KAMPALA INTERNATIONAL UNIVERSITY

SCHOOL OF MATHEMATICS AND COMPUTING STUDY GUIDE FOR MASTER OF COMPUTER SCIENCE

MCS 8101 DATABASE MANAGEMENT SYSTEM

INSTRUCTIONAL DESIGNER:

DR. BUSINGE PHELIX MBABAZI ABWOOLI
(BCI-KIU, MIS –KIU, PhD MIS, CCNA, HCIA, HETC)

Tel: +256782823607
E-Mail mphelixx@gmail.com /phelix.businge@kiu.ac.ug

Study Unit 3: Data Modeling

Introduction

This unit intend to introduce students to different concepts of Conceptual, logical and physical Data Model., Why do we need Conceptual, logical and physical Model ,State the characteristics of Conceptual, logical and physical Data Model.,State the approaches used in Conceptual Data Model , understand e Entity–Relationship Model (ER Model) and their terms e.g Entit,Attribute,Relationship,Basic concepts: to include key, foreign key, record, relation, Explain the different types of keys used in Entity–Relationship Model (ER Model, Explain the different types of cardinality of Relationship and State the advantages and disadvantages of Entity–Relationship Modeling.

Learning Outcomes of Study Unit 3

Upon completion of this study unit, you should be able to

1. *Define Conceptual, logical and physical Data Model.*
2. *Why do we need Conceptual, logical and physical Model.*
3. *State the characteristics of Conceptual, logical and physical Data Model.*
4. *State the approaches used in Conceptual Data Model.*
5. *Define Entity–Relationship Model (ER Model).*
6. *Explain the following terms used in Entity–Relationship Model (ER Model):*
   - ✓ *Entity*
   - ✓ *Attribute*
   - ✓ *Relationship*
   - ✓ *Basic concepts: to include key, foreign key, record, relation*
7. *Define a key. Explain the different types of keys used in Entity–Relationship Model (ER Model).*
8. *What is a cardinality. Explain the different types of cardinality of Relationship.*
9. *State the advantages and disadvantages of Entity–Relationship Modelling.*

3 Data model

- Data model is an integrated collection of concepts for describing data, relationships between data, and constraints on the data used by an organization.
- A model is a representation of 'real world' objects and events, and their associations.
- It should provide the basic concepts and notations that will allow database designers and end-users to communicate their understanding of the organizational data unambiguously and accurately

A data model can be thought of as comprising three components:

- A structural part, consisting of a set of rules that define how the database is to be constructed;
- A manipulative part, defining the types of operations (transactions) that are allowed on the data (this includes the operations that are used for updating or retrieving data and for changing the structure of the database);
- Possibly a set of integrity rules, which ensures that the data is accurate.

The purpose of a data model is to represent data and to make the data understandable. If it does this, then it can be easily used to design a database

DBMS Database Models

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. While the **Relational Model** is the most widely used database model, there are other models too:

1. Hierarchical Model
2. Network Model
3. Entity-relationship Model
4. Relational Model

Hierarchical Model

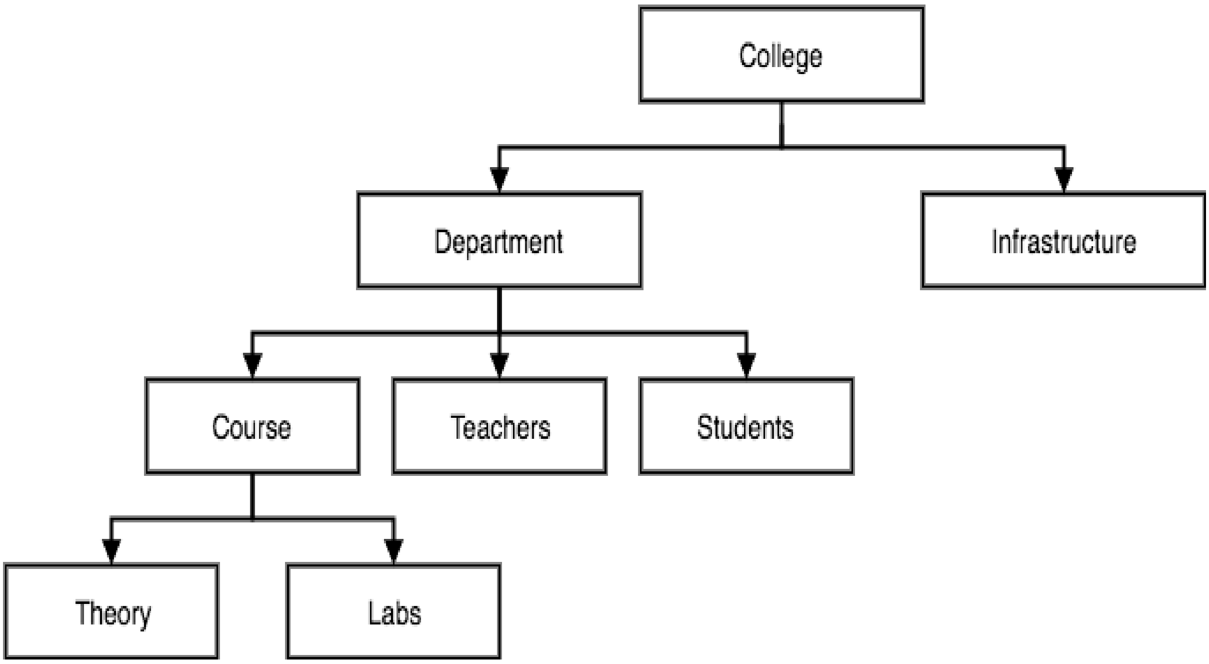
This database model organizes data into a tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

In this model, a child node will only have a single parent node.

