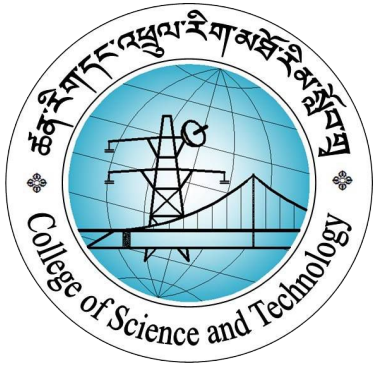


Royal University of Bhutan

DBS101 Database Systems Fundamentals

Lesson 3

Learning Outcomes

1. Explain the database design process.
2. Understand relational data models.
3. Explain entity relationship diagrams
4. Identify complex attributes.
5. Map cardinalities between entities
6. Determine primary keys in entity sets
7. Remove redundant attributes

Relational Model

Structure: The definition of the database's relations and their contents.

Integrity: Ensure the database's contents satisfy constraints.

Manipulation: Programming interface for accessing and modifying a database's contents.

Relational Model Structure

A relation is an unordered set that contain the relationship of attributes that represent entities.

A tuple is a set of attribute values (also known as its domain) in the relation.

- Values are (normally) atomic/scalar.
- The special value NULL is a member of every domain (if allowed).

Relational Database Terms

Relation - A Table

Tuple - Row of the table

Attribute - Column of the table.

Database schema - logical design

Database instance- snapshot of the database at a given instance of time.

StudentNo	CID	Name	Address
12345	12827827223	Sonam	Thimphu
24656	26735678768	Tandin Sonam	Bondey,Paro

Relational Model: Primary Keys

A relation's primary key uniquely identifies a single tuple.

Some DBMSs automatically create an internal primary key if a table does not define one.

StudentNo	CID	Name	Address
12345	12827827223	Sonam	Thimphu
24656	26735678768	Tandin Sonam	Bondey,Paro

Relational Model: Primary Keys

A relation's primary key uniquely identifies a single tuple.

Some DBMSs automatically create an internal primary key if a table does not define one.

StudentNo	CID	Name	Address
12345	12827827223	Sonam	Thimphu
24656	26735678768	Tandin Sonam	Bondey,Paro

Relational Model: Foreign Keys

A foreign key specifies that an attribute from one relation maps to a tuple in another relation.

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE
24656	26735678768	Tandin Sonam	Bondey,Paro	BEIT

Programme_ID	Programme_Name	Start_year
BESWE	Bachelors of Engineering in Software Engineering	2023
BEIT	Bachelors of Engineering in Information Technology	2011

Relational Model

Student(StudentNo,CID,Name,Address,Programme_ID)

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE
24656	26735678768	Tandin Sonam	Bondey,Paro	BEIT

StudentProgramme (Programme_ID,Student_ID)

StudentNo	**Programme_ID**
12345	**BESWE**
24656	**BEIT**

Programme(Programme_ID,Programme_Name ,Start_year)

Programme_ID	Programme_Name	Start_year
BESWE	Bachelors of Engineering in Software Engineering	2023
BEIT	Bachelors of Engineering in Information Technology	2011

How can i know the name of the department for each student?

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE
24656	26735678768	Tandin Sonam	Bondey,Paro	BEIT

Programme_ID	Programme_Name	Start_year	Dept_ID
BESWE	Bachelors of Engineering in Software Engineering	2023	SWE
BEIT	Bachelors of Engineering in Information Technology	2011	IT

Dept_ID	Dept_Name	HOD
SWE	Software Engineering Department	Douglas Sim
IT	Information Technology Department	Tandin Wangchuk

How can i know the name of the department for each student?

StudentNo	Programme_ID	Dept_ID
12345	BESWE	SWE
24656	BEIT	IT

Student department relation

Relational Model Keys

Superkey: a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation.

