



# DBMS Lab

SQL(Structured Query Language)

By zelalem abera

[zolaab12@gmail.com](mailto:zolaab12@gmail.com)

# DBMS Lab Outline part 1

## introduction(2lab)

- ▢ Introduction to Database and DBMS
- ▢ Installing Microsoft server 2008
- ▢ Sql introduction
- ▢ Sql server introduction
- ▢ Connecting to sql server
- ▢ Creating and altering a database
  - i- Sql select database
  - II- Deleting or dropping a database
- ▢ Creating and working with tables
  - i-creating a table
  - ii-SQL drop or delete table

# DBMS Lab Outline part 2

## basic sql statements(2lab)

- ▮ SQL insert QUERY
- ▮ SQL select query
- ▮ Alter Table Statement
  - i-Adding a column
  - ii-deleting a column
  - iii-changing a data type of a column
- ▮ SQL auto increment
- ▮ Sql constraints
  - i-Foreign key constraints
  - ii-default constraint
  - iii-unique constraint
  - iv-check cinstraint

# DBMS Lab Outline part 3

## basic sql statements(1lab)

- SQL Distinct
- SQL where
- Sql and & or operators
- Sql order by
- Sql update
- Sql delete

# DBMS Lab Outline part 4

## Advanced sql statements(1lab)

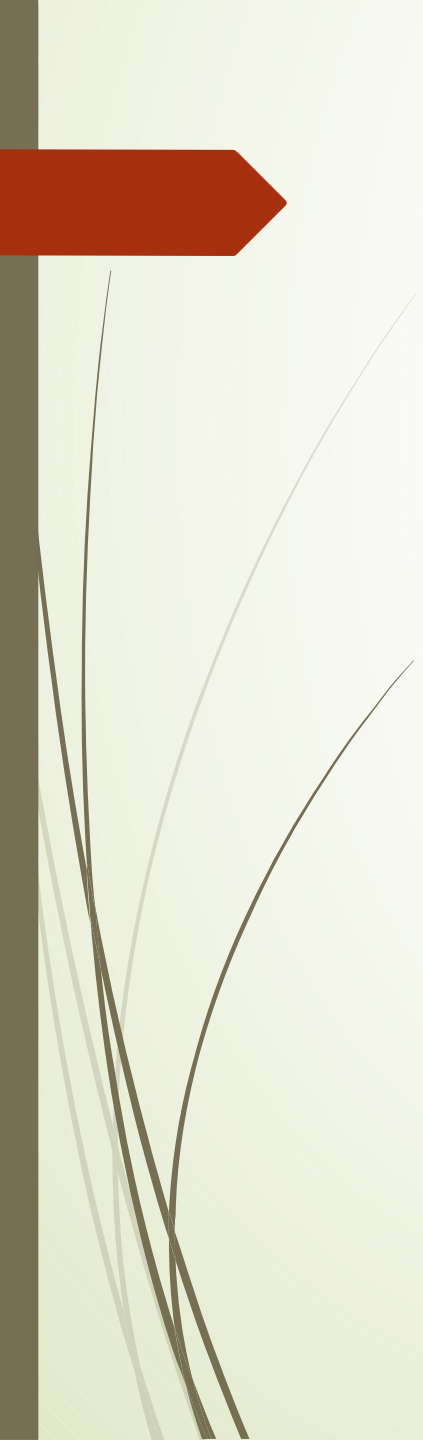
- ▢ SQL like
- ▢ SQL wild card
- ▢ Sql in
- ▢ Sql between
- ▢ Sql joins
  - i-inner join
  - ii-left join
  - iii-right join
  - iV-full join
- ▢ Sql union
- ▢ Sql views

## DBMS Lab Outline part 5

### sql functions(1lab)

- SQL avg()
- SQL count()
- SQL first()
- SQL last()
- SQL max()
- SQL min()
- SQL sum()
- SQL Group By
- SQL Having

.....MID exam.....

