Data Management and Database Design

Week #2

Northeastern University

What is a Database?

It’s a shared, integrated computer structure that stores –

- End-user data
- Metadata
  - Description of data characteristics and relationships
  - Stores information such as –
    - Name of each data element
    - Type of values (numeric, dates, or text)
    - whether or not the data element can be left empty

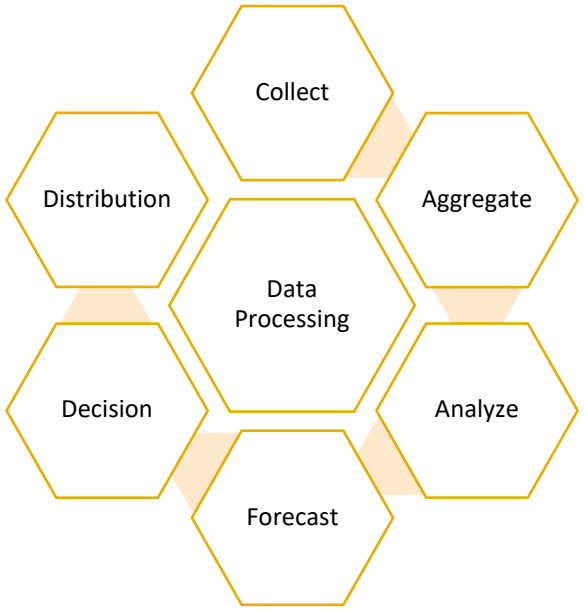
Database is also described as –

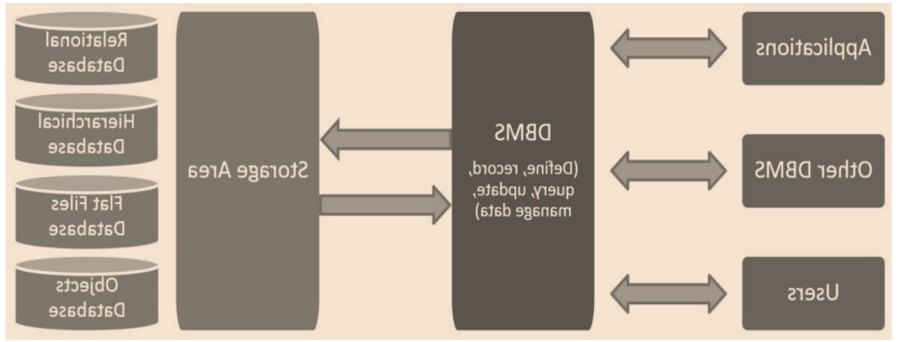
"*collection of self-describing collection of integrated records*"

Why Data Management?

- Let's discuss Problems with File based data management

- Data must be properly *managed for* –
  - Quality
  - Redundancy
  - Integrity
  - Security
  - Consistency





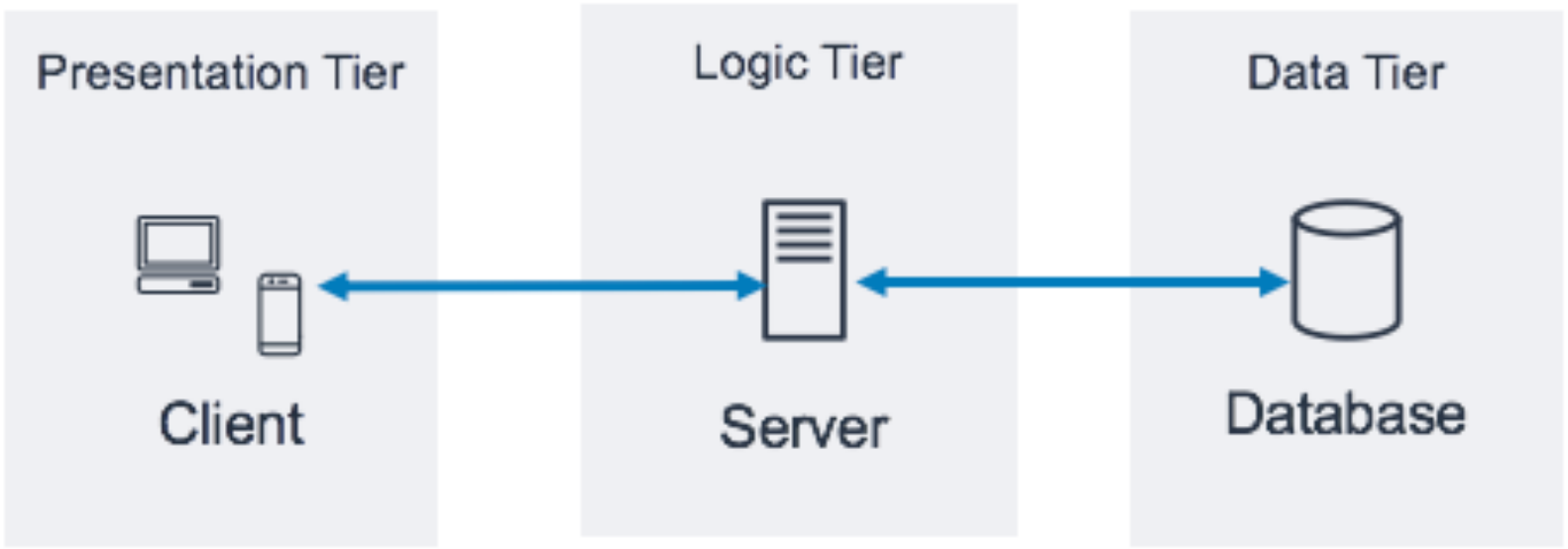
Types of Databases

- **Number of users**
  - *single-user*
  - *multiuser*
- **Types**
  - *Relational*
  - *Non-Relational*

- **Centralized**
- **Distributed**

- **OLTP**
- **Datawarehouse**

Client Server Architecture



Relational Databases

- Uses structured query language (SQL) for defining and manipulating data
- Since the mid-1980s, SQL has been a standard for querying and managing RDBMS data sets.
- Popular RDBMS
  - Oracle Server
  - MySQL
  - MS SQL Server
  - DB2
  - PostgreSQL
  - Aurora DB - RDS