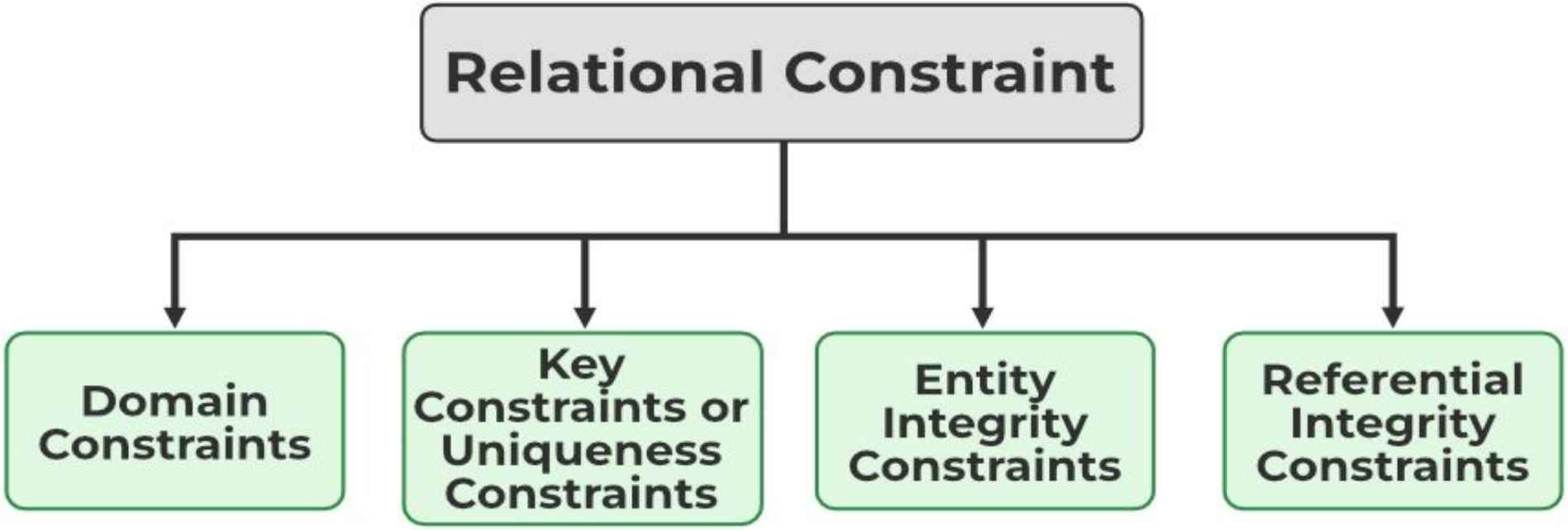
Example: dep_id or dept_id & dept_name

Candidate keys: a attribute or a combination of attributes that uniquely identifies each tuple in the relation.

Example: CID in student table, student_name and student_id

Primary Key: a candidate key that is chosen by the database designer as the principal means of identifying tuples within a relation.

