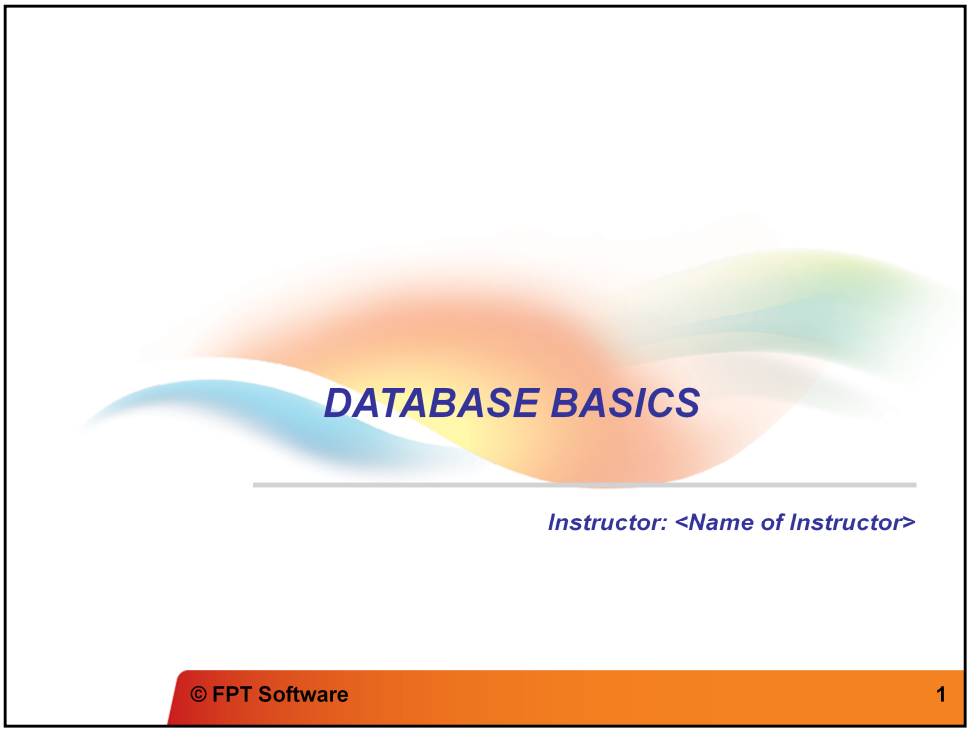


Latest updated by: HanhTT1

Agenda

- Database Concepts
- Entity Relationship Modeling
- MS SQL Server (MSSQL Express 2008)
  - Overview
  - Data Types
  - DDL
  - DML

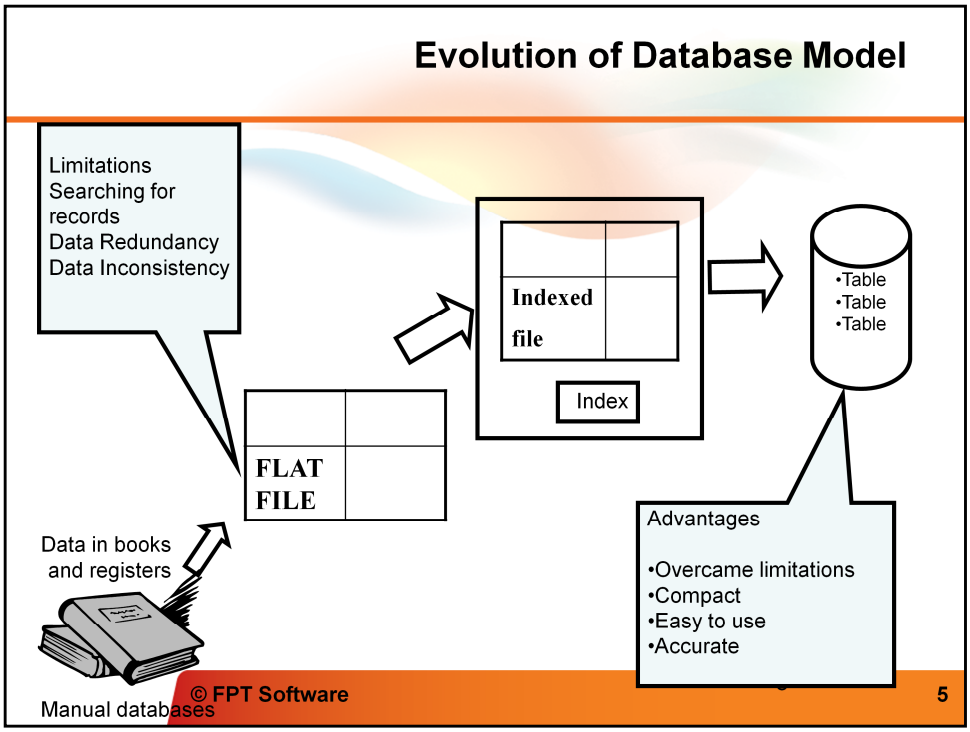
Data Model

- A data model is a “description” of both a container for data and a methodology for storing and retrieving data from container.
- “You can think of a data model as the infrastructure of the data organizations, in other words, the way data is presented to the user.”
- Data model is
  - ✓ Not a thing
  - ✓ You cannot touch it
  - ✓ Data model are abstractions, mathematical algorithms & Concepts.

Database Systems Models

Data base management systems follow particular models (known as database models) to store and manipulate
data. A data base model is characterized by:

1. The way it stores data : STRUCTURE
2. The way data in the structure are manipulated: OPERATIONS


Evolution of Database Model
Limitations
Searching for records
Data Redundancy
Data Inconsistency
Indexed file
Index
•Table
•Table
•Table
FLAT FILE
Data in books and registers
Manual databases
Advantages
•Overcame limitations
•Compact
•Easy to use
•Accurate
© FPT Software
5

The Relational Database

- "A DBMS that manages data as collection of tables in which all data relationships are represented by common values in related tables."
- "A DBMS that follows all the twelve rules of CODD is called RDBMS"

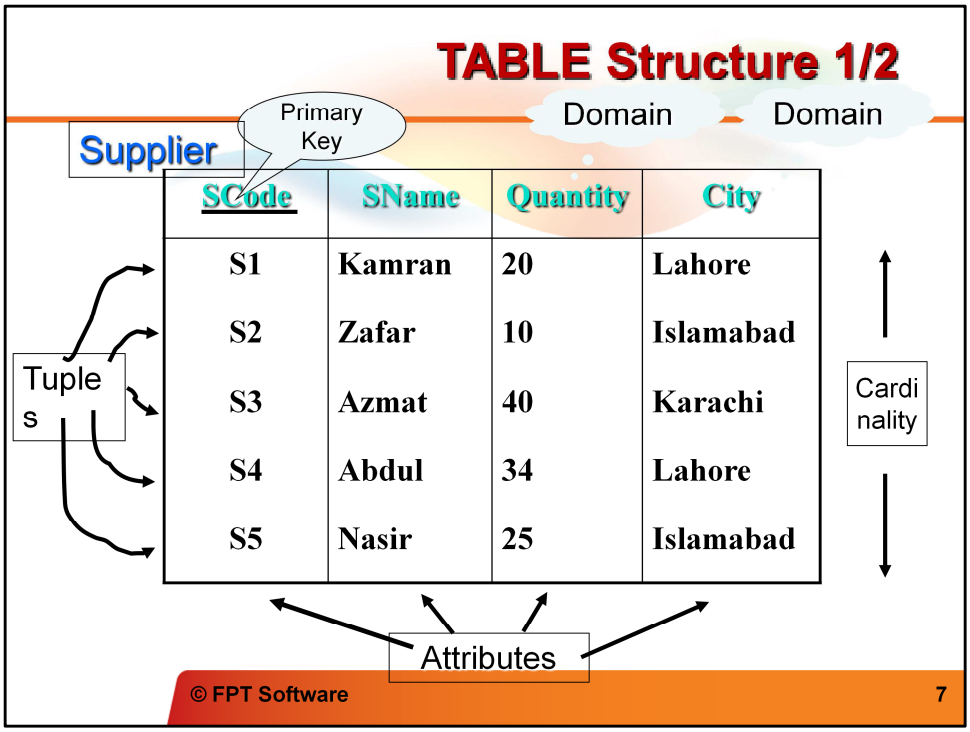


TABLE Structure 1/2
Domain
Domain
Primary Key
Supplier
SCode
SName
Quantity
City
S1
Kamran
20
Lahore
S2
Zafar
10
Islamabad
S3
Azmat
40
Karachi
S4
Abdul
34
Lahore
S5
Nasir
25
Islamabad
Tuples
Cardinality
Attributes
© FPT Software
7

TABLE Structure 2/2

- **Attribute (field):**
  Member of a relation type (set / table).
- **Attribute Name:**
  All attribute names must be unique within a table / relation.
- **Attribute Domain:**
  A set of all possible values that can be attain by an attribute.
- **Attribute Value Set:**
  Values currently contained in an attribute.
- **Relation / Table Degree:**
  Number of attributes in a relation / table.
- **Tuples:** Rows in a table / relation.
- Cardinality: Number of tuples in a relation / table.

