

# Relational Database Management System



## **What is a database?**

- ▶ Logical collection of objects and data
- ▶ Objects are tables, views, indexes, stored-procedure, logins
- ▶ Each Database needs at least one data file and one log (transaction log) file

## Types of Databases

There are two general types of databases:

- ▶ **User-defined databases:**

Databases we create to run business.

- ▶ **System-defined databases:**

Collection of databases that store vital system information

- ▶ There are four system defined databases in SQL Server: /more than four
- ▶ **master**
- ▶ **model**
- ▶ **msdb**
- ▶ **tempdb**

# Details of System Defined Databases

## **Master database:**

- ▶ Stores all system-level information including Configuration Settings (Initial Size, Growth pattern, Growth limit), Names of databases , Login accounts/Name

## **Model database:**

- ▶ Used as template to create new databases
- ▶ All user databases we create are based on this model database
- ▶ Unless we modify any features during installation, the new database will copy size, and all other features of the model database

## **msdb database:**

- ▶ Used by SQL Server Agent to store application specific data
- ▶ It stores jobs, schedules and alerts related information

## **tempdb database:**

- ▶ Used as temporary processing place for all databases activities
- ▶ It stores temporary data Tempdb used as workspace for query operation.
- ▶ The contents of tempdb will be deleted every time the SQL Server is restarted

# Structured Vs Unstructured data

- Structured Data – Usually exist in relational databases.
- Unstructured Data – Audio file, Video file, Pdf file, Excel, Social Media posting
- Over 90% of the world data are generated in the last 2 years

# DBMS(Database Management System)

- ▶ A system which stores data in
  - Organized format
  - Retrieve data
  - Maintain (manipulate) data

# What is RDBMS?

A type of database management system (DBMS) that stores data in the form of related tables.

- A Primary key – a column that uniquely identify a record in a table. Eg – Employee ID in an Employee Table(Provided an Employee will only have a single ID)
- A Table can only have one Primary key.
- A Foreign key – a column that uses to link two tables and refers to a Primary key in another table

- ▶ **Unique key** is a set of one or more fields/columns of a table that uniquely identify a record in a database table.
- ▶ Unlike PRIMARY KEY, UNIQUE Key allow one value to be NULL meaning UNIQUE constraint allows only one null value per column.
- ▶ A table can have only **primary key** whereas there can be multiple **unique key** on a table

# RDBMS

- My sql
- DB2
- Oracle
- SQL server
- Sybase, Postgrees

# *Entity-relationship diagram*

- ▶ a graphical representation of entities and their relationships to each other.

- ▶ There are three basic elements in ER models:
- ▶ Entities are the "things" about which we seek information. In a retail database there can be entities for Product, Store, etc.
- ▶ Attributes are the data we collect about the entities.
- ▶ Relationships provide the structure needed to draw information from multiple entities.

# There are three types of relationships between entities :

- ▶ **One-to-one:** one instance of an entity (A) is associated with one other instance of another entity (B). For example, in a database of employees, each employee ID (A) is associated with only one social security number (B).
- ▶ An employee can have one and only one SS Number and an SS number belong to one and only one employee.

- ▶ **One-to-many:** one instance of an entity (A) is associated with zero, one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A.
- ▶ Example – Flight and a Flight attendant

- ▶ **Many-to-many:** one instance of an entity (A) is associated with one, zero or many instances of another entity (B), and one instance of entity B is associated with one, zero or many instances of entity
- ▶ Example – A college student and Course.

# Data Modeling

A **data model** organizes data elements and standardizes how the data elements relate to one another

There are three phases in data modeling

- Conceptual
- Logical
- Physical

# Conceptual data model

Identifies the highest-level relationships between the different entities.