Relational Model Constraints



Schema Diagrams

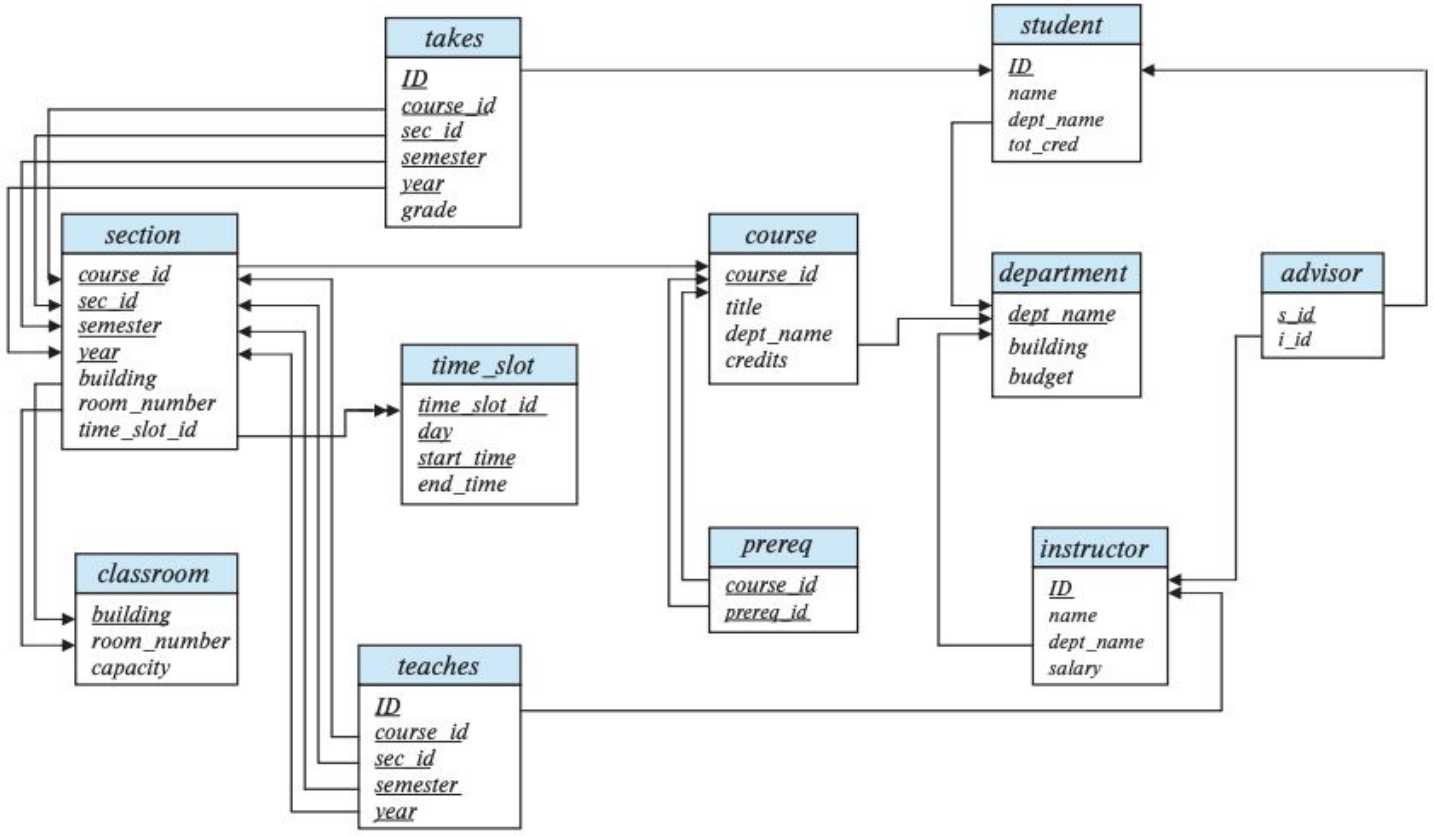


Figure 2.9 Schema diagram for the university database.

Relational Query Languages

A query language is a language in which a user requests information from the database.

Categories of query languages:

1. **Imperative query language**: user gives specific instructions.
2. **Functional query language**: input is given as evaluation of functions that may operate on data in the database
3. **Declarative query language**: the user describes the desired information without giving a specific sequence of steps or function calls for obtaining that information;

Two types of DML

Procedural

- The query specifies the (high-level) strategy to find the desired result based on sets / bags.

Non-procedural(Declarative)

- The query specifies only what data is wanted and not how to find it.

Relational Algebra

Relational Calculus

5 minutes Break !!!!

Relational Algebra

Relational Algebra is a procedural query language.

Relational algebra provides a theoretical foundation for relational databases and SQL.

Operationals

σ Select	∩ Intersection
π Projection	− Difference
∪ Union	× Product
	⋈ Join

Relational Algebra: SELECT

Choose a subset of the tuples from a relation that satisfies a selection predicate.

- Predicate acts as a filter to retain only tuples that fulfill its qualifying requirement.
- Can combine multiple predicates using conjunctions / disjunctions.
  Syntax: $\sigma_{predicate}(R)$

Relational Algebra: SELECT

Example

$$\sigma_{\text{studentNo} = "12345"}(\textbf{Student})$$

```
Result = SQL Query( SELECT * FROM Student
          WHERE StudentNo ="12345")
```

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE

Relational Algebra: PROJECTION

Generate a relation with tuples that contains only the specified attributes.

→ Rearrange attributes' ordering.
→ Remove unwanted attributes.
→ Manipulate values to create derived attributes.

Syntax: $\Pi_{A1,A2,\ldots,An}(R)$

Relational Algebra: PROJECTION

Example: $\Pi_{\text{CID-12800000000,Name}}(\text{StudentNo} = \text{'12345'})$

```
Result = SQL Query( SELECT CID-12800000000, Name
                    FROM student WHERE StudentNo = '12345')
```

CID	Name
27827223	Sonam

Relational Algebra: UNION

Generate a relation that contains all tuples that appear in either only one or both input relations.

Syntax: (R ∪ S)

Example: Student1 ∪ student2

```
SQL Query( SELECT * FROM Student1 ) UNION
          (SELECT * FROM Student2)
```

Relational Algebra: UNION

Student 1

StudentNo	CID	Name	Address	Programme_ID
12346	18978967854	Tandin	Chukha	BECivil

Student 2

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE
12346	18978967854	Tandin	Chukha	BECivil

Relational Algebra: INTERSECTION

Generate a relation that contains only the tuples that appear in both of the input relations.