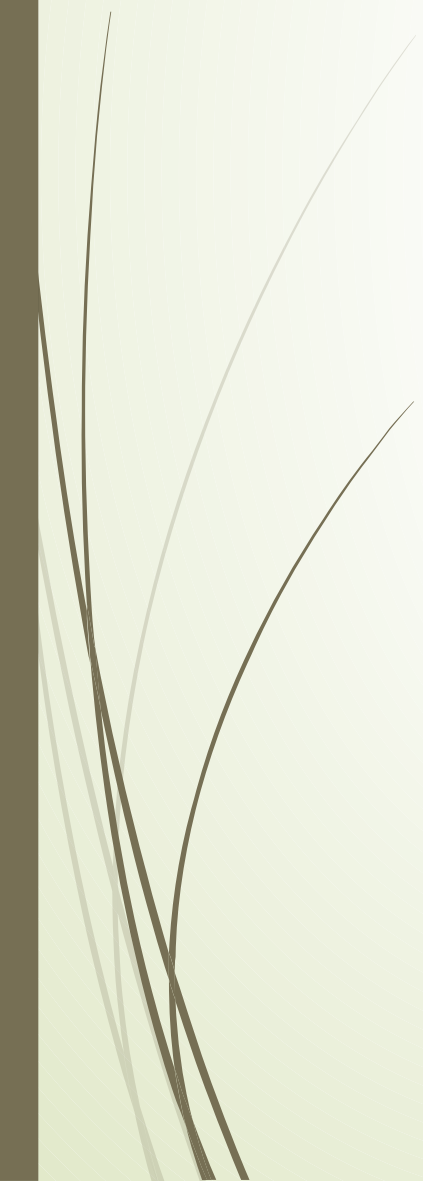


# **part 1**

## **introduction**

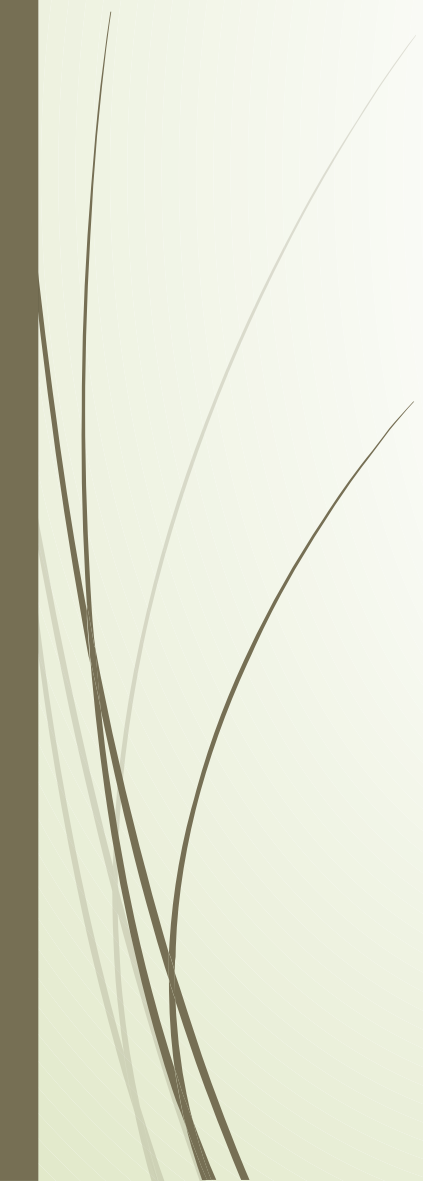


# What is Data ,information

- *Data*: -Meaningful facts, text, graphics, images, sound, video segments.
  - ▢ *Information*:- Data processed to be useful in decision making.
- 



# What is a Database?

- An organized collection of logically related data.
  - A collection of information that is organized so that it can *easily be accessed, managed and updated.*
- 



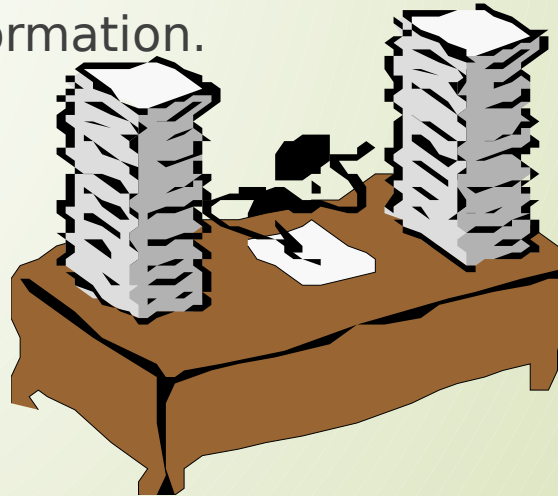
# Purpose of a Database

- The purpose of a database is:
    - To store data
    - To provide an organizational structure for data
    - To provide a mechanism for querying, creating, modifying, and deleting data
  - A database can store information and relationships that are more complicated than a simple list
- 

# Problem with lists

What is a list?