Ex: Instance of Students Relation

```
Student(studno,name,address)
Course(courseno,lecturer)
```

Schema

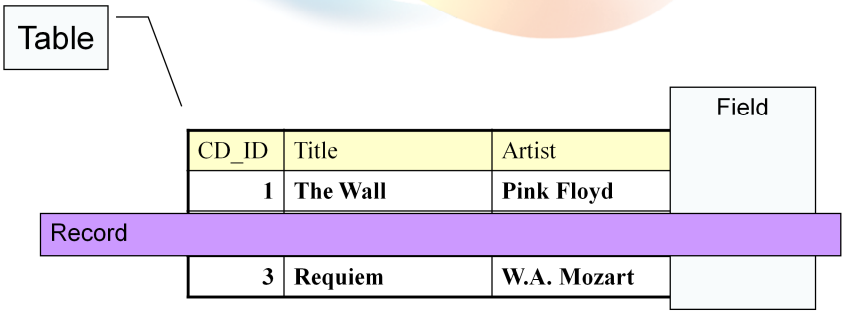
```
Student(123,Bloggs,Woolton)
        (321,Jones,Owens)
```

Instance

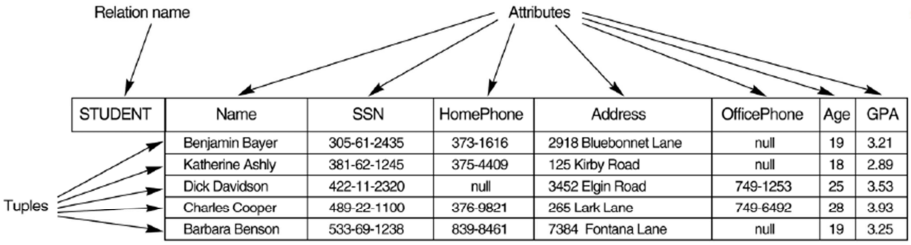
sid	Name	Login	age	GPA
53666	Jones	Jones@ca	18	3.4
53444	smith	Smith@ecs	18	3.2
53777	Blake	Blake@aa	19	3.8

- Cardinality = 3, arity = 5 , all rows distinct
- Do all values in each column of a relation instance have to be distinct?

Relational Database Concepts



Basic component of a Relation



STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Tuple:

- **The actual data values for the attributes of a relation are stored in *tuples*, or rows, of the table.**
- **It is not necessary for a relation to have rows in order to be a relation; even if no data exists for the relation**
- **The relation remains defined with its set of attributes**

Attribute:

The term attribute refers to characteristics. This simply means that what the column contains will be defined by the attribute of the column

Examples of Attribute Domains

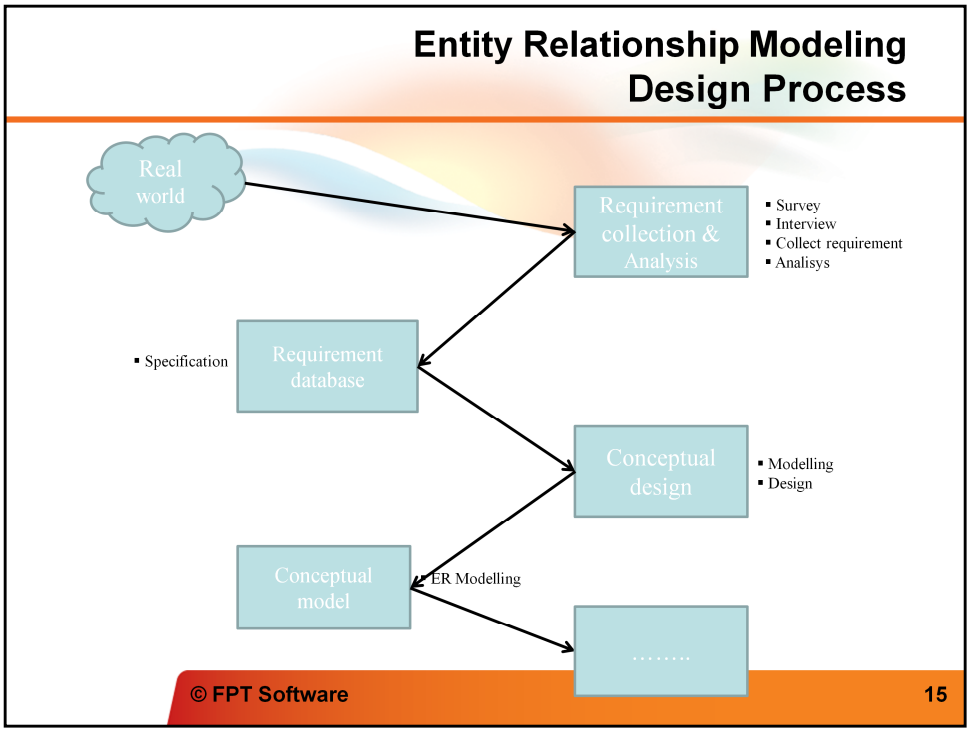
Attribute	*Domain Name*	*Meaning*	*Domain Definition*
Bno	BRANCH_NUMBERS	The set of all possible branch numbers	character: size 3, range B1–B99
Street	STREET_NAMES	The set of all street names in Britain	character: size 25
Area	AREA_NAMES	The set of all area names in Britain	character: size 20
City	CITY_NAMES	The set of all city names in Britain	character: size 15
Pcode	POST_CODES	The set of all postcodes in Britain	character: size 8
Tel_No	TELFAX_NUMBERS	The set of all telephone and fax numbers in Britain	character: size 13
Fax_No	TELFAX_NUMBERS	The set of all telephone and fax numbers in Britain	character: size 13
Sex	SEX	The sex of a person	character: size 1, value M or F
DOB	DATES_OF_BIRTH	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
Salary	SALARIES	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00

DBMS vs. RDBMS

DBMS	RDBMS
The concepts of relationships is missing in a DBMS. If it exits it is very less.	It is based on the concept Of relationships
Speed of operation is very slow	Speed of operation is very Fast
Hardware and Software requirements are minimum	Hardware and Software requirements are High
Platform used is normally DOS	Platform used can be any DOS, UNIX,VAX,VMS, etc
Uses concept of a file	Uses concept of table
DBMS normally use 3GL	RDBMS normally use a 4GL
Examples are dBase, FOXBASE, etc	Examples are ORACLE, INGRESS, SQL Server 2000 etc

Entity Relationship Modeling
Database design

- **Database design** is the process of producing a detailed data model of a database
- **Process of Database Design:**
  - Determine the relationships between the different data elements.
  - Superimpose a logical structure upon the data on the basis of these relationships
  - **Database designs also include ER (Entity Relationship Model) diagrams.**
  - An ER diagram is a diagram that helps to design databases in an efficient way.



Determine the purpose of the database - This helps prepare for the remaining steps.

Find and organize the information required - Gather all of the types of information to record in the database, such as product name and order number.

Divide the information into tables - Divide information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.