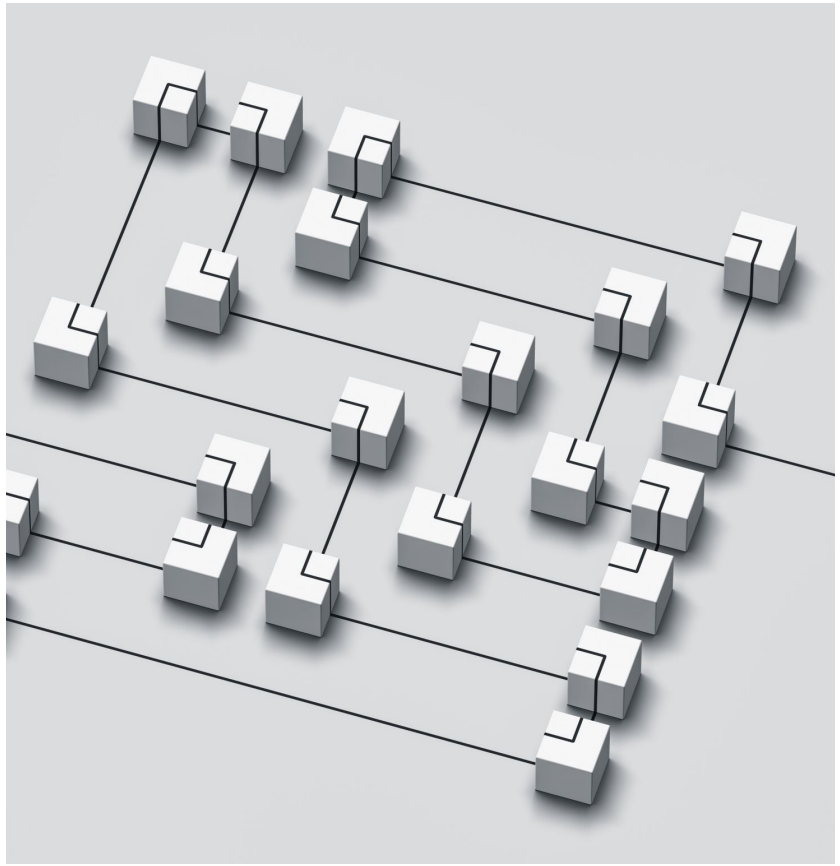
Relational Databases

- Data organization in terms of rows and columns

- Row must be unique
- Column names must be unique

- SQL requires predefined schemas to determine the structure of your data

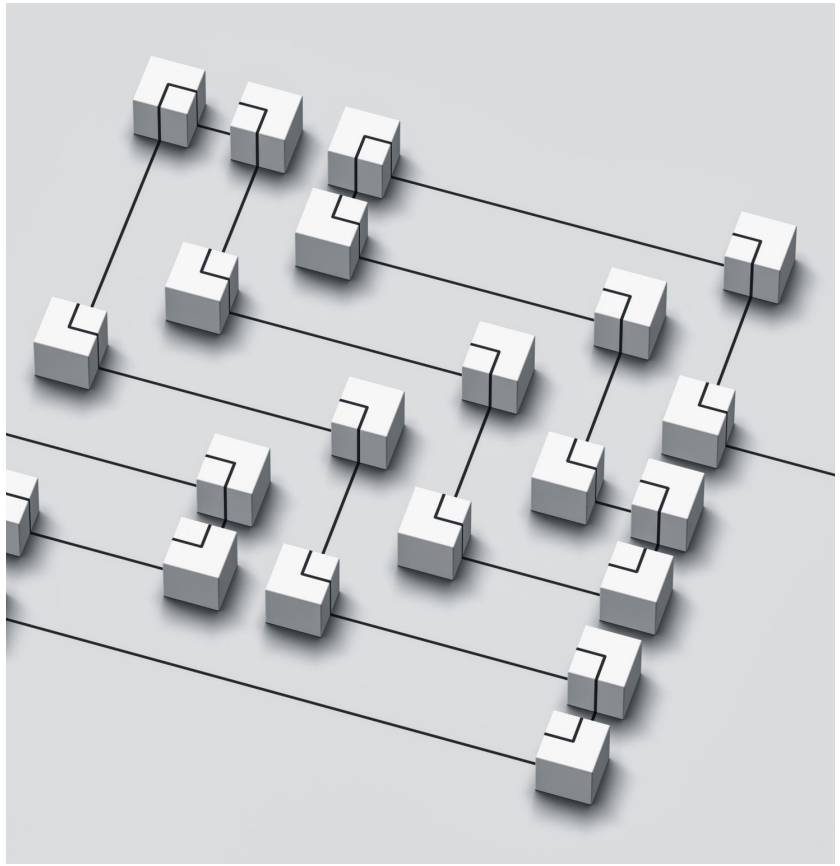
- A Relationship can be specified at any time using any column name



Relational Databases Advantages

Advantages

- Rely on relational **tables**
- Utilize defined data schema
- Reduce redundancy through **normalization**
- Engineered for data **integrity**
- Rely on a simple, **standardized query language**