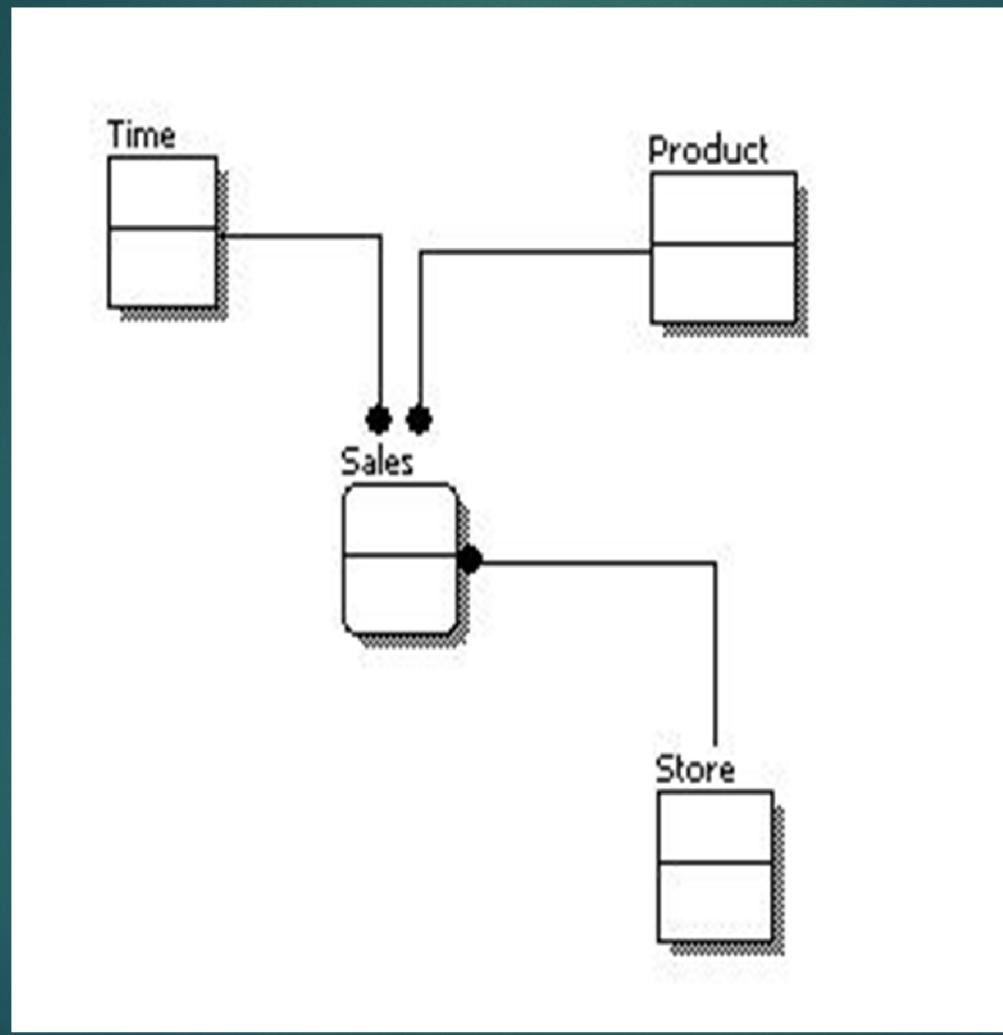
Features of conceptual data model include:

Main entities and the relationships among them.

No attribute is specified.

No primary key is specified.

# Conceptual Data Model



# Logical data model

- ▶ Describes the data in as much detail as possible, without regard to how they will be physical implemented in the database.