This model efficiently describes many real-world relationships like index of a book, recipes etc.

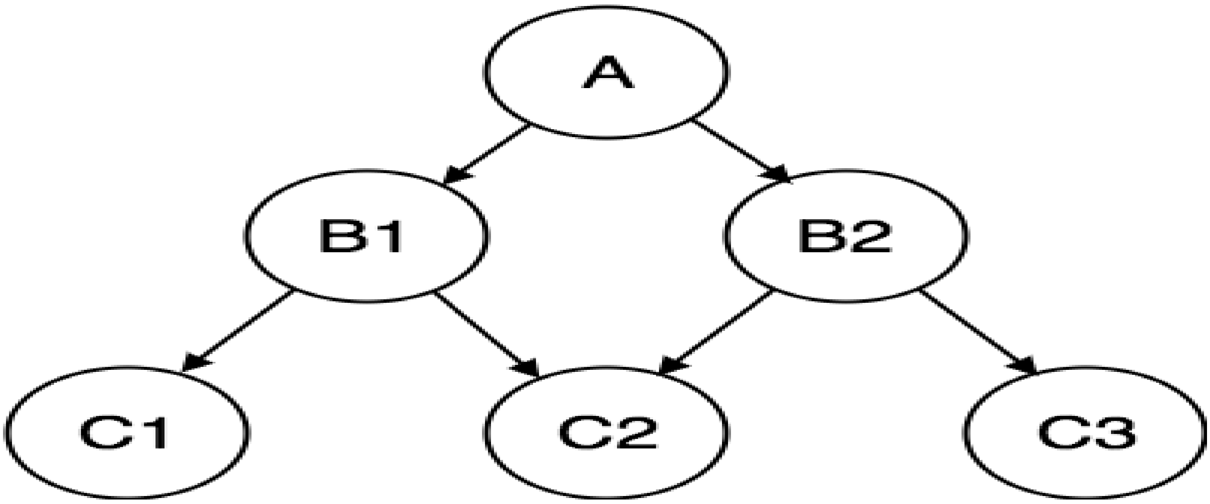
In hierarchical model, data is organised into tree-like structure with one one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and of-course many students.

Hierarchical Model



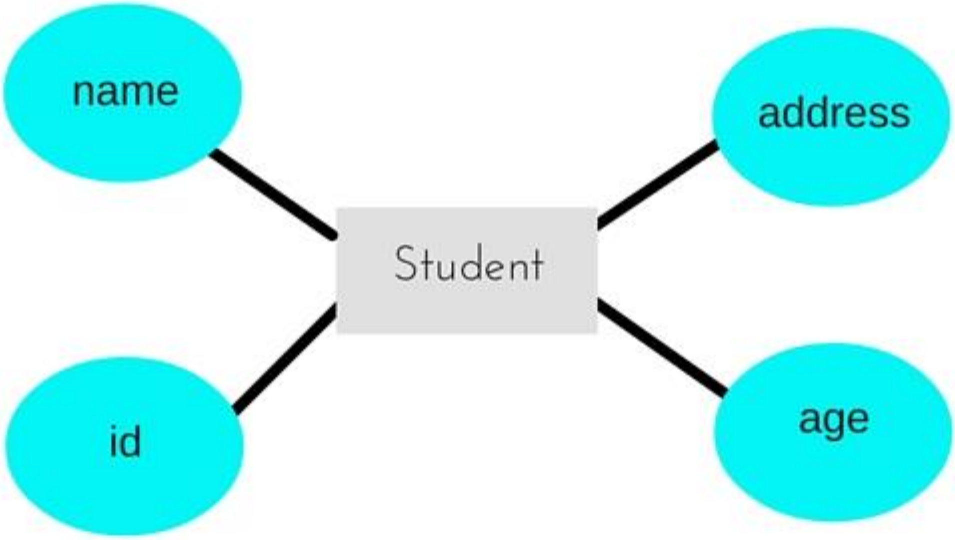
Network Model

- This is an extension of the Hierarchical model. In this model data is organized more like a graph, and are allowed to have more than one parent node.
- In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast. This database model was used to map many-to-many data relationships.
- This was the most widely used database model, before Relational Model was introduced.