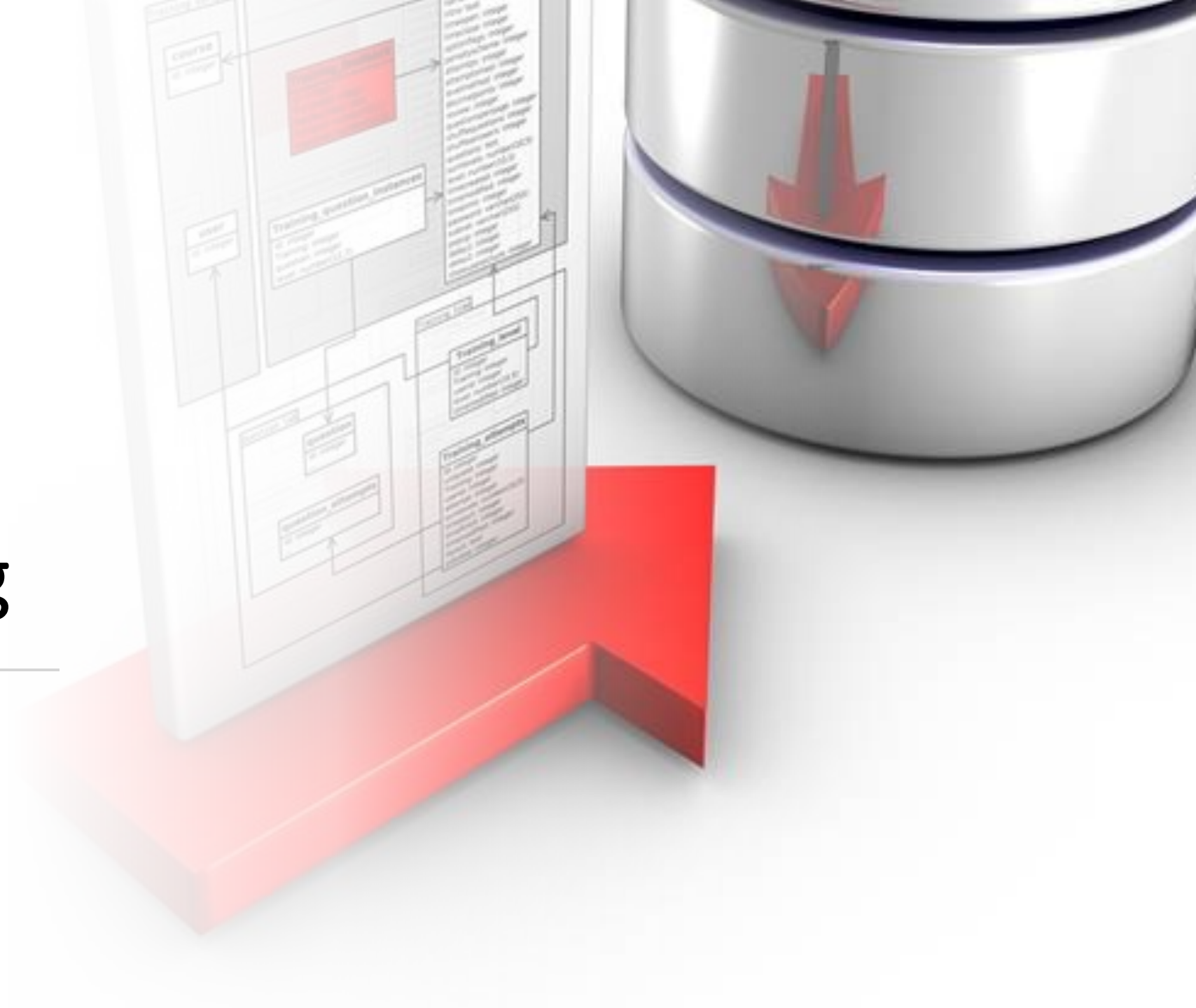


This week topics

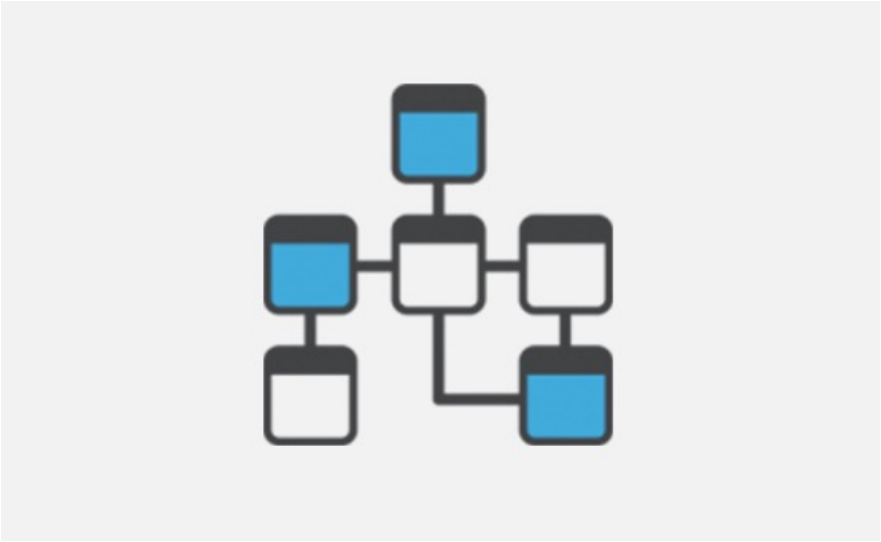
- Data modeling
  - Relational Data models

- database management system **rules**
- Entity
- Attribute
- Domain
- Relationships
- Relational and Non-Relational databases
  - Why we need
  - When to use

Data modeling



What is Data modeling?



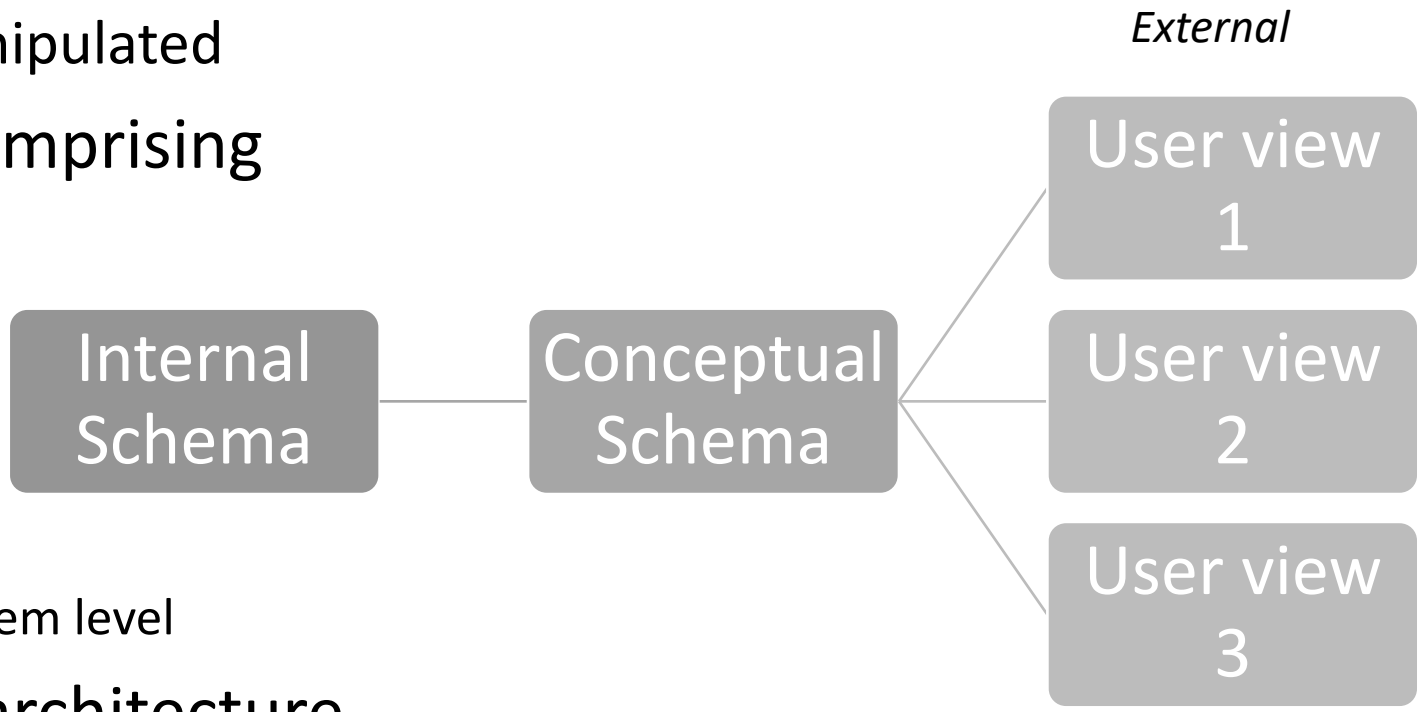
- **Graphical representation of –**
  - Data structures
  - Characteristics
  - Relations
  - Constraints
- **It is an Iterative process**
- **A perfect model meets all business requirements**

Why Data modeling?

