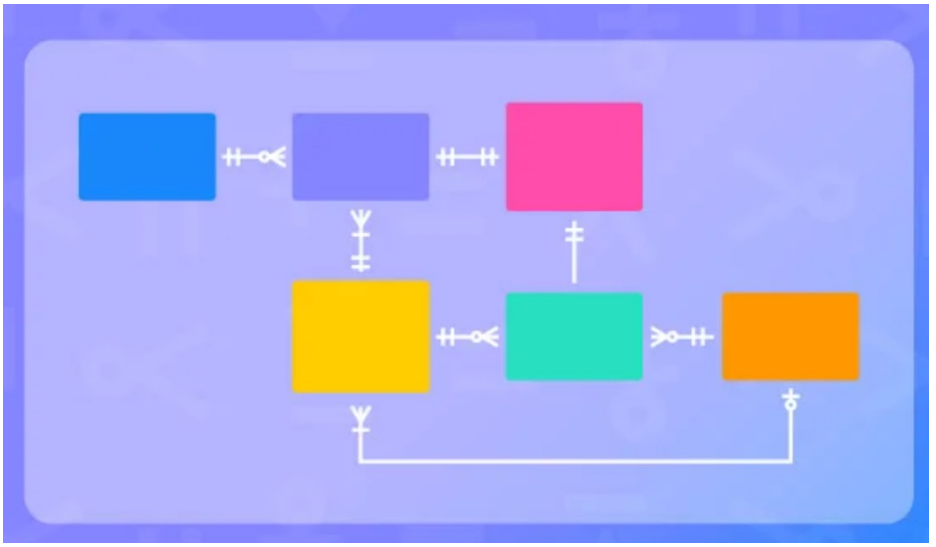
4.2

Developing an ER Diagram

Developing an Entity-Relationship Diagram (ERD) involves a systematic process of identifying entities, relationships, and attributes within a system.

An ERD, or entity relationship diagram, is a type of flowchart that helps you clearly visualize your database design by showing how the "entities" in the system relate to one another.

HOW TO DRAW ERDs



WHY DRAW AN ER DIAGRAM?

- ERDs, also called ER diagrams or ER models, are used to describe data and how pieces of data interact with one another.
- For this reason, ERDs are extremely important in database design and projects that require a clear structure of all data — think of it as the standardized way to draw a database diagram. By applying this standard, your team can easily understand the structure of a database or the information you collect within your system.