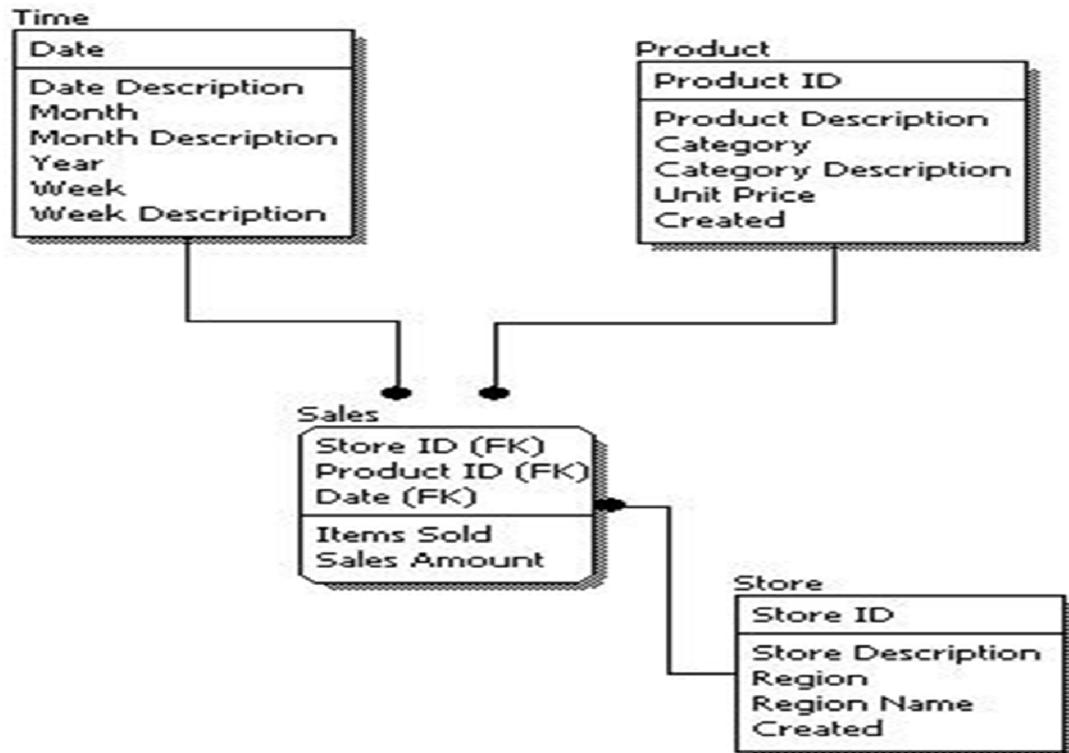
Features of a logical data model include:

- ▶ All entities and relationships among them.
- ▶ All attributes for each entity are specified.
- ▶ The primary key for each entity is specified.
- ▶ Foreign keys (keys identifying the relationship between different entities) are specified.

# Steps for designing the logical data model

- ▶ Specify primary keys for all entities.
- ▶ Find the relationships between different entities.
- ▶ Find all attributes for each entity.
- ▶ Resolve many-to-many relationships.
- ▶ Normalization.

# Logical Data Model



# Logical Vs Conceptual

- ▶ In a logical data model, primary keys are present, whereas in a conceptual data model, no primary key is present.
- ▶ In a logical data model, all attributes are specified within an entity. No attributes are specified in a conceptual data model.
- ▶ Relationships between entities are specified using primary keys and foreign keys in a logical data model. In a conceptual data model, the relationships are simply stated, not specified, so we simply know that two entities are related, but we do not specify what attributes are used for this relationship.

# Physical data model

- ▶ Physical data model represents how the model will be built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables.