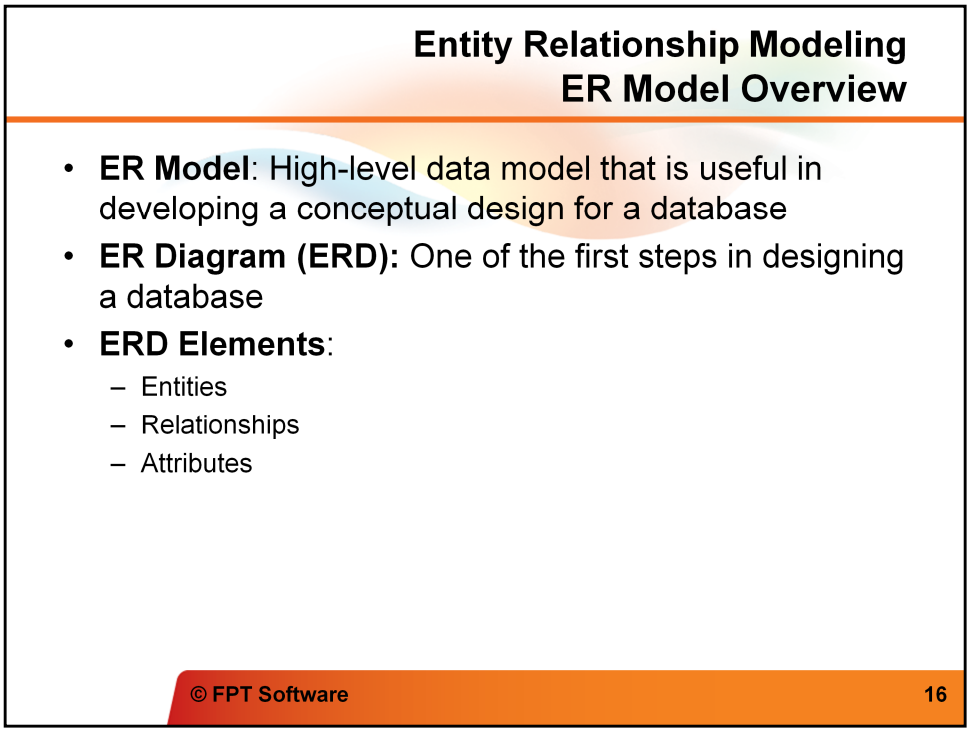
Turn information items into columns - Decide what information needs to stored in each table. Each item becomes a field, and is displayed as a column in the table. For example, an Employees table might include fields such as Last Name and Hire Date.

Specify primary keys - Choose each table's primary key. The primary key is a column that is used to uniquely identify each row. An example might be Product ID or Order ID.

Set up the table relationships - Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.

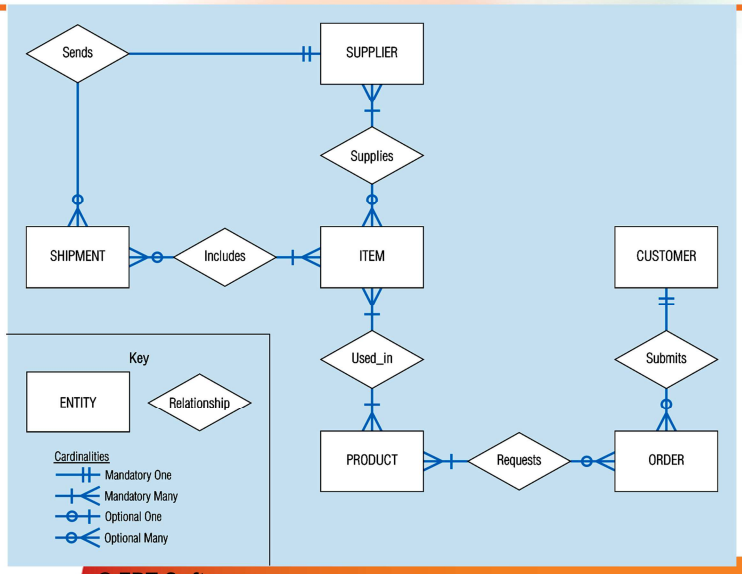
Refine the design - Analyze the design for errors. Create tables and add a few records of sample data. Check if results come from the tables as expected. Make adjustments to the design, as needed.

Apply the normalization rules - Apply the data normalization rules to see if tables are structured correctly. Make adjustments to the tables



In 1976, Chen developed the **Entity-Relationship (ER) model**, a high-level data model that is useful in developing a conceptual design for a database. Creation of an ER diagram, which is one of the first steps in designing a database, helps the designer(s) to understand and to specify the desired components of the database and the relationships among those components. An ER model is a diagram containing entities or "items", relationships among them, and attributes of the entities and the relationships.

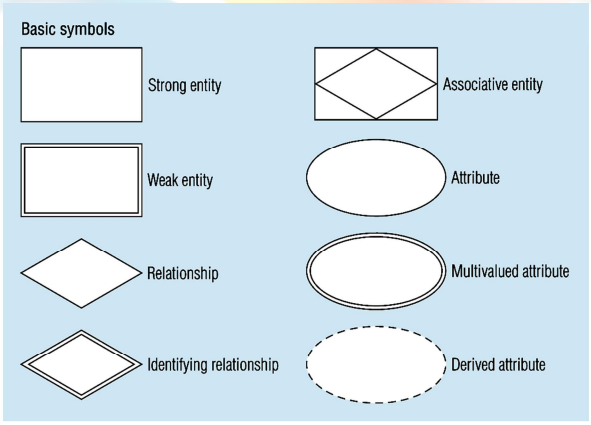
Entity Relationship Modeling
Sample E-R Diagram



Entity Relationship Modeling
Basic E-R Notation

Entity symbols

Relationship symbols



A special entity that is also a relationship

Attribute symbols

Entity Relationship Modeling Attributes

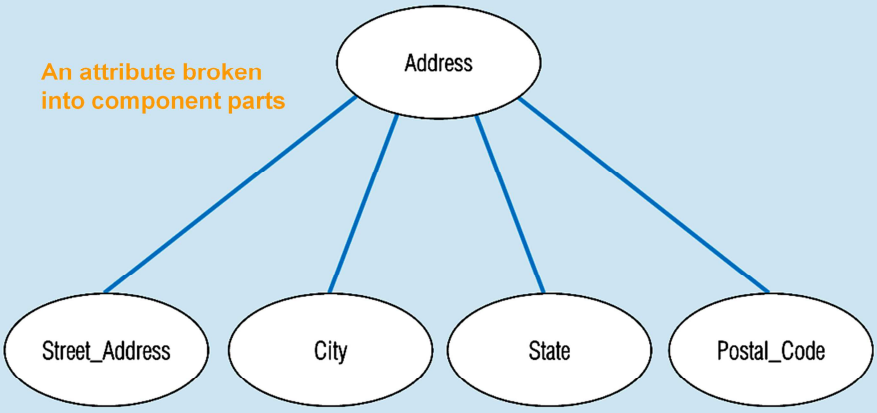
- Attribute - property or characteristic of an entity type
- Classifications of attributes:
  - Simple versus Composite Attribute
  - Single-Valued versus Multivalued Attribute
  - Stored versus Derived Attributes
  - Identifier Attributes

Entity Relationship Modeling
Identifiers (Keys)

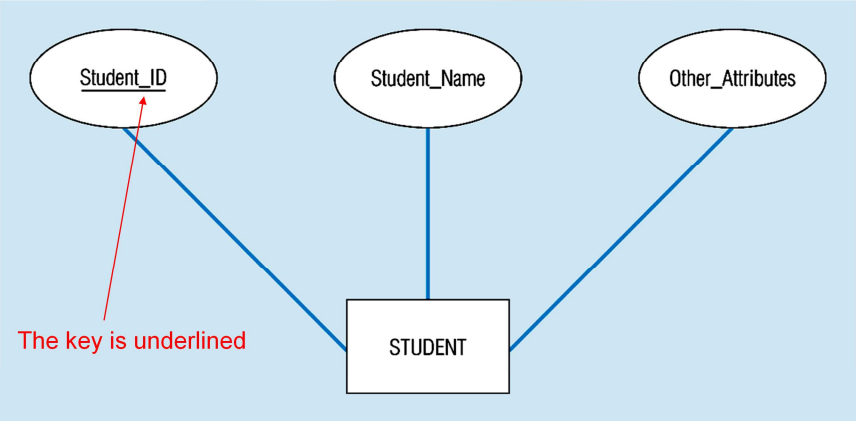
- Identifier (Key) - An attribute (or combination of attributes) that uniquely identifies individual instances of an entity type
- Simple Key versus Composite Key
- Candidate Key – an attribute that could be a key…satisfies the requirements for being a key
- Characteristics of Identifiers:
  - Will not change in value
  - Will not be null
  - No intelligent identifiers (e.g. containing locations or people that might change)
  - Substitute new, simple keys for long, composite keys

Entity Relationship Modeling
A composite attribute

An attribute broken into component parts



Entity Relationship Modeling
Simple key attribute



Entity Relationship Modeling
Cardinality of Relationships 1/2

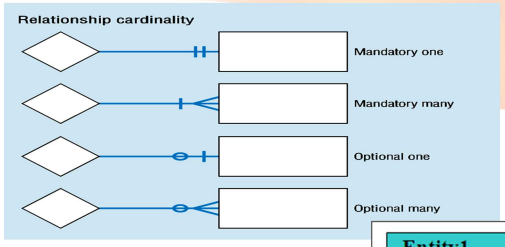
Relationships

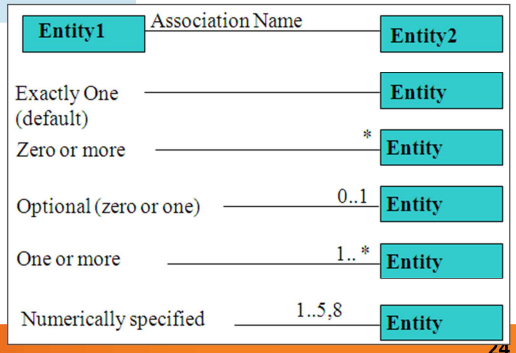
- One – to – One: Each entity in the relationship will have exactly one related entity
- One – to – Many: An entity on one side of the relationship can have many related entities, but an entity on the other side will have a maximum of one related entity
- Many – to – Many: Entities on both sides of the relationship can have many related entities on the other side

Cardinality Constraints - the number of instances of one entity that can or must be associated with each instance of another entity.