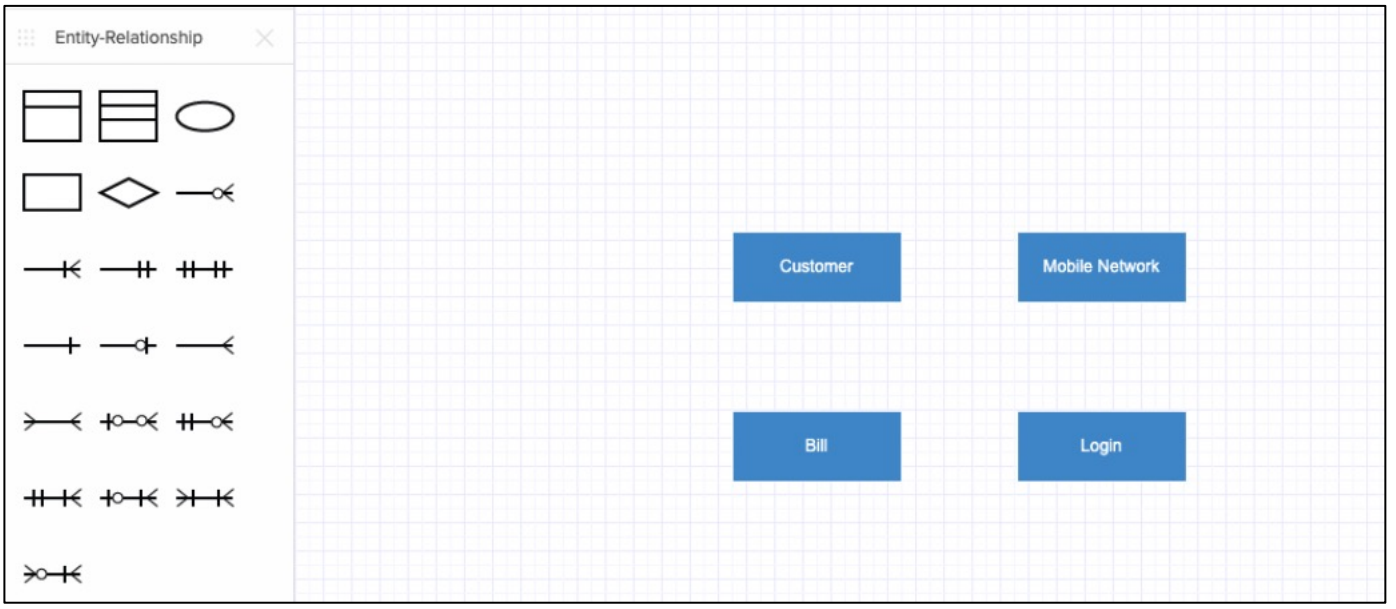
HOW TO DRAW ERDs

BASIC ORDER TO FOLLOW?

- There are a few basic steps to take to draw an ER diagram anywhere:

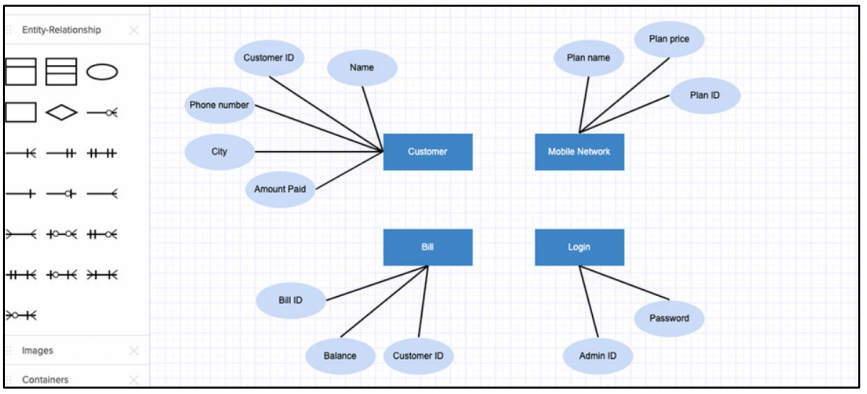
1. Determine the Entities in Your ERD:

   Start by identifying the “what”s in your system or architecture. Entities are represented with a rectangle, and you’ll want to give them plenty of room so that you can add to your diagram in the next steps.

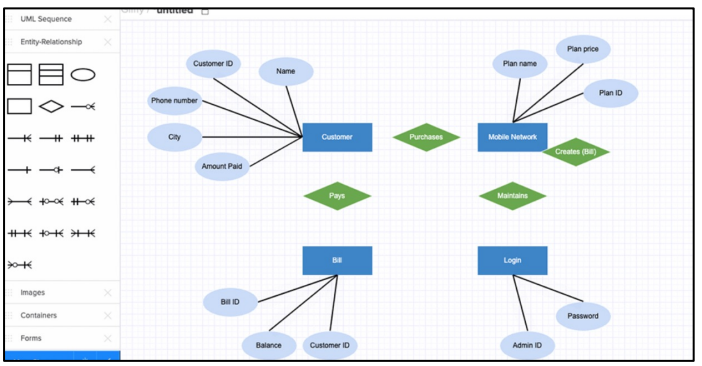


BASIC ORDER TO FOLLOW?