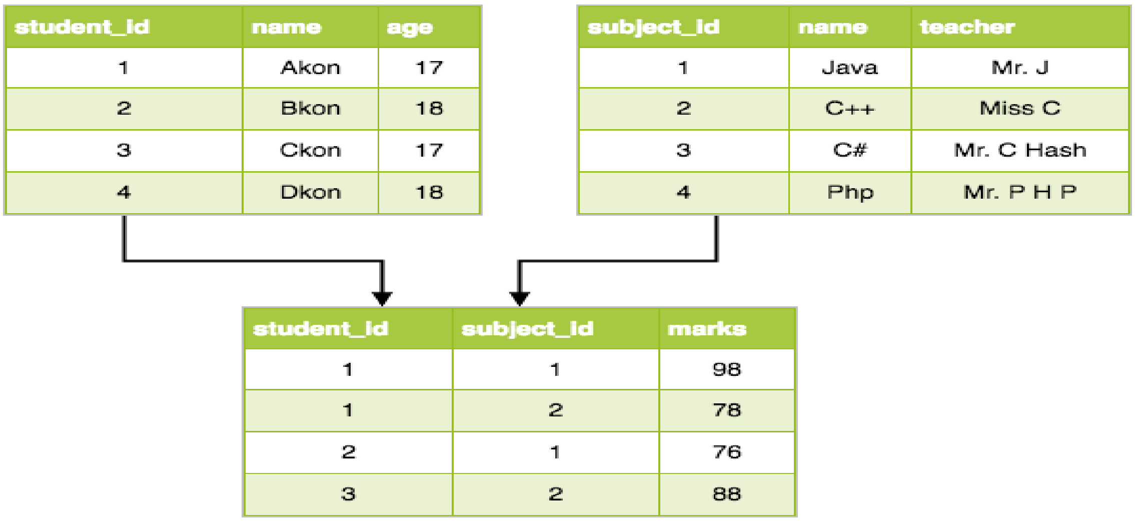
Entity-relationship Model

- In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes.
- Different entities are related using relationships.
- E-R Models are defined to represent the relationships into pictorial form to make it easier for different stakeholders to understand.
- This model is good to design a database, which can then be turned into tables in relational model(explained below).
- Let's take an example, If we have to design a School Database, then **Student** will be an **entity** with **attributes** name, age, address etc. As **Address** is generally complex, it can be another **entity** with **attributes** street name, pincode, city etc, and there will be a relationship between them.



Relational Model

- In this model, data is organised in two-dimensional **tables** and the relationship is maintained by storing a common field.
- This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, infact, we can say the only database model used around the world.
- The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table.
- Hence, tables are also known as **relations** in relational model.
- In the coming tutorials we will learn how to design tables, normalize them to reduce data redundancy and how to use Structured Query language to access data from tables.

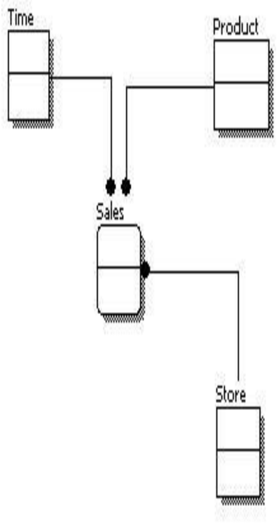


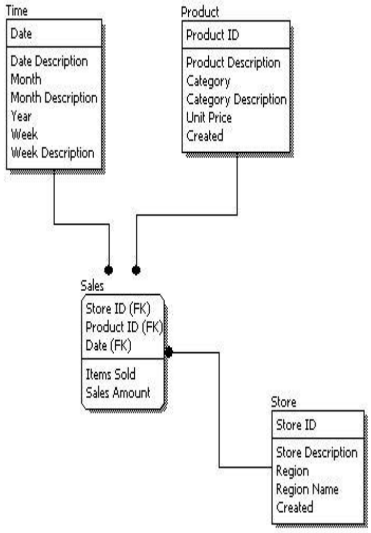
student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

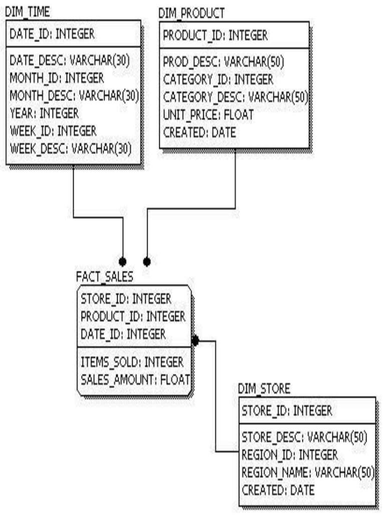
subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Conceptual, **Logical,** **Physical**