- Data modeling is an abstraction process
- Organize
- Improve Performance

3 Level ANSI-SPARC Architecture

- Abstraction
  - How data is stored and manipulated
- Three-level architecture comprising
  - External schema
    - User level access
  - Conceptual schema
    - Data and relationships
  - Internal schema
    - Database and operating system level
- Major objective of 3 level architecture
  - Data independence (Logical and Physical)



Schema and its mappings

- At the highest (External) level of abstraction
  - we have multiple external schemas (also called subschemas)
  - Each external schema has different views of the data.
- At the conceptual level
  - Describes all the entities, attributes
  - Describes relationships between entities and integrity constraints
- At the lowest level (Internal) of abstraction
  - Definitions of storage

How is Data modeling performed?

- **Bottom Up**
  - Redesign of existing application
- **Top Down**
  - New applications
- **Hybrid**
  - mix of both Bottom up and Top Down
  - Example: Existing system enrichment with new features

Data Model

Artifacts of Data Model –

- Data structure that stores data
- Rules to maintain data integrity
- Data manipulation methods

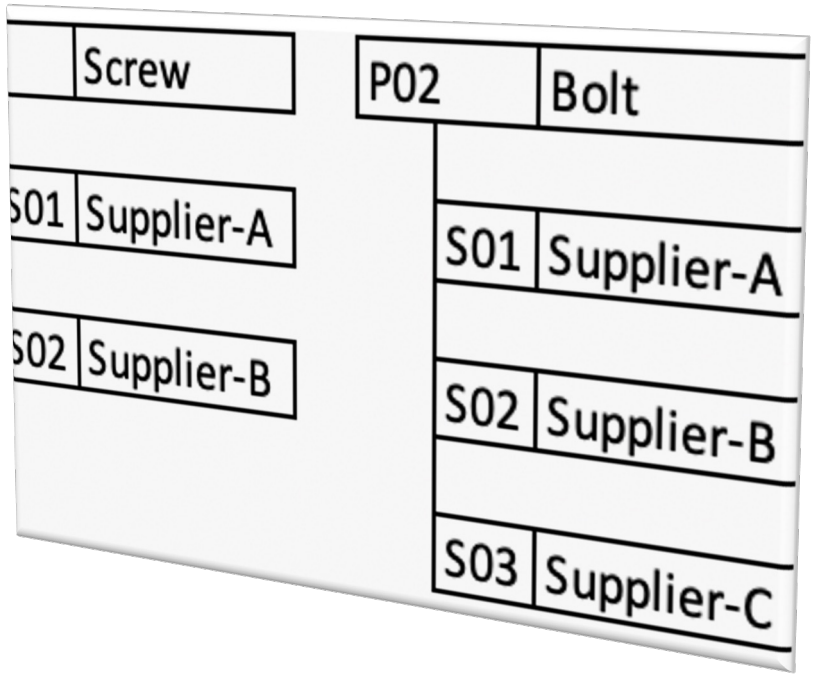
Types of Database models

- Hierarchical
- Network
- Relational

Hierarchical Model

- Developed by IBM
- Tree like structure with Single root
- Parent child relationship
  - Parent may have one or more child units
  - Child is restricted to have only one Parent

- Drawback
  - Cannot represent many to many relationship



Network Model

- Improvement over Hierarchical model
  - Multiple parent-child relationships
  - Multiple access paths
  - Most widely used before Relational was introduced
- Drawback
  - Transaction management via pointers and tracing

Relational Model

- The relational model was first proposed by E. F. Codd
- RDBMS can hide the complexities of the relational model from the user
- Think of a relation as a matrix composed of intersecting rows and columns
  - Table
- Main Objectives –
  - High degree of Independence
    - Application programs must not be affected by changes to file organizations, record orderings
  - Normalization
    - How to Normalize is discussed in later sessions
  - Data Consistency
  - Avoid Redundancy
  - Enable self oriented data manipulation language

Relational Model

- IBM's San José Research Laboratory in California –
  - First company to develop prototype relational DBMS System R
- System R project led to –
  - Development of a structured query language called SQL

- Production of various commercial relational DBMS products – early 1980s
  - DB2 from IBM
  - Oracle from Oracle Corporation.

Relational Model

Supplier

Supplier Code	Supplier Name	Supplier City
S01	Supplier-A	Roxbury
S02	Supplier-B	Lynn
S03	Supplier-C	Fraingham
S04	Supplier-C	Andavor

Part

Part Code	Part Name
P01	Screw
P02	Bolt
P03	Spacers
P04	Cable Ties

Shipment

Supplier Code	Part Code	Qty
S01	P01	150
S01	P02	200
S03	PO4	125
S03	PO3	300

Database Design

- **Requirements Formulation and Analysis**
  - Collection and documentation of requirement
  - Analyze requirements

- **Conceptual Design**
  - Identify Entities and relationships

- **Data Modeling**
  - First Level – ER Diagrams
  - Second Level – Normalization

- **Design Implementation**

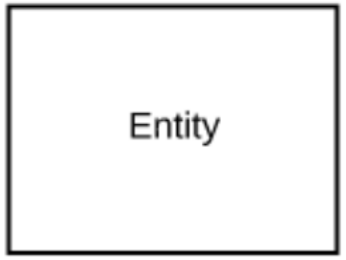
ER Diagram components

- **Entity**
  - Entity is a “Thing” which can be distinctly identified
  - Is an Object which has characteristics
    - Example: Supplier, Company, Product, Customer, Student, User, Sales
  - Entity type(category) is a collection of similar Entities
    - A Name that we give to an Entity
  - Entity Set is a collection of entities of an entity type (All entities of an entity type)
    - All rows/records of a Entity/Table

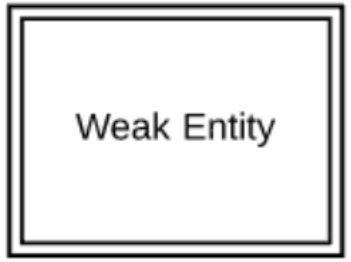
- **Attribute**
  - Set of Attributes define Entity / *Data element(s) that describes an Entity*
  - ***Domain*** of attribute is of permitted values
    - Phone number, SSN number, Date of Birth, Zip code

Entity Symbols….