Add Attributes to Each Entity



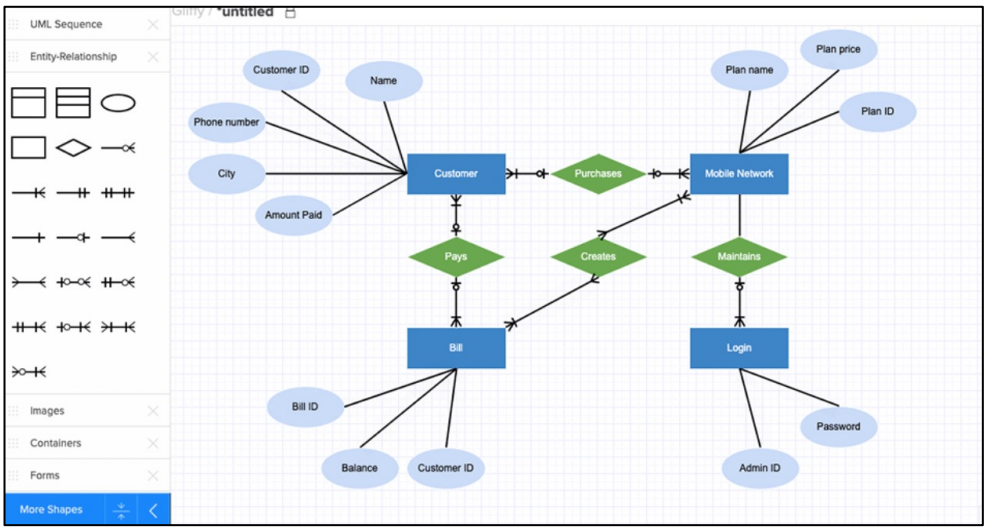
Next, consider the attributes that you need to describe each entity. These are drawn and labeled inside ovals. Connect these to the relevant entity and position your attributes to the outside of your diagram, which leaves room for relationships.



Now, think through the relationships or verbs taking place within your system. The easiest way to do this is to look at each entity and try to connect it to another by saying, “What does the ___ do with the ___.” The customer *purchases* the phone. The cell service *maintains* the phone. The cell service *creates* a bill. The customer *pays* the bill.

BASIC ORDER TO FOLLOW?

- Add Cardinality to Every Relationship in your ER Diagram



The final step for this simple ER diagram is to define the amount of data that will come from each entity. Cardinality is a simple notation that quickly tells your ERD reader whether there are zero, one, many, or some combination of those factors between each entity. Your customer can purchase *one or many* phones. The cell service maintains *many* phones. The customer pays *one* bill.

5. Finish and Save Your ERD

- This is just a high-level ER diagram, but it provides enough detail that you should now have a teammate or partner check your work — if you're working in Confluence, you can even invite them to a real-time collaboration session. One of the best ways to check your work is to simply have them try to read your diagram out loud. If they end up telling a different story than you intended, you need to do some tweaking.

EXERCISE:

LIBRRAY SYSTEM ERD

Exercise for creating an Entity-Relationship Diagram (ERD) for a simple library system.

The system involves entities such as "Book," "Author," "Publisher," and "Borrower," and relationships like "Author writes Book," "Book published by Publisher," and "Borrower borrows Book."

- Entities:

1. Book: [Attributes: ISBN (Primary Key), Title, Genre, PublishedDate]

2. Author:[Attributes: AuthorID (Primary Key), FirstName, LastName, BirthDate]

3. Publisher:[Attributes: PublisherID (Primary Key), PublisherName, Address, Phone]

4. Borrower:[Attributes: BorrowerID (Primary Key), FirstName, LastName, Email]

- Relationships:

1. Author writes Book:

- Many-to-Many relationship between "Author" and "Book" entities.
- Additional attribute: PublishedDate (when the book was published by the author).

2. Book published by Publisher:

- Many-to-One relationship between "Book" and "Publisher" entities.

3. Borrower borrows Book:

- Many-to-Many relationship between "Borrower" and "Book" entities.
- Additional attributes: BorrowDate, ReturnDate.