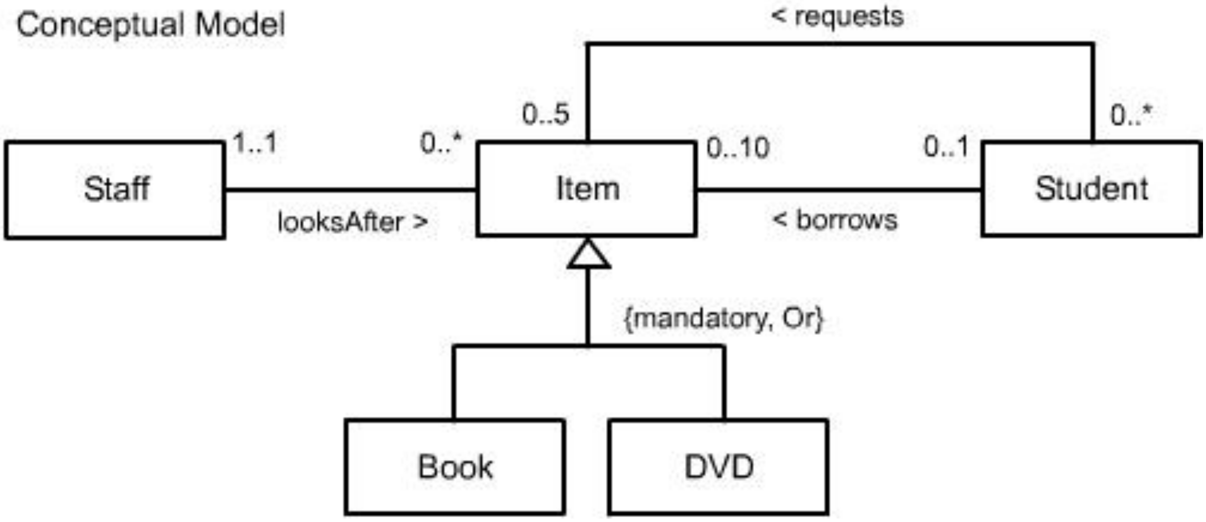




Conceptual Data Model

- Conceptual Data Model is a map of concepts and their relationships used for databases.
- Specifically, it describes the things of significance to an organization (entity classes), about which it is inclined to collect information, and characteristics of (attributes) and associations between pairs of those things of significance (relationships).
- A Conceptual Data Model is the most abstract form of data model.
- It is helpful for communicating ideas to a wide range of stakeholders because of its simplicity.
- Therefore platform-specific information, such as data types, is omitted from a Conceptual data model.
- Other implementation details, such as procedures and interface definitions, are also excluded.

- The purpose of a Conceptual model is to simply establish the Entities, their Attributes and their ‘high-level’ relationships.
- A Conceptual Data Model identifies the highest-level relationships between the different entities.
- Features of Conceptual Data Model include:
    - Includes the important entities and the relationships among them.
    - No attribute is specified.
    - No primary key is specified.
- The figure below is an example of a Conceptual Data Model.



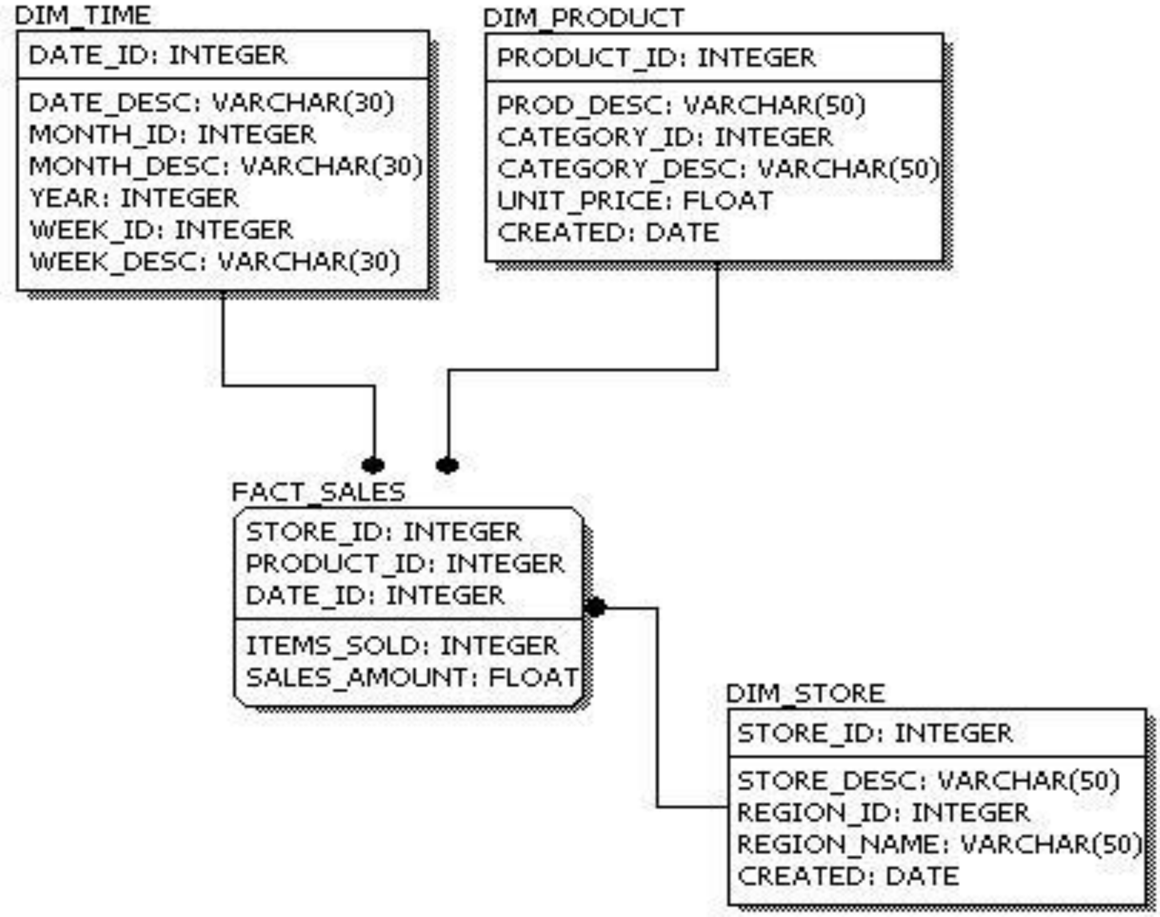
From the figure above, we can see that the only information shown via the Conceptual Data Model is the entities that describe the data and the relationships between those entities.