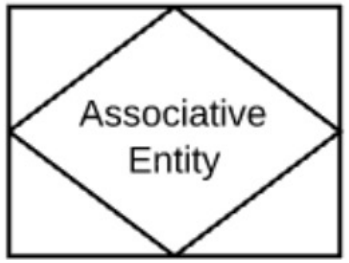
- Entity types
  - Strong Entity
    - Sometimes called as parent entities or Kernels
    - Contains primary keys
    - Building blocks of Database



  - Weak Entity / Dependent Entity
    - Depends on some other entity type
    - primary keys (combination with foreign key)

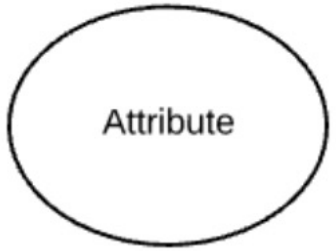


  - Associative Entity
    - Bridge tables
    - Created to resolve linkage between many to many relation ships

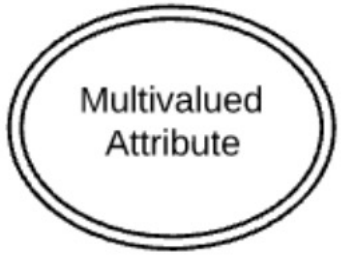


Attribute Symbols….

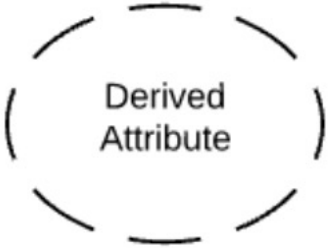
- Types of Attributes
  - Simple Attribute
    - Characteristics of an entity and cannot be broken down further
      - Example: Product Name



  - Multivalued attribute
    - Attributes which are capable of taking more than one value

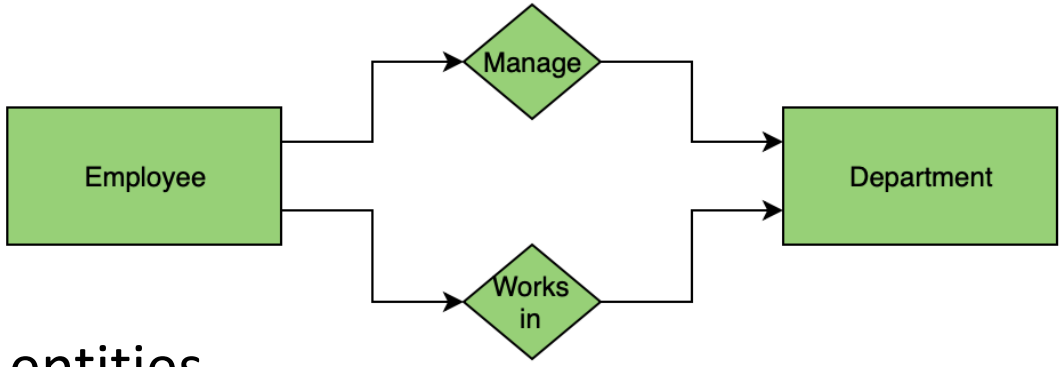


  - Derived Attributes
    - Calculated fields
    - These doesn’t exists in real
    - Logical representation

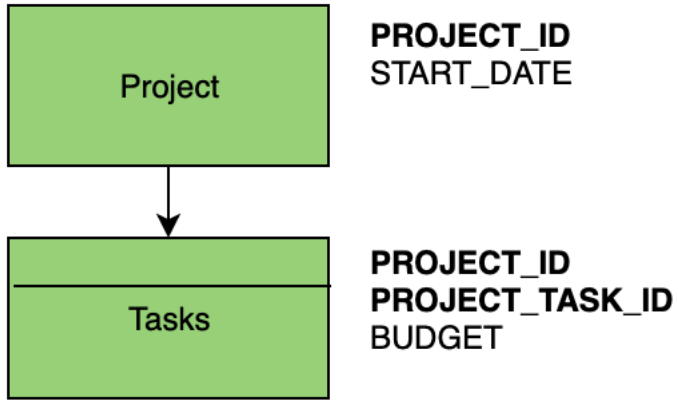


ER Diagram Relations

- Relationship
  - Association between Entities
  - Relates two entities
  - Several relationships may exist between same entities

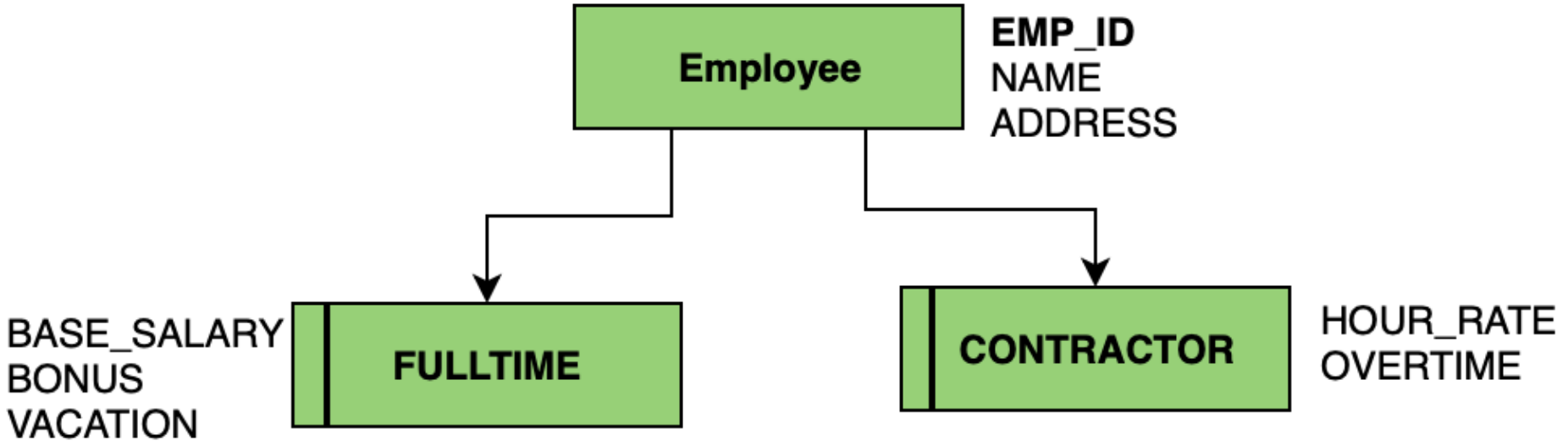


- Dependent Entity
  - An Entity whose existence depends on existence of another entity
  - Example
    - A project comprises of multiple tasks and task cannot be present with project



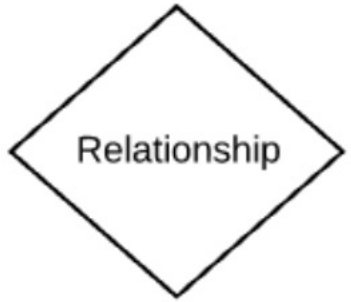
Subsets and Supersets

- A **subset** is derived from parent entities called **supersets**
- Common attributes among subsets are "**attributes of superset**"
- Identifier of a subset is always superset Identifier



Relationship symbols….