- **Instructions:**
  1. Create an ERD for the library system based on the provided entities and relationships.
  2. Clearly indicate primary keys, foreign keys, attributes, and relationships in your diagram.
  3. Consider cardinalities and modality for each relationship.
  4. Use standard ERD symbols for entities, relationships, and attributes.

EXERCISE:

LIBRRAY SYSTEM ERD

Tips:

- Use rectangles for entities, diamonds for relationships, ovals for attributes.
- Clearly label each entity, attribute, and relationship.
- Pay attention to cardinalities (one-to-one, one-to-many, many-to-many) and modality (optional or mandatory participation).
- Verify that each entity has a primary key.
- Ensure foreign keys are appropriately identified.

This ERD represents the relationships between entities in a library system.

The "AuthorBook" and "BorrowerBook" tables serve as junction tables to handle the many-to-many relationships.

LESSON #

4.3

Advantages and Disadvantages of ER Modeling

Entity-Relationship (ER) Modeling is a powerful technique for conceptualizing and representing the structure and relationships within a database system.

Like any modeling approach, it has its advantages and disadvantages.

Advantages & Disadvantages

ADVANTAGES:

1. Clarity and Simplicity:

- ER diagrams provide a clear and intuitive visualization of the data structure. The use of standardized symbols makes it easy to understand for both technical and non-technical stakeholders.

2. Communication:

- ER diagrams serve as a communication tool between various stakeholders, including developers, database administrators, and business analysts. It helps in conveying complex relationships in a simplified manner.

3. Database Design:

- ER modeling is a fundamental step in the process of database design. It helps in identifying entities, relationships, and attributes, laying the foundation for creating a normalized and efficient database schema.

4. Database Maintenance:

- ER diagrams assist in maintaining databases over time. Changes to the system or additions of new features can be visualized and planned more effectively, reducing the risk of errors during implementation.

5. Normalization:

- ER modeling encourages the normalization process, which involves organizing data in a way that reduces redundancy and improves data integrity.

6. Data Integrity:

- By clearly defining relationships between entities, ER modeling helps ensure data integrity. It enables the enforcement of constraints, such as primary and foreign keys, to maintain the accuracy and consistency of data.

7. Querying and Reporting:

- Understanding the relationships between entities facilitates the creation of effective queries and reports. ER modeling guides the identification of key elements for reporting purposes.

8. Tool Support:

- Various database design tools support ER modeling, providing a platform for creating, visualizing, and documenting database structures.

DISADVANTAGES:

1. Complexity in Large Systems:

- In large and complex systems, ER diagrams can become intricate and challenging to manage. Understanding and maintaining large diagrams may require additional effort.

2. Over-Simplification:

- In certain situations, ER diagrams might oversimplify the real-world complexities of a system. Some aspects of a business process may be challenging to represent accurately using only ER modeling.

3. Subjectivity:

- Interpreting and modeling relationships can be subjective. Different stakeholders may have different perspectives on how entities are related, leading to potential inconsistencies.

4. Implementation Details:

- ER modeling focuses on the conceptual level, and certain implementation details may be left out. While this abstraction is useful, it may lead to discrepancies during the actual implementation phase.

5. Evolutionary Changes:

- As the system evolves, making changes to the ER model might become complex. Evolving requirements and continuous changes may result in the need for frequent updates to the ER diagram.

6. Learning Curve:

- For individuals new to ER modeling, there might be a learning curve associated with understanding the symbols, notations, and conventions used in creating ER diagrams.

7. Inclusion of Business Rules:

- ER diagrams primarily focus on the structure of the database. Inclusion of detailed business rules and process flows may require supplementary documentation.

- In conclusion, while ER modeling is a valuable tool for designing databases, it's essential to recognize its limitations and carefully consider the specific needs of a given project. It is often used in conjunction with other modeling techniques to provide a comprehensive understanding of the system.