- No other information is shown through the conceptual data model.
- Conceptual Data Model includes all major entities and relationships and does not contain much detailed level of information about attributes and is often used in the initial planning phase.

- Conceptual Data Model is created by gathering business requirements from various sources like business documents, smart management experts and end users who do the reporting on the database.
- Data modelers create Conceptual Data Model and forward that model to functional team for their review.
- Conceptual Data Model is the first step in constructing a data model in top-down approach and is a clear and accurate visual representation of the business of an organization.
- Conceptual Data Model visualizes the overall structure of the database and provides high-level information about the subject areas or data structures of an organization.
- Conceptual Data Model discussion starts with the main subject of an organization and then all the major entities of each subject area are discussed in detail conceptual data model.

PHYSICAL DATA MODEL

- A Physical Database Model shows all table structures, including column name, column data type, column constraints, Primary Key, Foreign Key, and Relationships between Tables.
- In addition to providing a visual abstraction of the database structure, an important benefit of defining a Physical Data Model is that you can automatically derive the database schema from the model;
- Physical Data Model represents how the model will be built in the database.
- Physical Modeling involves the actual design of a database according to the requirements that were established during Logical Modeling.
- Logical Modeling mainly involves gathering the requirements of the business, with the latter part of Logical Modeling directed toward the goals and requirements of the database.
- Physical Data Models for each implementation would differ significantly, not least due to underlying Operating System requirements that may sit underneath them.
- For example: SQL Server runs only on Microsoft Windows Operating Systems, while Oracle and MySQL can run on Solaris, Linux and other UNIX-based Operating Systems as well as on Windows.
- This means that the disk requirements, security requirements and many other aspects of a Physical Data Model will be influenced by the RDBMS that a database administrator (or an organization) chooses to use.
- The Physical Data Model visually represents the structure of the data as implemented by a Relational Database Schema.
- The figure below is an example of a Physical Data Model.



- Physical Modeling deals with the conversion of the Logical, or Business Model, into a Relational Database Model.
- During Physical Modeling, objects such as tables and columns are created based on entities and attributes that were defined during Logical Modeling.
- Constraints are also defined, including Primary Keys, Foreign Keys, other Unique Keys, and Check Constraints.

- Views can be created from database tables to summarize data or to simply provide the user with another perspective of certain data.
- Other objects such as indexes and snapshots can also be defined during Physical Modeling.
- Physical Modeling is when all the pieces come together to complete the process of defining a database for a business.

Features of a Physical Data Model

1. Specification of all tables and columns.
2. Foreign keys are used to identify relationships between tables.
3. Denormalization may occur based on user requirements.
4. Physical considerations may cause the Physical Data Model to be quite different from the Logical Data Model.
5. Physical Data Model will be different for different RDBMS.

Steps for Physical Data Model Design

- The steps for Physical Data Model Design are as follows:
    1. Convert Entities into Tables.
    2. Convert Attributes into Columns.
    3. Convert Relationships into Foreign Keys.
    4. Modify the Physical Data Model based on Physical Constraints / Requirements.
- Comparing the Physical Data Model shown above with the Logical Data Model, we see the main differences between the two:
    1. Entity names are now Table names.
    2. Attributes are now Column names.
    3. Data type for each column is specified.
- Data types can be different depending on the actual database being used.

Implementation of the Physical Model

a. The implementation of the Physical Model is dependent on the Hardware and Software being used by the company.

b. The Hardware can determine what type of Software can be used because Software is normally developed according to common Hardware and Operating System Platforms.

c. Some database software might only be available for Windows NT systems, whereas other software products such as Oracle are available on a wider range of Operating System Platforms, such as UNIX.

Assignments 1: What are the different examples of Data Types.

Approaches in Conceptual Data Model

There are two (2) approaches used in Conceptual Data Model:

1. Entity–Relationship Model (ER Model)
2. Object-Oriented Modeling (OOM)

Relational Model Terminologies

Relation -Relation is a table with columns and rows
Attribute-Attribute is a named column of a relation
Domain-Domain is the set of allowable values for one or more attributes
Tuple-Tuple is a row of a relation
Degree-Degree is the number of attributes in a relation
Cardinality-Cardinality is the number of tuples in a relation

Relational Model Terminologies

Formal terms	**Alternative1**	**Alternative 2**
Relation	Table	File
Turple	Row	Record
Attribute	Column	Field

What is Table?

In Relational database model, a **table** is a collection of data elements organised in terms of rows and columns. A table is also considered as a convenient representation of **relations**. But a table can have duplicate row of data while a true **relation** cannot have duplicate data. Table is the most simplest form of data storage. Below is an example of an Employee table.

ID	Name	Age	Salary
1	Adam	34	13000
2	Alex	28	15000
3	Stuart	20	18000
4	Ross	42	19020

RDBMS: What is a Tuple?

A single entry in a table is called a **Tuple** or **Record** or **Row**. A **tuple** in a table represents a set of related data. For example, the above **Employee** table has 4 tuples/records/rows.
Following is an example of single record or tuple.