- Relationship symbols and its usage
  - Relationship
    - These are associations/connections between strong entities



  - Weak Relationship
    - These are connections between weak entities

Degree of Relationship

1 :1

One to One

1 : N

One to Many

N : N

Many to Many

Degree of Relationship is also called as **CARDINALITY**

One to One Relationship

- For one occurrence of first entity –
  - There can be at most one related occurrence of second entity and vice-versa

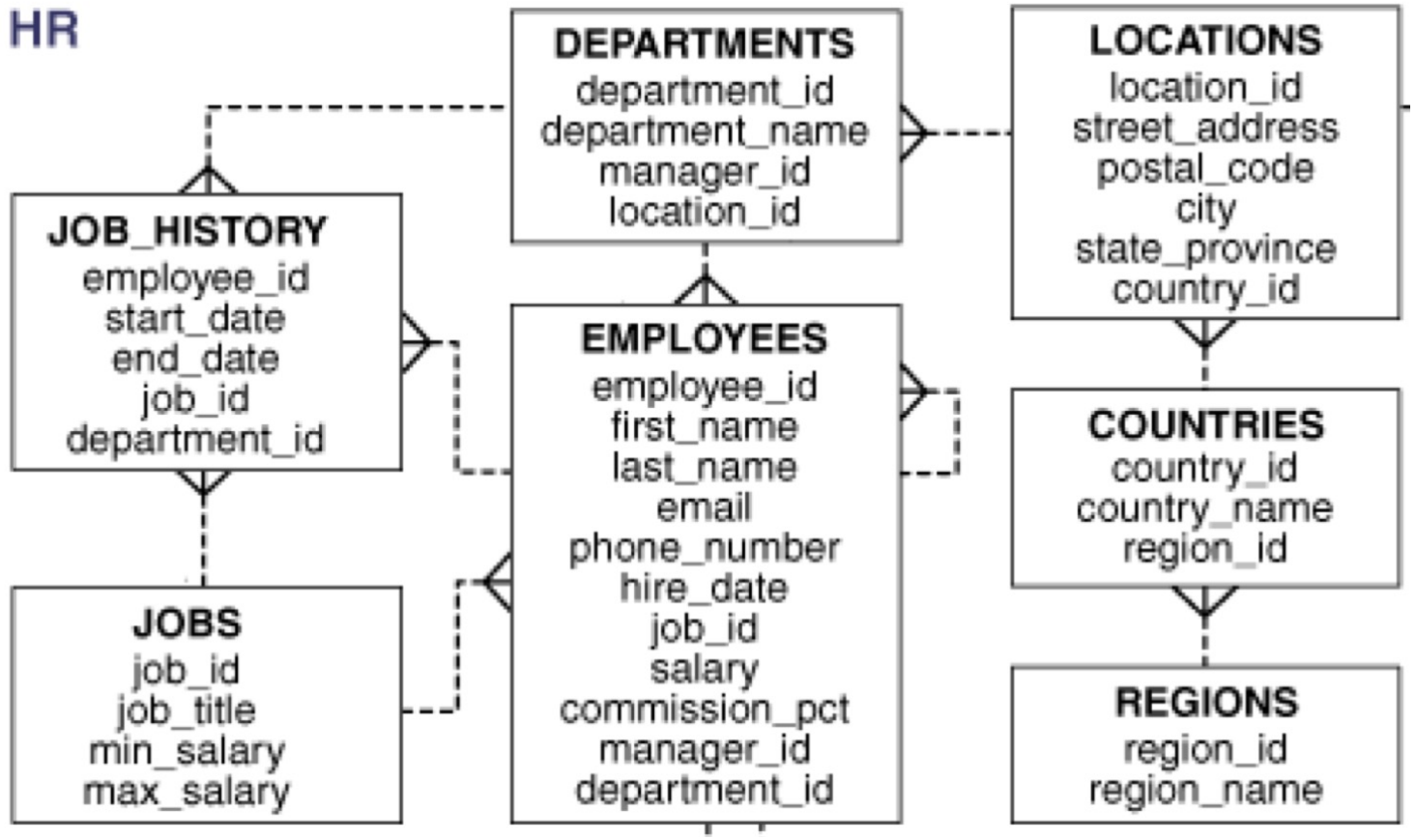
One to Many Relationship

- For one occurrence of first entity –
  - There can be more than one related occurrence of second entity
  - For every occurrence of second entity, only one occurrence of first entity

Many to Many Relationship

- For one occurrence of first entity –
  - There can be more than one related occurrence of second entity
  - For every occurrence of second entity, one or more occurrence of first entity