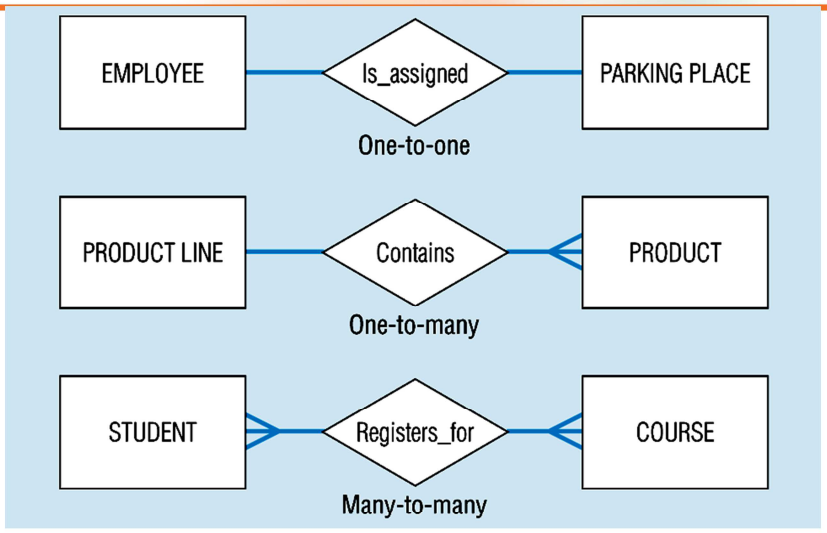
- Minimum Cardinality
  - If zero, then optional
  - If one or more, then mandatory
- Maximum Cardinality: The maximum number

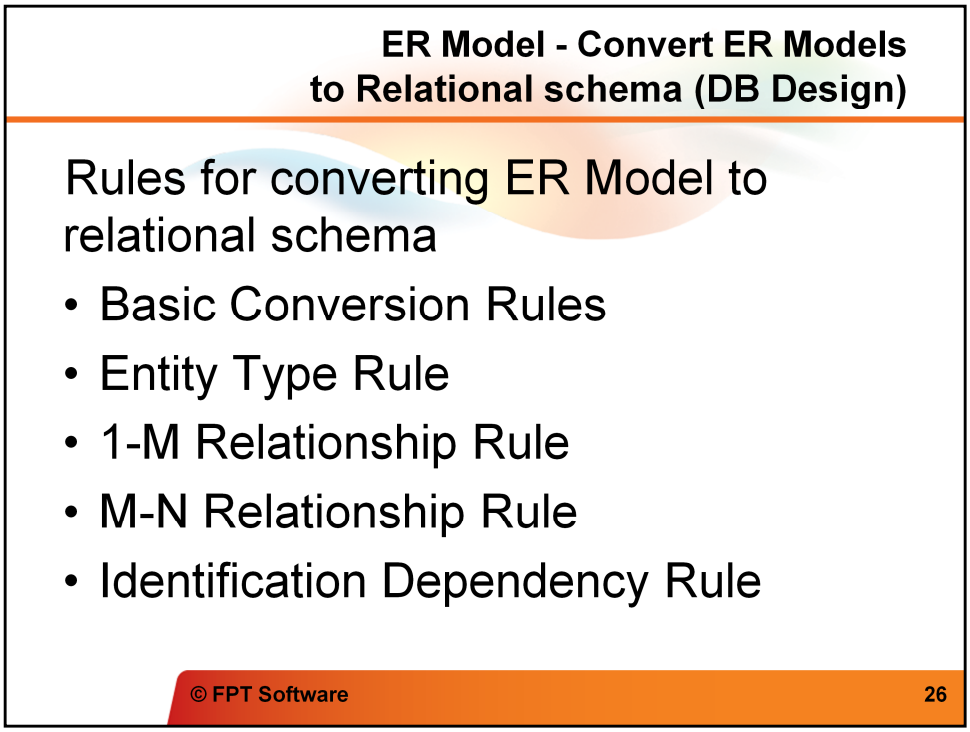
Entity Relationship Modeling
Cardinality of Relationships 2/2





Entity Relationship Modeling
Binary relationships





Entity Type Rule: Each entity type (except subtypes) becomes a table. The PK of ET (if not weak) becomes the PK of the table. The attributes of the ET become columns in the table. This rule **should be used first before the relationship rules**.

1-M Relationship Rule: Each 1-M relationship becomes a FK in the table corresponding to the child type (the entity type near Crow's Foot symbol). If the minimum cardinality on the parent side of the relationship is one, the FK cannot accept null values (NOT NULL must be used).

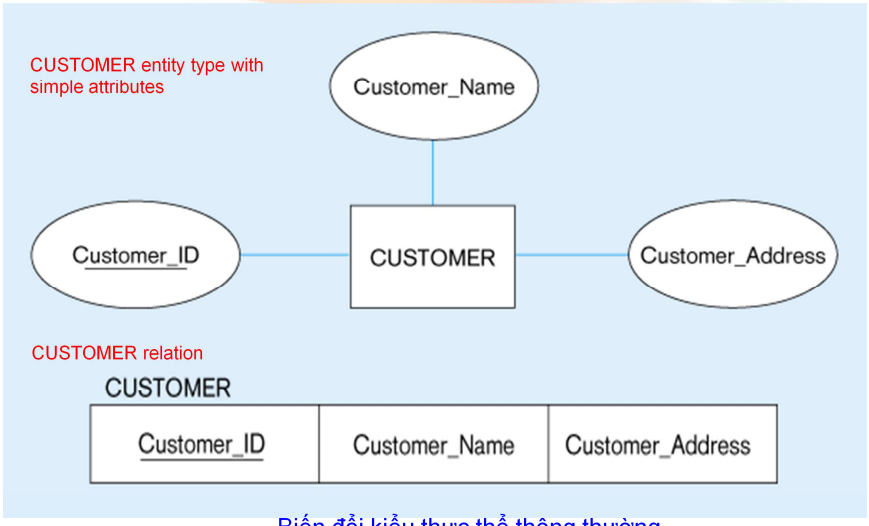
M-N Relationship Rule: Each M-N relationship becomes a separate table. The **PK of the table is a combined key** consisting of the primary keys of the entity types participating in the M-N relationship.

Identification Dependency Rule: Each identifying relationship (denoted by a solid relationship line) adds a component to a PK. The PK of the table corresponding to the weak entity consists of:

The underlined local key (if any) in the weak entity and

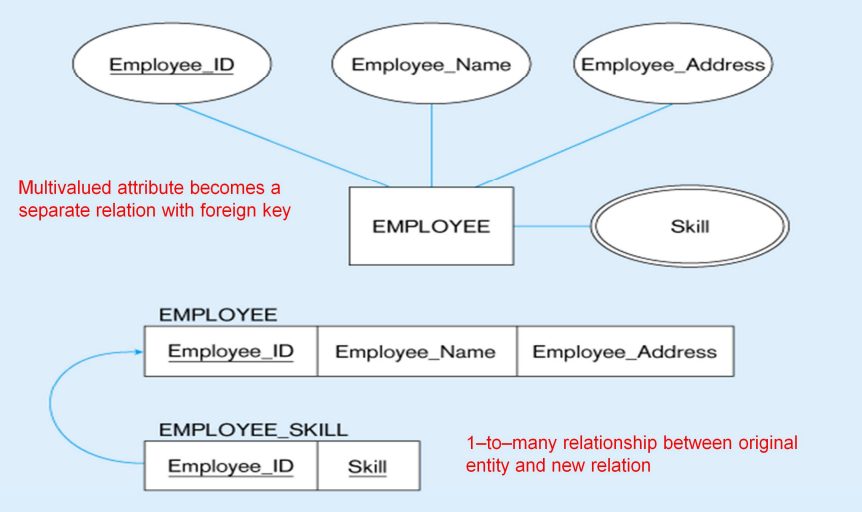
The PK(s) of the entity type(s) connected by identifying relationship(s).