1	**Adam**	**34**	**13000**

Attribute

What is an Attribute?

A table consists of several records(row), each record can be broken down into several smaller parts of data known as Attributes. The above Employee table consist of four attributes, ID, Name, Age and Salary.

Attribute Domain

When an attribute is defined in a relation(table), it is defined to hold only a certain type of values, which is known as Attribute Domain.

Hence, the attribute Name will hold the name of employee for every tuple. If we save employee's address there, it will be violation of the Relational database model.

Name
Adam
Alex
Stuart - 9/401, OC Street, Amsterdam
Ross

What is a Relation Schema?

- A relation schema describes the structure of the relation, with the name of the relation(name of table), its attributes and their names and type.

What is a Relation Key?

- A relation key is an attribute which can uniquely identify a particular tuple(row) in a relation(table).

Relational Integrity Constraints

- Every relation in a relational database model should abide by or follow a few constraints to be a valid relation, these constraints are called as **Relational Integrity Constraints**.

The three main Integrity Constraints are:

1. Key Constraints
2. Domain Constraints
3. Referential integrity Constraints

Key Constraints

We store data in tables, to later access it whenever required. In every table one or more than one attributes together are used to fetch data from tables. The **Key Constraint** specifies that there should be such an attribute(column) in a relation(table), which can be used to fetch data for any tuple(row).

The Key attribute should never be **NULL** or same for two different row of data.

For example, in the **Employee** table we can use the attribute ID to fetch data for each of the employee. No value of ID is null and it is unique for every row, hence it can be our **Key attribute**.

Domain Constraint

Domain constraints refers to the rules defined for the values that can be stored for a certain attribute.

Like we explained above, we cannot store **Address** of employee in the column for **Name**. Similarly, a mobile number cannot exceed 10 digits.

Referential Integrity Constraint

- We will study about this in detail later. For now remember this example, if I say **Supriya** is my girlfriend, then a girl with name Supriya should also exist for that relationship to be present.
- If a table reference to some data from another table, then that table and that data should be present for referential integrity constraint to hold true.

Relational database

Relational databases is a collection of normalized relations with distinct relation names

- **The goals of a relational database are to satisfy dynamic information management needs and to eliminate duplicate data entry.**

Properties of relations

1. Relation name is distinct from all other relation names in relational schema
2. Each cell of relation contains exactly one (atomic) value
3. Each attribute has distinct name
4. values of an attribute are all from the same domain
5. Each tuple is distinct: there are no duplicate turples
6. Order of attribute has no significance
7. Order of turples has no significance

Entity–Relationship Model (ER Model)

a. The Entity-Relationship Model (or ER model) is a way of graphically representing the logical relationships of entities (or objects) in order to create a database.

b. An entity is a piece of data-an object or concept about which data is stored.

c. The main components of ER models are entities (things) and the relationships that can exist among them.

d. In ER Modeling, the structure for a database is portrayed as a diagram, called An Entity-Relationship Diagram (or ER diagram), that resembles the graphical breakdown of a sentence into its grammatical parts.

e. Any ER diagram has an equivalent relational table, and any relational table has an equivalent ER diagram.

f. ER diagramming is an invaluable aid to engineers in the design, optimization, and debugging of database programs.

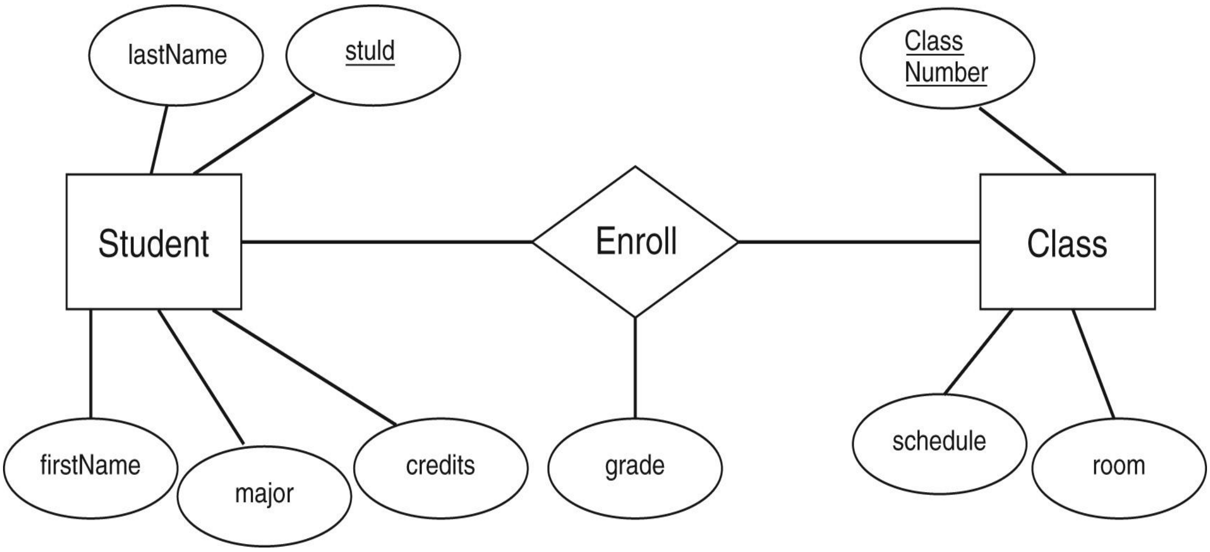
g. In a logical sense, entities are the equivalent of grammatical nouns, such as employees, departments, products, or networks.

h. An entity can be defined by means of its properties, called attributes.

i. Relationships are the equivalent of verbs or associations, such as the act of purchasing, the act of repairing, being a member of a group, or being a supervisor of a department.

j. A relationship can be defined according to the number of entities associated with it, known as the degree.

k. Diagrams created to represent these Entities, Attributes, and Relationships graphically are called Entity–relationship Diagrams.