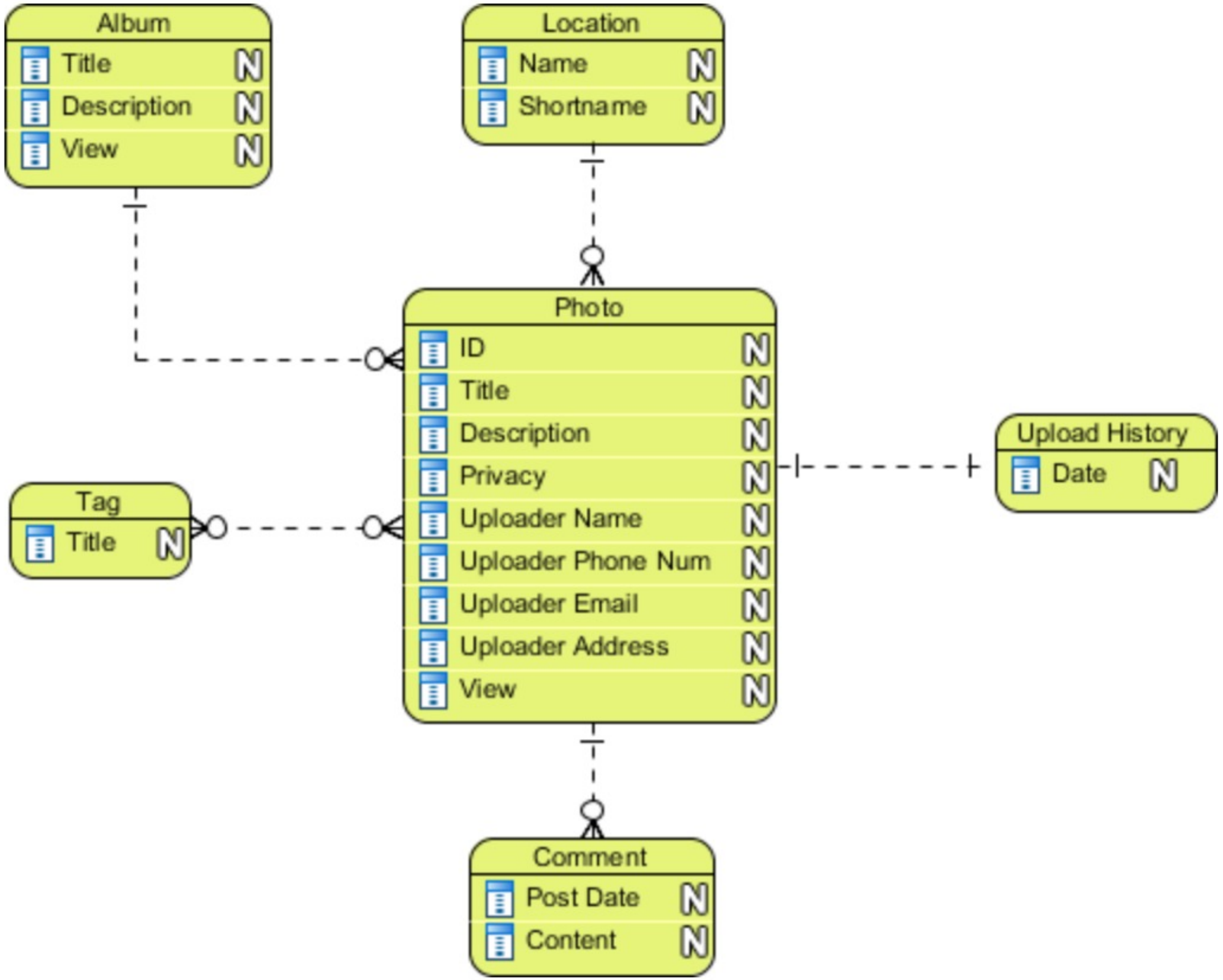
ER Diagram
Conceptual Representation



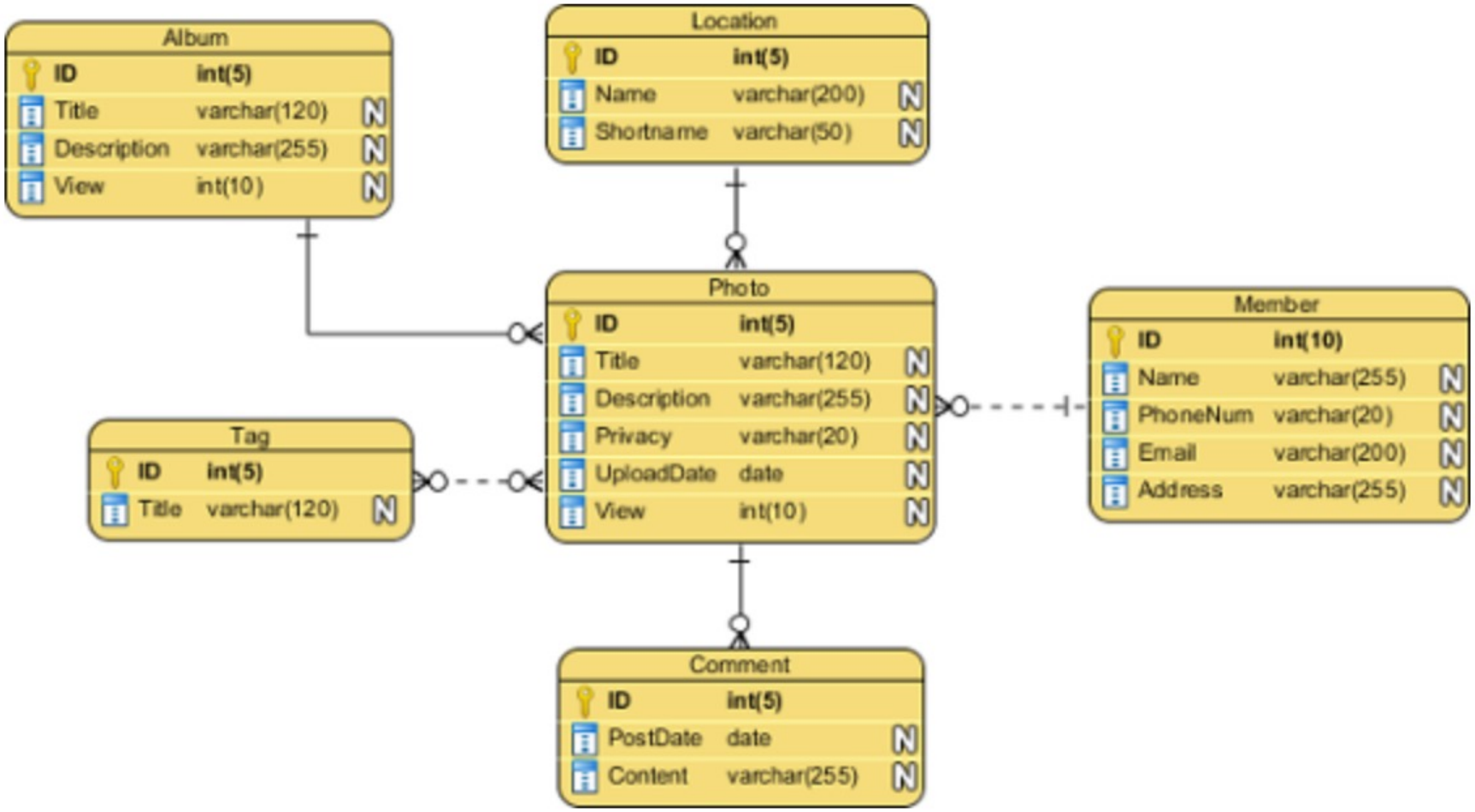
Let’s discuss with example

- Social media picture/photo upload (Write down requirements)
  - What all you can think of for this use case / Interactions

Conceptual Model



Now Read this Logical Model



E. F. Codd Rules

12 Rules

Information representation	Guaranteed Access	Systematic treatment of NULL values	Database description rule
Comprehensive data sub language	View updating	High level update, Insert, Delete	Physical Data Independence
Logical Data Independence	The Distribution Rule	Non-Subversion	Integrity Rule

12 Rules

- **Information Rule**
  - All information including metadata stored in table.