ER Model: DB Design
Rule 1 - Example



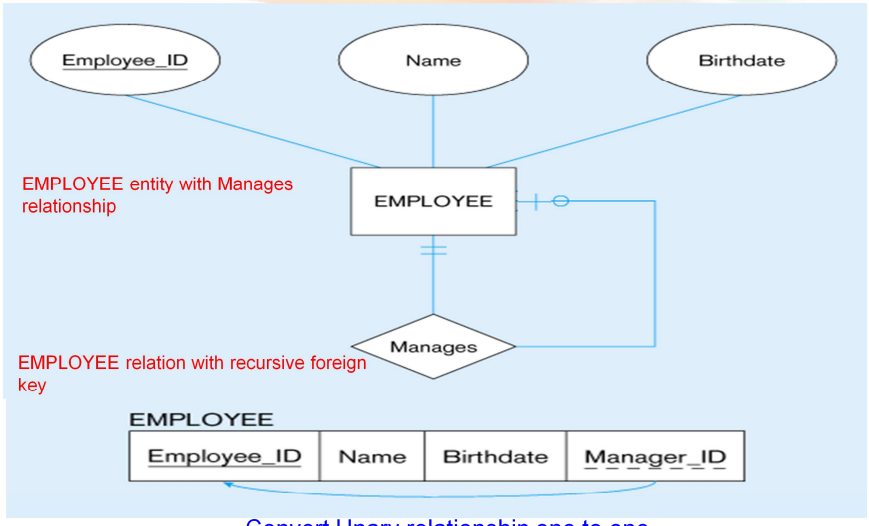
Biến đổi kiểu thực thể thông thường

ER Model: DB Design
Rule 2 - Example



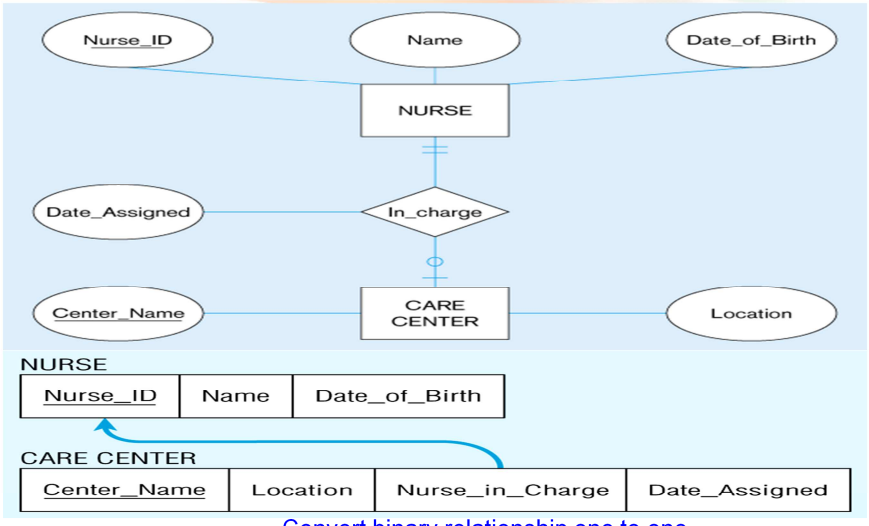
Convert Multivalue attribute

ER Model: DB Design
Rule 3 - Example



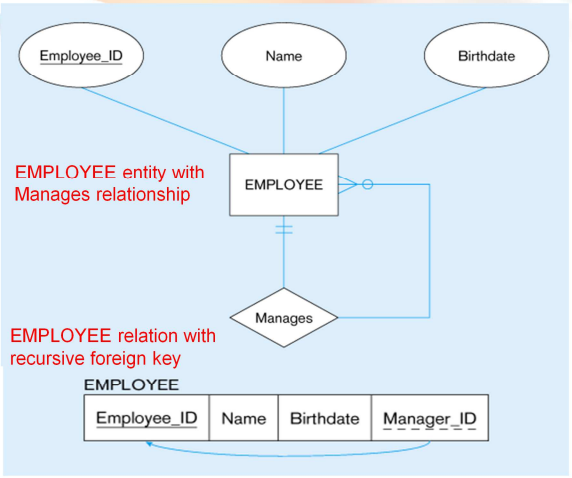
Convert Unary relationship one to one

ER Model: DB Design
Rule 3 - Example



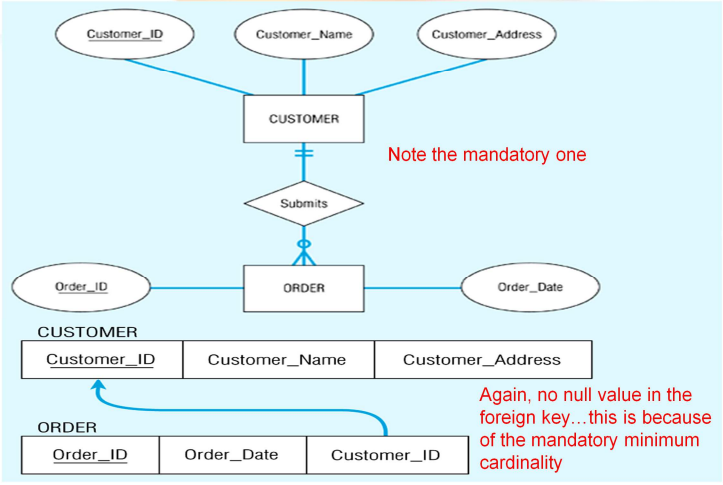
Convert binary relationship one to one

ER Model: DB Design
Rule 3 - Example



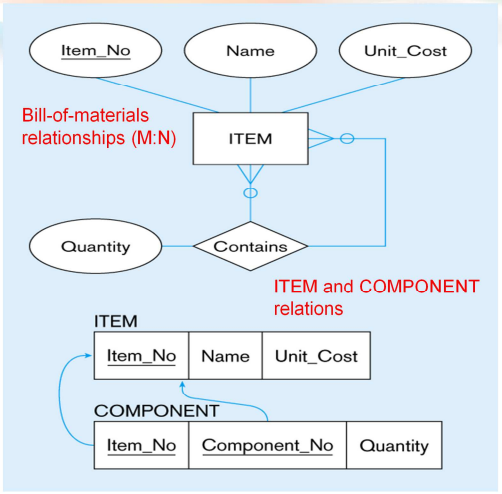
Convert Unary relationship one to many

ER Model: DB Design
Rule 3 - Example



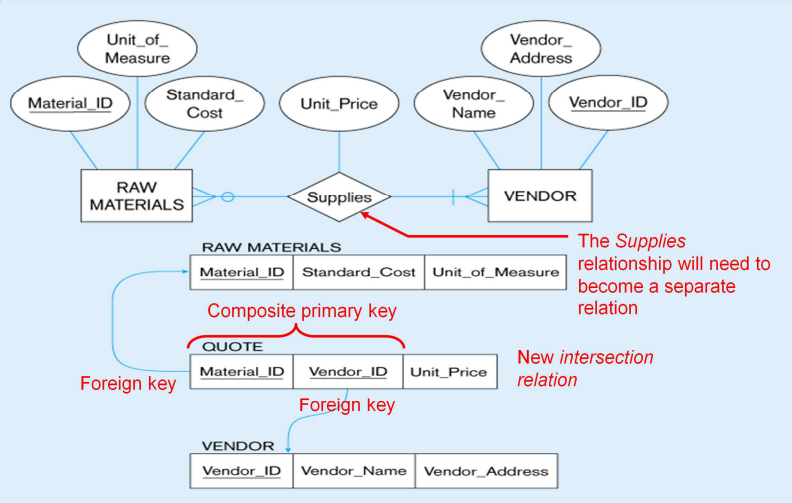
Convert Binary relationship one to many

ER Model: DB Design
Rule 3 - Example



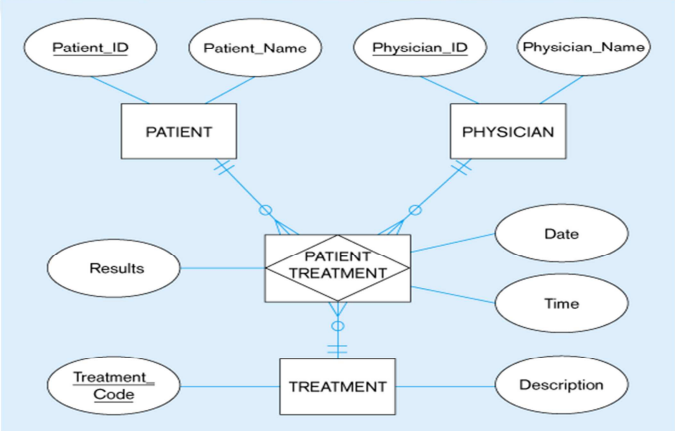
Convert Unary relationship many to many

ER Model: DB Design
Rule 3 - Example



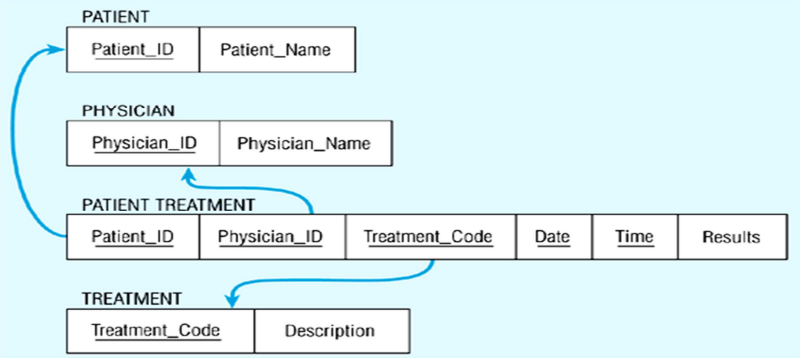
Convert Binary relationship many to many

ER Model: DB Design
Another Converting Example



Convert Ternary relationship

ER Model: DB Design
Another Converting Example



Convert Ternary relationship

Ms SQL Server Overview

- What is SQL?
- Ms SQL Server Overview
- Ms SQL Server Components

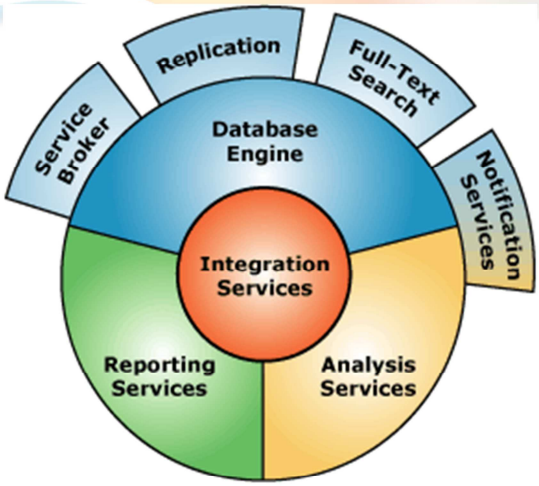
What is SQL?

- SQL stands for **S**tructured **Q**uery **L**anguage
- SQL allows access and manipulate databases
- SQL is an ANSI (American National Standards Institute) standard
- What Can SQL do?
  - Retrieve, insert, update, and delete database records
  - Create new or update database structures: table, view, stored procedures, etc.
  - Grant access permission to database objects: tables, stored procedures, views.

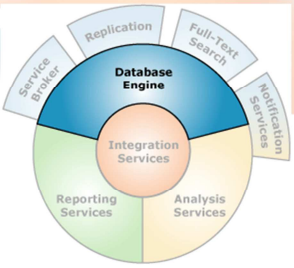