Entity-Relationship Model - Basic Concepts

1. The ER model defines the conceptual view of a database.
2. It works around real-world entities and the associations among them.
3. At view level, the ER Model is considered a good option for designing databases.

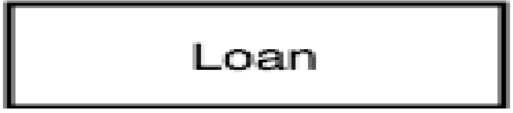
Entity

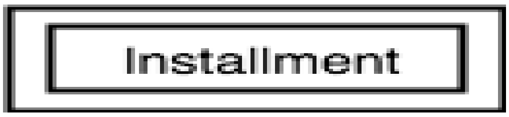
1. An entity can be a real-world object, either animate or inanimate, that can be easily identifiable.
2. For example, in a ***School Database***, *Students*, *Teachers*, *Classes*, and *Courses* offered can be considered as entities.
3. All these entities have some attributes or properties that give them their identity.
4. An entity set is a collection of similar types of entities.
   - An entity set may contain entities with attribute sharing similar values.
   - For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties.
   - Entity is represented in ER Model by a Rectangle Shape.

   - Its can either be **Weak Entity or Strong Entity**

Weak Entity

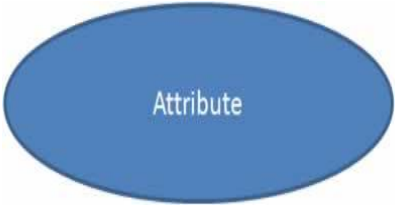
- A weak Entity is represented using double rectangular boxes. It is generally connected to another entity.





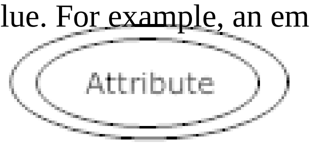
Attributes

a. Entities are represented by means of their properties, called attributes.
b. All attributes have values.
c. For example, a student entity may have name, class, and age as attributes.
d. There exists a domain or range of values that can be assigned to attributes.
e. For example, a student's name cannot be a numeric value.
f. It has to be alphabetic.
g. A student's age cannot be negative, etc.
h. Attribute is represented in ER Model by a Circles (ovals) Shape.

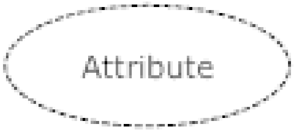


Attributes can either be

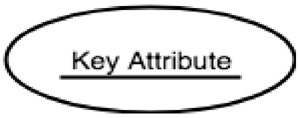
- **Multivalued attribute**
  A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.



- **Derived attribute**
  A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

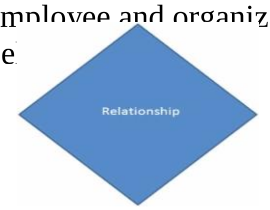


- **Key attribute: To** represent a Key attribute, the attribute name inside the Ellipse is underlined.

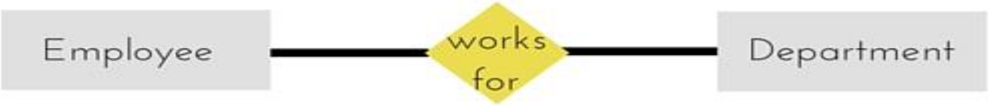


Relationship

1. Associations between entities are called relationships
2. Example: An *Employee* ***works for*** an *Organization*.
3. Here "works for" is a relation between the entities employee and organization.
4. In ER modeling, notation for relationship is given be



- In an ER diagram, symbols are commonly used to represent the types of information.
- Most diagrams will use the following:
    - Boxes represent Entities.
    - Diamonds represent Relationships
    - Circles (ovals) represent Attributes.



INTEGRITY RULES

Entity Integrity Rule

- The entity integrity rule states that for every instance of an entity, the value of the primary key must exist, be unique, and cannot be null.
- **Entity Integrity:** Each entity must have a Primary Key. This ensures that each record in the entity is unique.
- Without entity integrity, the primary key could not fulfill its role of uniquely identifying each instance of an entity

Referential Integrity Rule

- The referential integrity rule states that every foreign key value must match a primary key value in an associated table.
- Referential integrity ensures that we can correctly navigate between related entities.

DATABASE KEYS

Keys are very important part of Relational database model. They are used to establish and identify relationships between tables and also to uniquely identify any record or row of data inside a table.

A Key can be a single attribute or a group of attributes, where the combination may act as a key

Why we need a Key?