- **Guaranteed Access**
  - Every value of data must be **logically** addressable using combination of –
    - TABLE Name [ENTITY]
    - COLUMN Name [ATTRIBUTE]
    - Primary Key

- **Systematic treatment of NULL values**
  - NULL value is representation of missing information
  - Support for NULL values must be consistent and independent of data types

12 Rules

- **Database description Rule / Data Catalog**
  - Metadata to be stored in form of tables
  - Allow users with appropriate authority to access catalog data
  - Same query language should be used

- **Comprehensive Data sublanguage**
  - RDBMS manageable through its own extension of SQL
  - SQL should support –
    - Data Definition
    - Views
    - Data Manipulation
    - Integrity constraints
    - Authorization
    - Transaction boundary

12 Rules

- **View Updatable**
  - All views that are updatable in theory should be updatable by the system

- **Insert, Update, Delete Rule**
  - RDBMS must do more than just be able to retrieve relational data sets
  - Should be capable to Modify data as a relational set

- **Physical Data Independence**
  - Physical data storage should not impact the system
  - User access must remain logically consistent even when storage changes
  - Application must be limited to interfacing with logical layer

12 Rules

- **Logical Data Independence**
  - Application programmers must be independent of table changes
  - A table should be divisible into 1 or more other tables, Provided it –
    - Preserves all the data (No Data loss)
    - Maintains Primary Key in every fragment

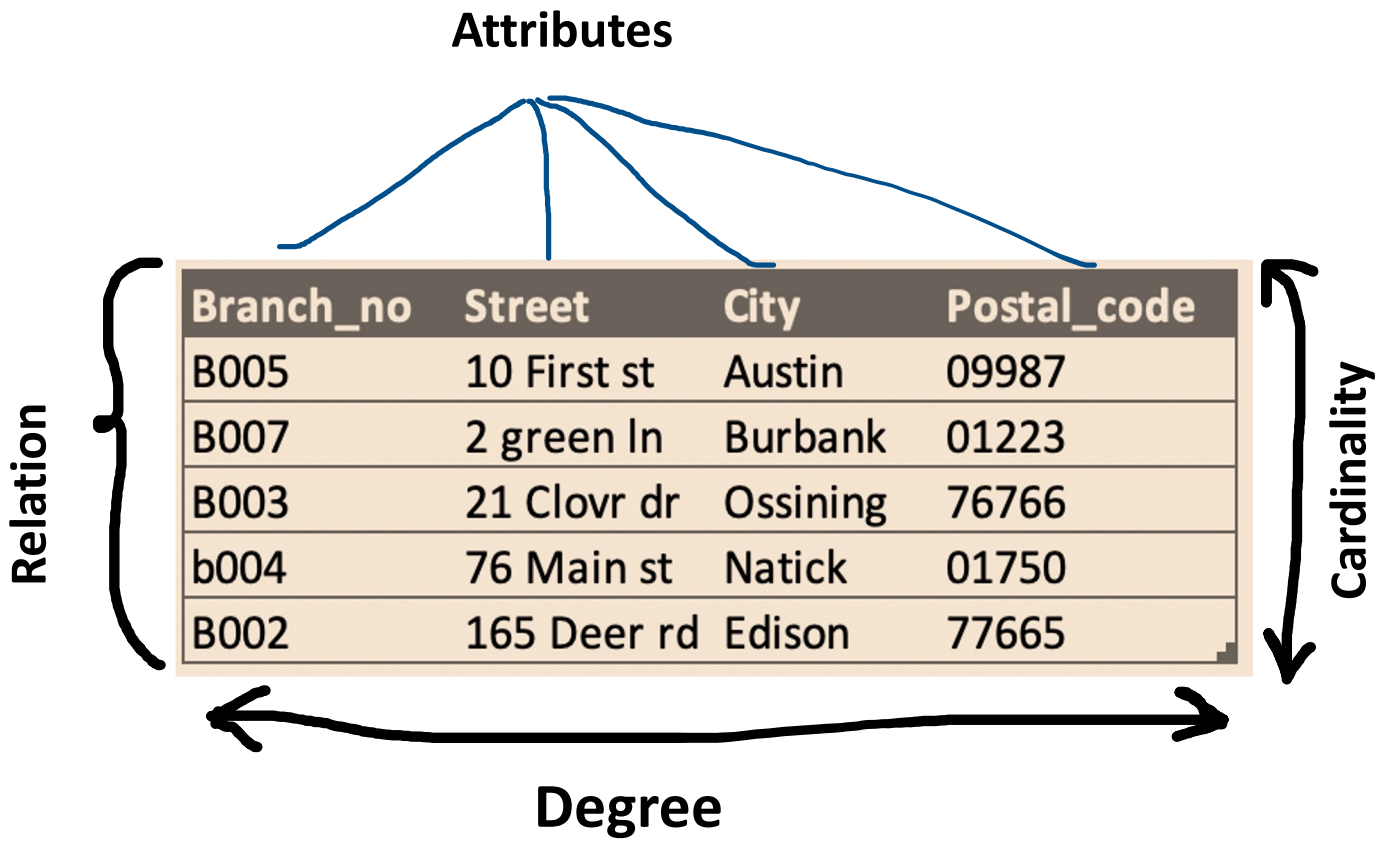
- **Distribution Rule**
  - RDBMS must have distribution independence
  - Geographically distributed database with data stored in pieces is transparent to users.

12 Rules

- **Non-Subversion**
  - Row level access to modify should not bypass integrity constraints
  - RDBMS must be governed by relational rules as its primary laws

- **Integrity Rule**
  - Database should be able to enforce its own integrity
  - Integrity constraints defined in relation data and storable in catalog
  - Should not depend on application programs

Terminology



Branch_no	Street	City	Postal_code
B005	10 First st	Austin	09987
B007	2 green ln	Burbank	01223
B003	21 Clovr dr	Ossining	76766
b004	76 Main st	Natick	01750
B002	165 Deer rd	Edison	77665

- **Relation** is table with rows and columns
- **Attribute** is a named column of a relation
- **Degree** of a Relation is number of attributes
- **Tuple** is row in a relation
- **Cardinality** of a relation is the number of tuples

Relation / Table characteristics

- A table is a two-dimensional structure composed of rows and columns.
- Each table row (Tuple) represents a single entity occurrence within the entity set.
- Each table column represents an attribute, and each column has a distinct name.
- Each row/column intersection represents a single data value.
- All values in a column must conform to the same data format.
- Each column has a specific range of values known as the attribute domain.
- The order of the rows and columns is immaterial to the DBMS.
- Each table must have an attribute or a combination of attributes that uniquely identifies each row.

Functions of any Database Management System

- Data processing
  - Store
  - Retrieve
  - Modify
    - Add/Insert, Change/Update, Remove/Delete
- Catalog access
  - names, types, and sizes of data items
  - names of relationships
  - integrity constraints on the data
  - names of authorized users who have access to the data
  - Types of access allowed
- Transaction
  - Apply all changes or nothing to maintain consistency
- Concurrency
  - multiple users are updating the database concurrently
- Recovery
- Authorization
- Utility
  - File uploads
  - DBA activities
  - Migrations

Questions?