Ms SQL Server Overview

- Microsoft SQL Server is a database platform for:
  - Online Transaction Processing (OLTP)
  - Data warehousing, e-commerce applications
- It is also a business intelligence platform for:
  - Data integration, analysis, and reporting solutions.

Ms SQL Server Components



Database Engine



- Database Engine is the core service for storing, processing, and securing data
  - This includes creating tables for storing data, and database objects such as indexes, views, and stored procedures for viewing, managing, and securing data

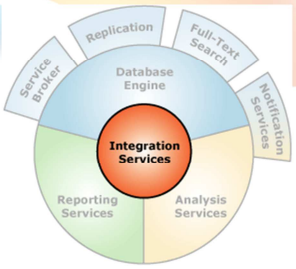
Reporting Service

- Reporting Services (SSRS) includes:
  - A complete set of tools that you can use to create and manage reports,
  - An application programming interface (API) that allows developers to integrate or extend data and report processing in custom applications

Replication

- Replication is a set of technologies for copying and distributing data and database objects from one database to another, and then synchronizing between databases to maintain consistency

Integration Services



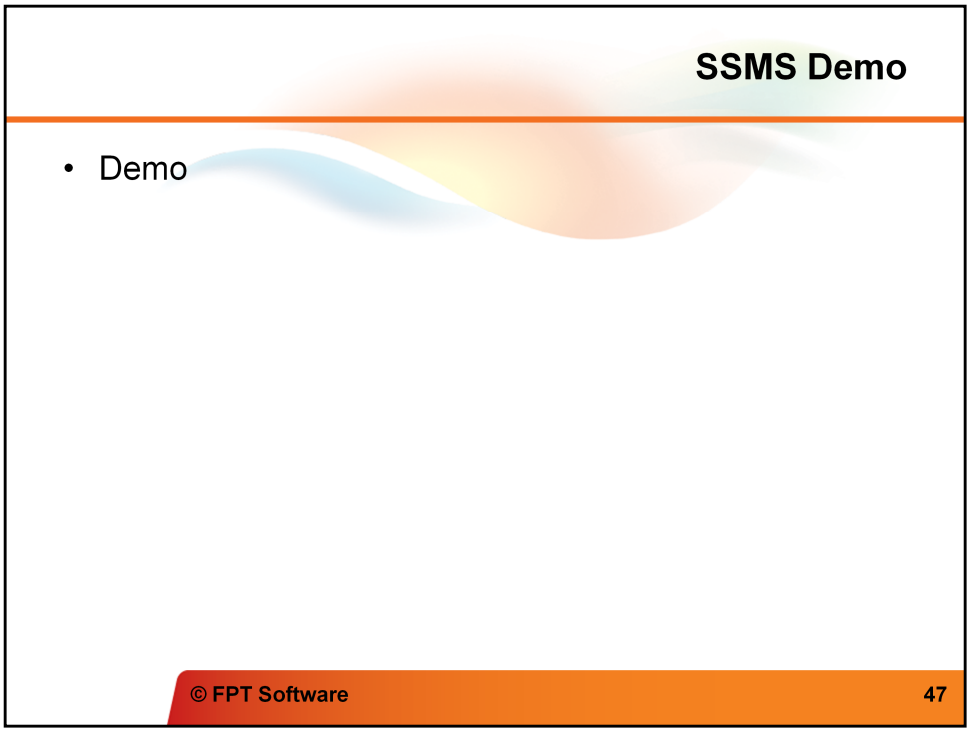
- Integration Services (SSIS) is a platform for building high performance data integration solutions, including extraction, transformation, and load packages for data warehousing

Ms SQL Server Tools

- SQL Server Management Studio
- SQL Profiler

SQL Server Management Studio

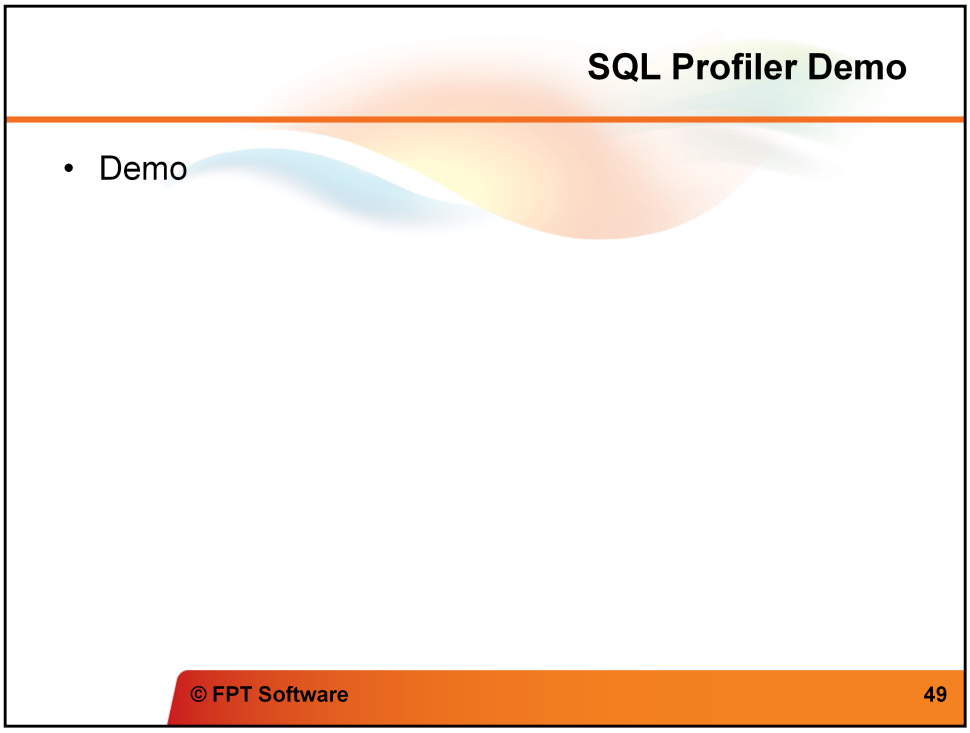
- **SQL Server Management Studio** (SSMS) is used for configuring, managing, and administering all components within Microsoft SQL Server
  - This includes both script editors and graphical tools which work with objects and features of the server