- In real world applications, number of tables required for storing the data is huge, and the different tables are related to each other as well.
- Also, tables store a lot of data in them. Tables generally extends to thousands of records stored in them, unsorted and unorganised.
- Now to fetch any particular record from such dataset, you will have to apply some conditions, but what if there is duplicate data present and every time you try to fetch some data by applying certain condition, you get the wrong data. How many trials before you get the right data?
- To avoid all this, **Keys** are defined to easily identify any row of data in a table.

Understand about all the keys using a simple example.
Student table, with field’s student_id, name, phone and age

student_id	name	phone	age
1	Akon	9876723452	17
2	Akon	9991165674	19
3	Bkon	7898756543	18
4	Ckon	8987867898	19
5	Dkon	9990080080	17

Super Key

Super Key is defined as a set of attributes within a table that can uniquely identify each record within a table. Super Key is a superset of Candidate key.
In the table defined above super key would include student_id, (student_id, name), phone etc.
Confused? The first one is pretty simple as student_id is unique for every row of data, hence it can be used to identity each row uniquely.
Next comes, (student_id, name), now name of two students can be same, but their student_id can't be same hence this combination can also be a key.
Similarly, phone number for every student will be unique, hence again, phone can also be a key.
So they all are super keys

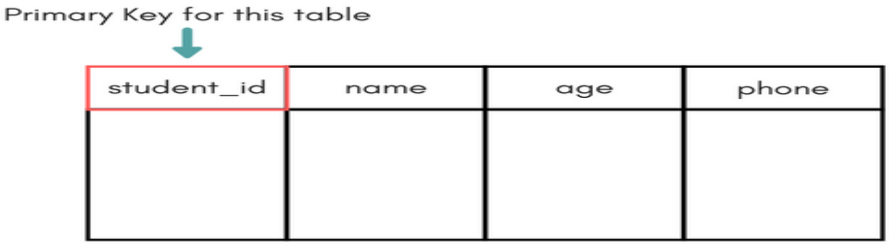
Candidate Key

Candidate keys are defined as the minimal set of fields which can uniquely identify each record in a table. It is an attribute or a set of attributes that can act as a Primary Key for a table to uniquely identify each record in that table. There can be more than one candidate key.
In our example, student_id and phone both are candidate keys for table Student.

- A candiate key can never be NULL or empty. And its value should be unique.
- There can be more than one candidate keys for a table.
- A candidate key can be a combination of more than one columns(attributes)**).**

Primary Key

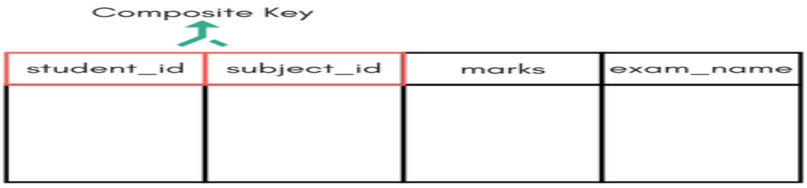
- Primary key is a candidate key that is most appropriate to become the main key for any table. It is a key that can uniquely identify each record in a table.



student_id	name	age	phone

Composite Key

- Key that consists of two or more attributes that uniquely identify any record in a table is called **Composite key**. But the attributes which together form the **Composite key** are not a key independently or individually.


Composite Key

student_id	subject_id	marks	exam_name

Score Table – To save scores of the student for various subjects.

In the above picture we have a **Score** table which stores the marks scored by a student in a particular subject.

In this table student_id and subject_id together will form the primary key, hence it is a composite key.

Secondary or Alternative key

- The candidate key which are not selected as primary key are known as secondary keys or alternative keys.

Non-key Attributes

- **Non-key** attributes are the attributes or fields of a table, other than **candidate key** attributes/fields in a table.

Non-prime Attributes

- **Non-prime** Attributes are attributes other than **Primary Key attribute(s).**.

Review questions

1. *Define Conceptual, logical and physical Data Model.*
2. *Why do we need Conceptual, logical and physical Model.*
3. *State the characteristics of Conceptual, logical and physical Data Model.*
4. *State the approaches used in Conceptual Data Model.*
5. *Define Entity–Relationship Model (ER Model).*
6. *Explain the following terms used in Entity–Relationship Model (ER Model):*
    - ✓ *Entity*
    - ✓ *Attribute*
    - ✓ *Relationship*
    - ✓ *Basic concepts: to include key, foreign key, record, relation*
7. *Define a key. Explain the different types of keys used in Entity–Relationship Model (ER Model).*

8. *What is a cardinality. Explain the different types of cardinality of Relationship.*
9. *State the advantages and disadvantages of Entity–Relationship Modeling*.

References and Additional Reading Materials and

1. *C.J. Date*: An *Introduction to Database Systems*, 7th edn, Addison-Wesley

2. Hoffer, Jeffrey A., Prescott, Mary B. & McFadden R. Fred, *"Modern Database Management"*, Sixth Edition, Prentice Hall, 2002, ISBN 0-13-033969-5

3. Ceri S., C. Zaniolo. Advanced Database Systems.

4. Ceri S., P. Fraternali. Designing Database Applications with Objects and Rules.

5. Frakes W.B., R. Baeza-Yates. Information retrieval. Data structure & Algorithms, Prentice Hall, 1992.

6. Fundamental of Database Systems by R. Elmasri; S. Navate; Benjamin Cummings;

7. Principles of Database Management by James Martin