Syntax: (R n S) Student1 n Student2

```
Result = SQL Query( SELECT * FROM Student1 )  INTERSECT
          (SELECT * FROM Student2)
```

Relational Algebra: INTERSECTION

Student 1

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE
12346	18978967854	Tandin	Chukha	BECivil

Student 2

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE

Relational Algebra: DIFFERENCE

Generate a relation that contains only the tuples that appear in the first and not the second of the input relations.

Syntax: (R - S)

```
Result = SQL Query( SELECT * FROM Student1 ) EXCEPT
         (SELECT * FROM Student2)
```

Relational Algebra: DIFFERENCE

Student 1

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE
12346	18978967854	Tandin	Chukha	BECivil

Student 2

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE

StudentNo	CID	Name	Address	Programme_ID
12346	18978967854	Tandin	Chukha	BECivil

Relational Algebra: Product

Generate a relation that contains all possible combinations of tuples from the input relations.

Syntax: (R X S)

```
  Result = SQL Query( SELECT * FROM Student1 CROSS JOIN
Student2
            SELECT * FROM Student1,Student2;)
```

Relational Algebra: Product

Student 1

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE

Student 2

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE

Relational Algebra: Product

Generate a relation that contains all possible combinations of tuples from the input relations.

Syntax: (R X S)

```
   Result = SQL Query( SELECT * FROM Student1 CROSS JOIN Student2
                SELECT * FROM Student1,Student2;)
```

StudentNo	CID	Name	Address	Programme_ID
12345	12827827223	Sonam	Thimphu	BESWE
12345	12827827223	Sonam	Thimphu	BESWE

Another Example

Relational Algebra: JOIN

Generate a relation that contains all tuples that are a combination of two tuples (one from each input relation) with a common value(s) for one or more attributes.

Syntax: (R ⋈ S)

There are any types of joins namely;

1. Inner Join
2. Left(Outer Join)
3. Right(Outer Join)
4. Full outer join
5. Self Join

Example

Relational Algebra: EXTRA OPERATORS

Rename (ρ)
Assignment (R←S)
Duplicate Elimination (δ)
Aggregation (γ)
Sorting (τ)
Division (R÷S)

Relational Algebra NOTES

Relational algebra defines an ordering of the high level steps of how to compute a query.
- Example: σb_id=102(R⋈S) vs. (R⋈(σb_id=102(S))

A better approach is to state the high-level answer that you want the DBMS to compute.
 - Example: Retrieve the joined tuples from R and S where b_id equals 102.

5 minutes Break !!!!

Entity Relationship Model

The entity-relationship (E-R) data model was developed to facilitate database design by allowing specification of an enterprise schema that represents the overall logical structure of a database.Builds the foundation of the logical data models.

Components of an Entity Relationship Diagram(ERD):

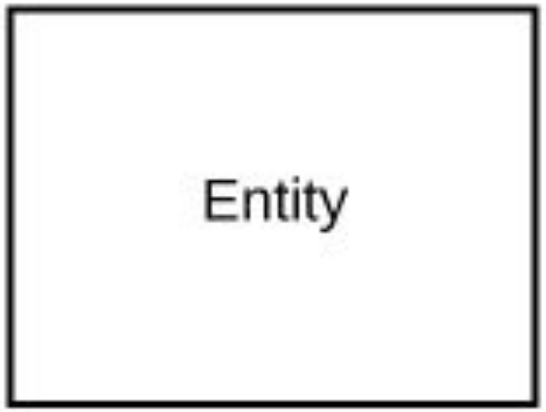
1. Entity: “thing” or “object” in the real world that is distinguishable from all other objects.
2. Attributes: Attributes are descriptive properties possessed by each entity.
3. Relationship: A relationship is an association among several entities.

Entity Relationship Model

Components of an ERD:

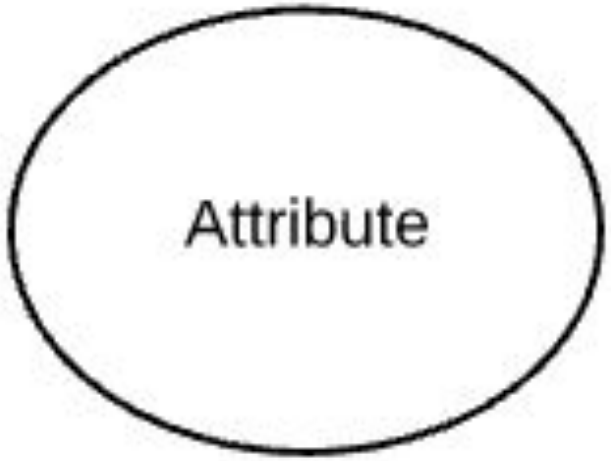
1. Entity: “thing” or “object” in the real world that is distinguishable from all other objects

.