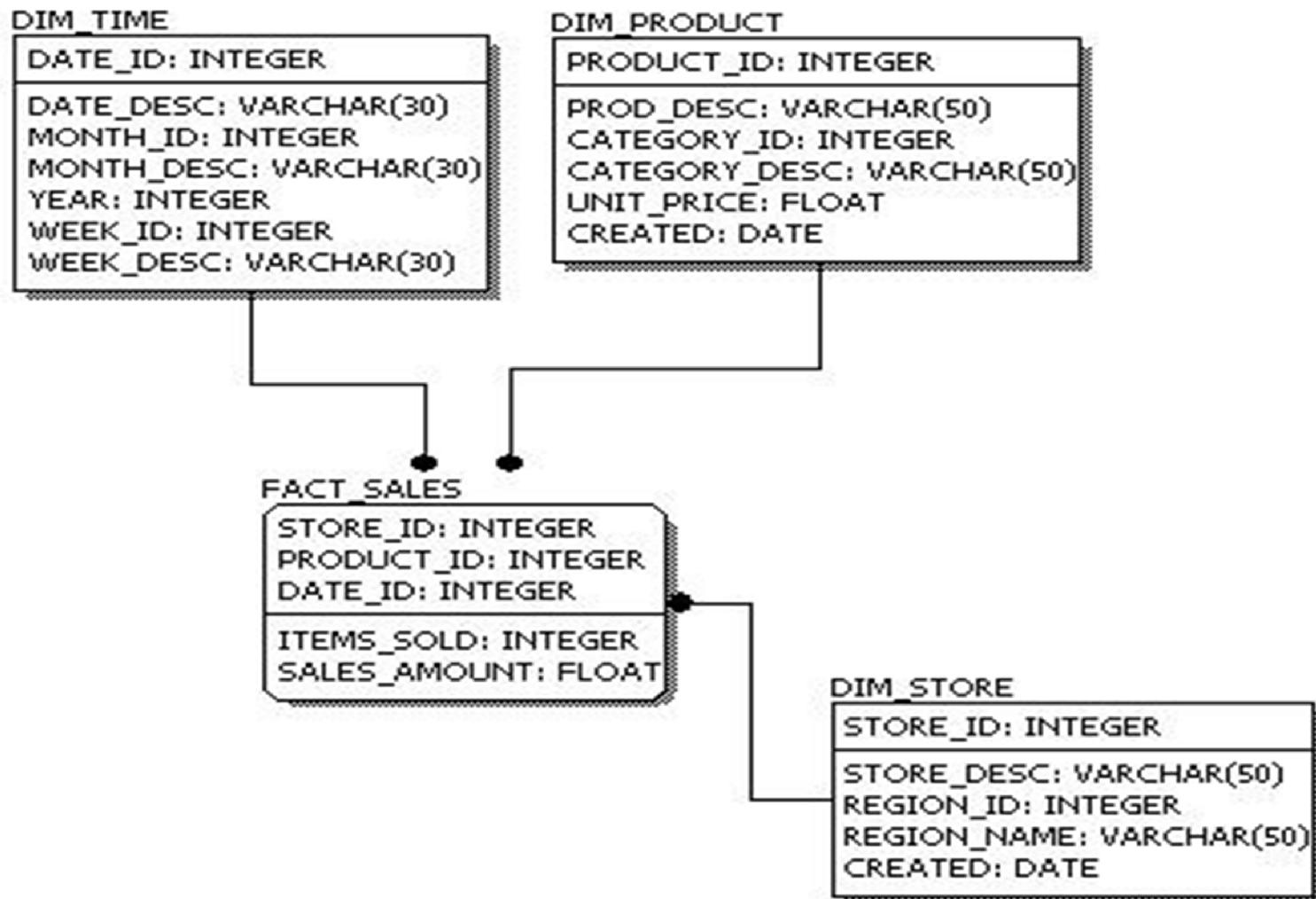
# Features of a physical data model

- ▶ Specification all tables and columns.
- ▶ Foreign keys are used to identify relationships between tables.
- ▶ Denormalization may occur based on user requirements.
- ▶ Physical considerations may cause the physical data model to be quite different from the logical data model.
- ▶ Physical data model will be different for different RDBMS. For example, data type for a column may be different between MySQL and SQL Server.

# The steps for physical data model design

- ▶ Convert entities into tables.
- ▶ Convert relationships into foreign keys.
- ▶ Convert attributes into columns.
- ▶ Modify the physical data model based on physical constraints / requirements

# physical data model.



Comparing the physical data model shown above with the logical data model diagram

- ▶ Entity names are now table names.
- ▶ Attributes are now column names.
- ▶ Data type for each column is specified. Data types can be different depending on the actual database being used.