See SSMS_Demo.docx

SQL Profiler

- SQL Server Profiler is a tool that allows capture and analyze events, such as the execution of a stored procedure, occurring within SQL Server. This information can be used to identify and troubleshoot many SQL Server-related problems



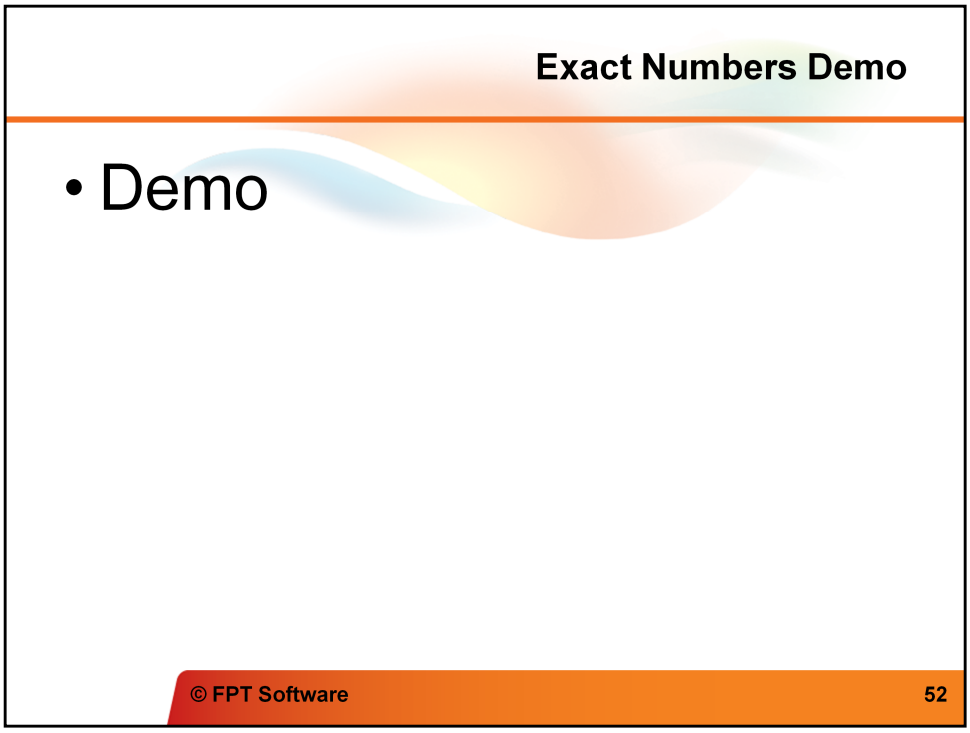
See SQLServerProfiler_Demo.docx

Ms SQL Server Data Types

- **SQL Server supports below data types. NULL is default value for most data type:**
  - Exact Numerics
  - Approximate Numerics
  - Date and Time
  - Character Strings
  - Unicode Character Strings
  - Binary Strings
  - Other Data Types

Exact Numbers

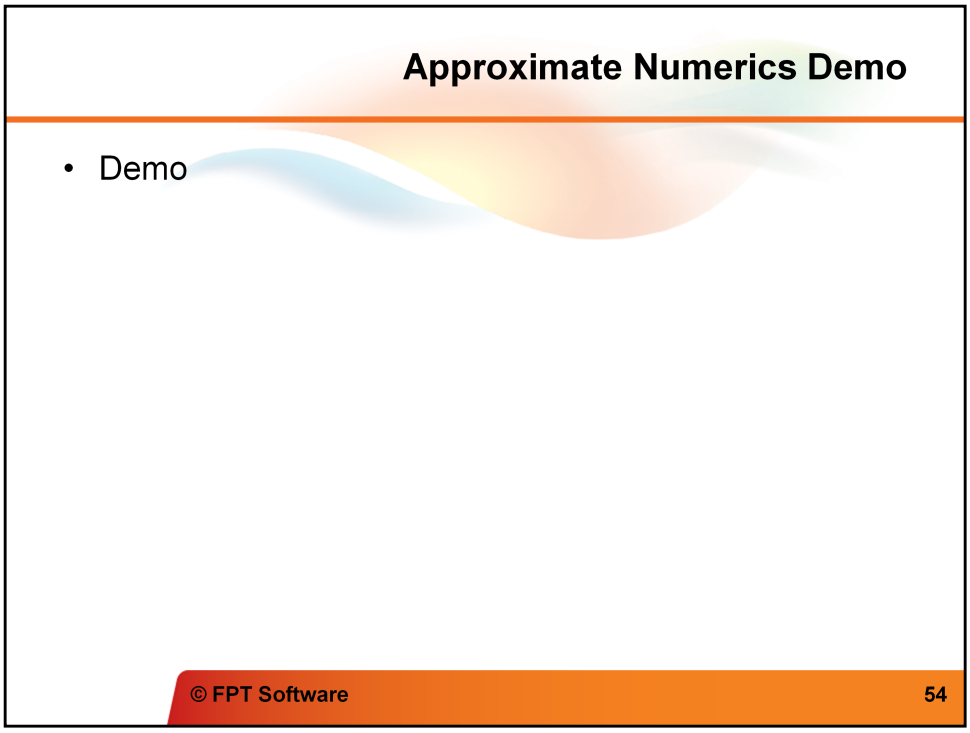
- Bigint: 8 Bytes.
  - Store integer data from -2^63 to 2^63-1
  - This is used in special case where the integer values exceed the range supported by the int data type.
- Int: 4 Bytes.
  - Store integer data from -2^31 to 2^31 - 1
  - This can be used to store primary key of a table.
- Smallint. 2 Bytes.
  - Store integer data from -2^15 to 2^15 - 1
- Tinyint: 1 Byte.
  - Store integer data from 0 through 255
  - This can be used to store primary key of table as Gender or Title
- Decimal: Numeric data types that have fixed precision and scale
- Numeric: The same as Decimal
- Bit: This can take a value of 1, 0, or NULL.
- Money: 8 Bytes. This should be used in special case, in normal case we should use decimal instead



Use Exact_DataType.sql file

Approximate Numerics

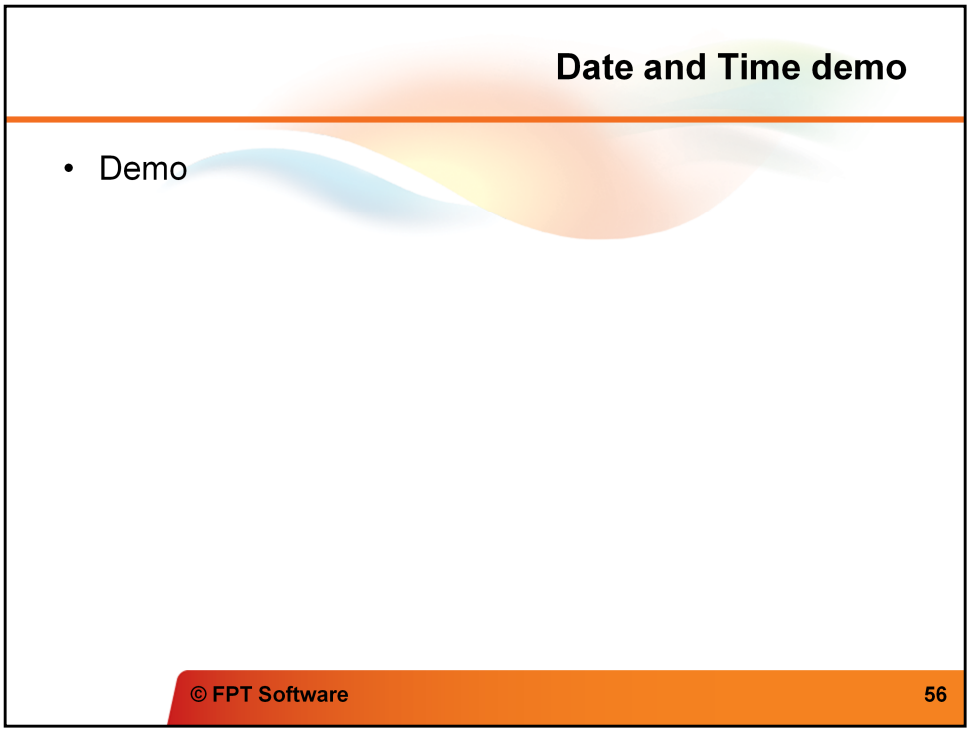
- Float
- Real



Use Approxiate_DataType.sql file

Date and Time

- SQL support below Date and Time data type:
  - Date
  - Datetimeoffset
  - Datetime2
  - Smalldatetime
  - Datetime
  - Time