Entity Relationship Model

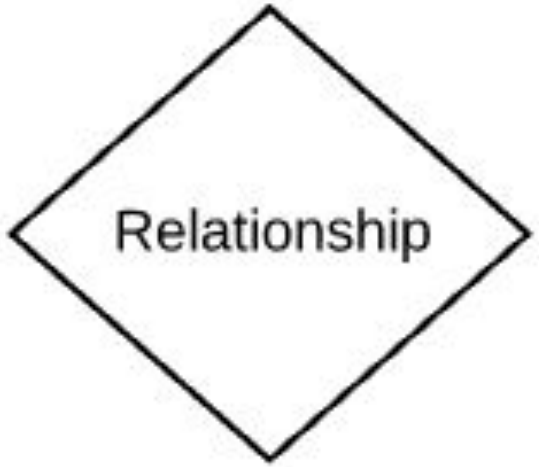
2. Attributes: are descriptive properties possessed by each entity.



Entity Relationship Model

3. Relationship: is an association among several entities.

Relationships are usually verbs such as assign, associate, or track and provide useful information that could not be discerned with just the entity types.

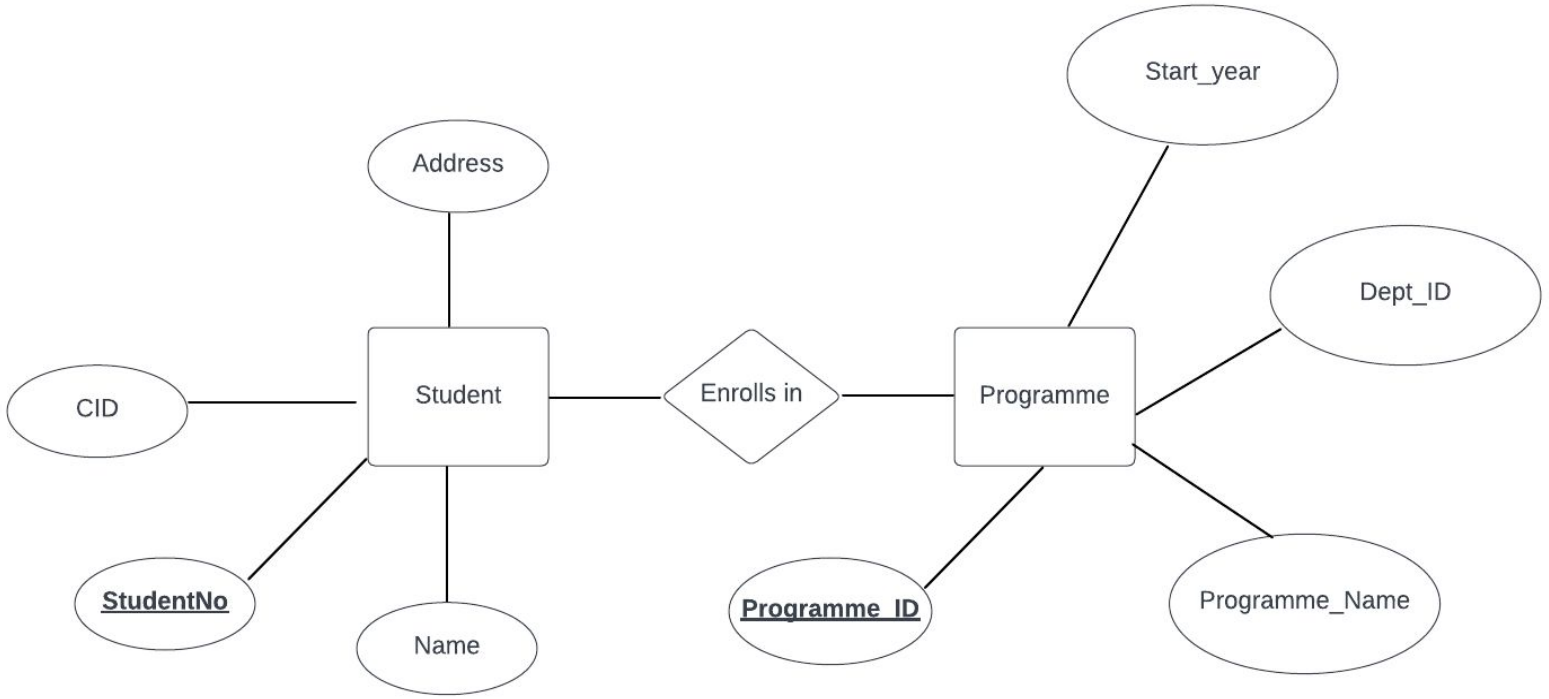


Entity Relationship Model

Simple ERD

Entity Relationship Model

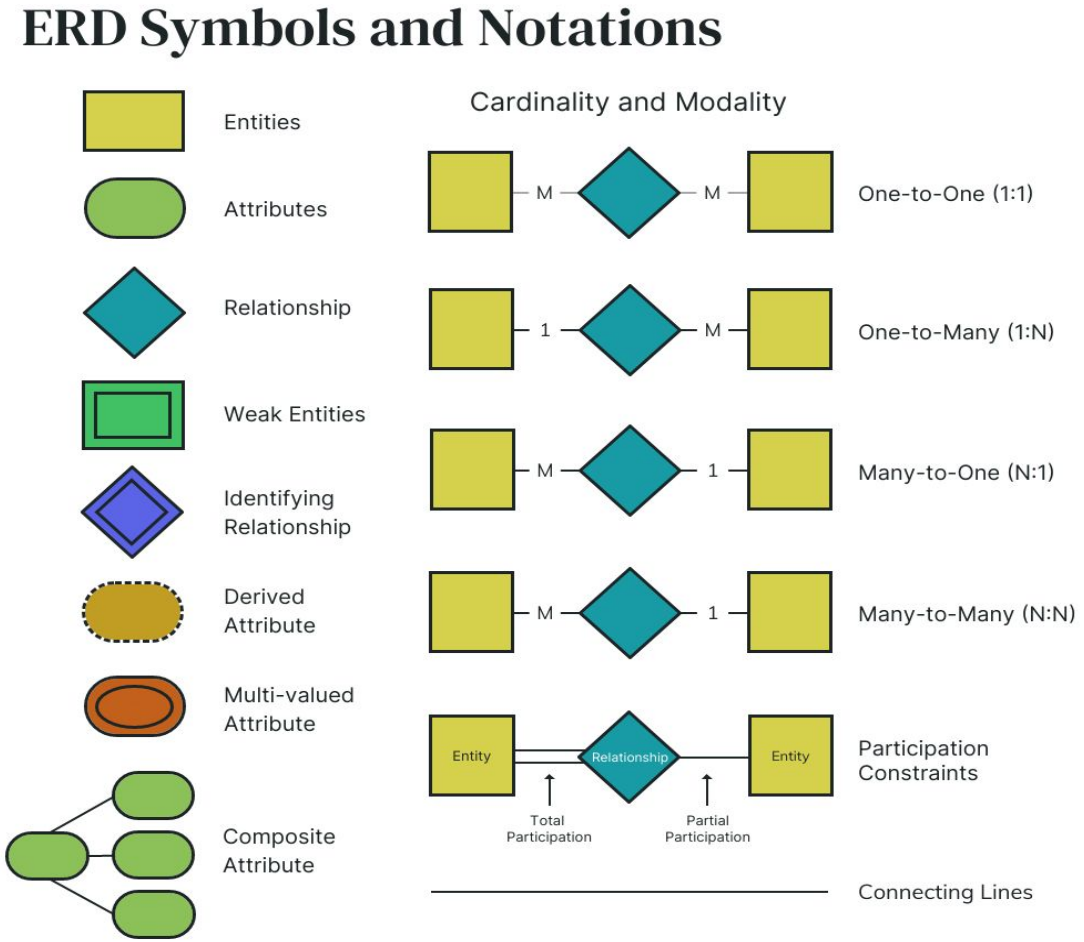
Simple ERD : Adding Attributes



Entity Relationship Model

Other notations:

ERD notations



Session Will Be Continued Tomorrow:

Homework; Read Chapter 6 Database Design Using E-R Model, topics 6.1 to 6.5

References

Entity-Relationship Diagram Symbols and notation. (n.d.). Lucidchart. https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning

GfG. (2023, November 6). *Constraints on relational database model*. GeeksforGeeks. https://www.geeksforgeeks.org/constraints-on-relational-database-model/