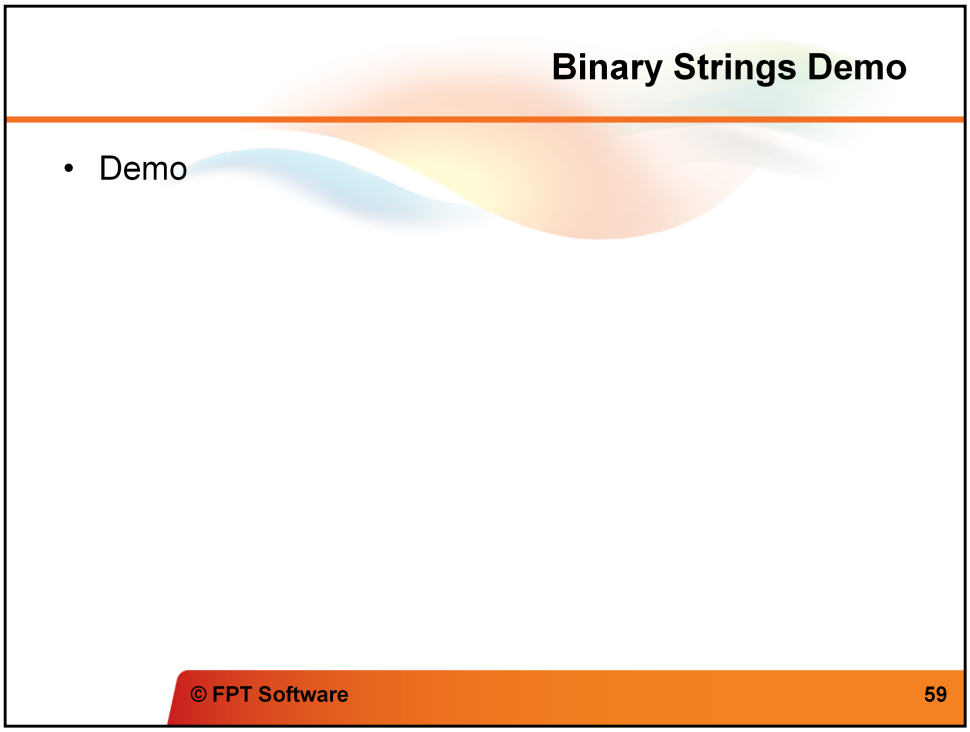
See Date_and_Time.sql

Character Strings

- Char(n): Fixed length, non-Unicode string data. 1<=n<=8000
- Varchar(n|max): Stores non-Unicode string data
  - n defines the string length and can be a value from 1 through 8,000
  - max indicates that the maximum storage size 2GB
- Nchar(n): Fixed length, Unicode string data
- Nvarchar(n|max): Stores Unicode string data

Binary Strings

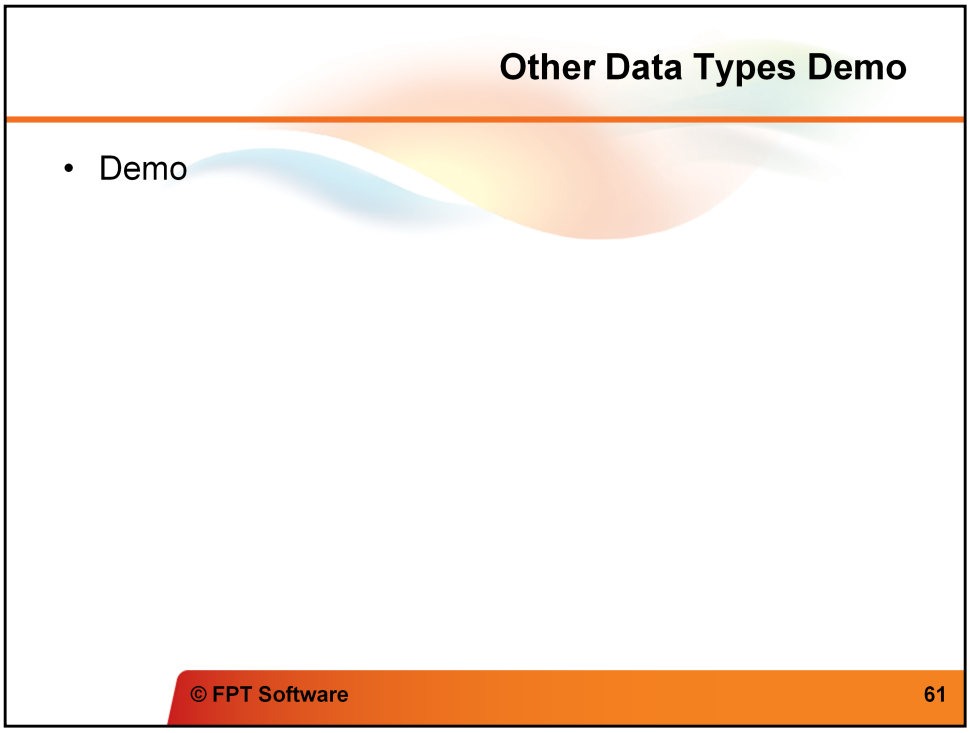
- Binary
- Varbinary
- Image



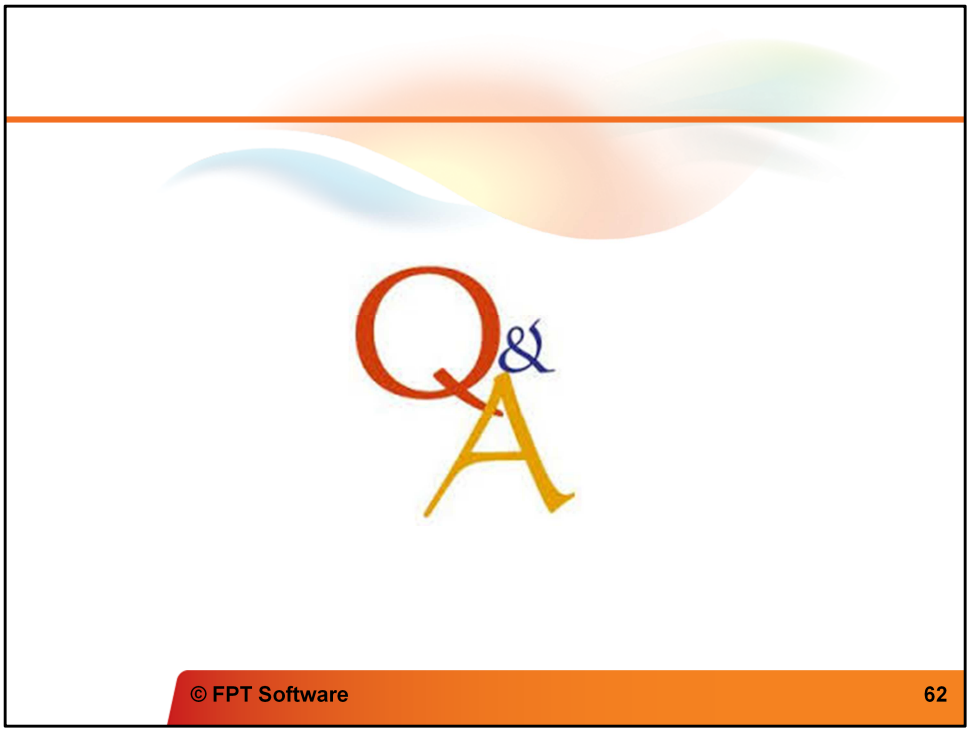
Use Binary_DataType.sql

Other Data Types

- timestamp
- hierarchyid
- uniqueidentifier
- sql_variant
- xml
- table



Use Other_DataType.sql


Q&A
© FPT Software
62