- List:-a simple two dimensional table of data, which we store information.



# Problems with lists

## Redundancy

- In a list , each row is intended to stand on its own as a result the same information may be entered several times.

| Project Manager | Manager ID | Phone Extension |
|-----------------|------------|-----------------|
| Dan             | 3          | x7270           |
| Dan             | 3          | x7270           |
| Dan             | 3          | x7270           |
| Cho             | 2          | x2114           |
| Dan             | 3          | x7270           |
| Dan             | 3          | x7270           |
| Rajesh          | 1          | x5754           |
| Dan             | 3          | x7270           |
| Dan             | 3          | x7270           |
| Dan             | 3          | x7270           |



# Problems with lists

## Multiple theme

In a list , **each row may contain information on more than one theme or business concept**. As a result certain information might appear in a list only if information about other themes or business concept is also present.

**Business concept***:-refers to a number of different attributes or properties that are all naturally related to one type of business entity.* Eg..Employee

# Problems with lists

## Multiple theme

| Project Manager | Manager ID | Phone Extension | Project Name     | Project ID | Start Date  | Budget  |
|-----------------|------------|-----------------|------------------|------------|-------------|---------|
| Dan             | 3          | x7270           | OS upgrade       | 101        | 01 Mar 2014 | \$12000 |
| Dan             | 3          | x7270           | SQL training     | 102        | 03 Apr 2014 | \$9000  |
| Dan             | 3          | x7270           | New tablet PCs   | 103        | 11 May 2014 | \$14200 |
| Cho             | 2          | x2114           | Website design   | 104        | 07 Jun 2014 | \$37450 |
| Dan             | 3          | x7270           | Cloud hosting    | 105        | 15 Jun 2014 | \$4650  |
| Dan             | 3          | x7270           | A.I. research    | 106        | 30 Jul 2014 | \$3000  |
| Rajesh          | 1          | x5754           | New web server   | 107        | 17 Aug 2015 | \$12500 |
| Dan             | 3          | x7270           | WiFi upgrade     | 108        | 03 Sep 2015 | \$9800  |
| Dan             | 3          | x7270           | Client training  | 109        | 22 Oct 2015 | \$2200  |
| Dan             | 3          | x7270           | DB upgrade       | 110        | 19 Nov 2015 | \$8700  |
| Dan             | 3          | x7270           | New email server | 111        | 09 Jan 2016 | \$11900 |



# List modification issues

- ▮ Redundancy and multiple themes in a list create modification problems.
- ▮ **Deletion problems**
- ▮ **Update problems**
- ▮ **Insertion problems**

# List Modification Issues

If Adviser Baker is changed to Taing, we need to change *AdviserEmail* as well. If changed to Valdez, we need to change *AdviserEmail*, *Department*, and *AdminLastName*.

|   | A        | B         | C                                                                                | D               | E                                                                              | F           | G             |
|---|----------|-----------|----------------------------------------------------------------------------------|-----------------|--------------------------------------------------------------------------------|-------------|---------------|
| 1 | LastName | FirstName | Email                                                                            | AdvisedLastName | AdvisedEmail                                                                   | Department  | AdminLastName |
| 2 | Andrews  | Matthew   | <a href="mailto:Matthew.Andrews@ourcampus.edu">Matthew.Andrews@ourcampus.edu</a> | Baker           | <a href="mailto:Linda.Baker@ourcampus.edu">Linda.Baker@ourcampus.edu</a>       | Accounting  | Smith         |
| 3 | Christen | Lisa      | <a href="mailto:Lisa.Christen@ourcampus.edu">Lisa.Christen@ourcampus.edu</a>     | Valdez          | <a href="mailto:Richard.Valdez@ourcampus.edu">Richard.Valdez@ourcampus.edu</a> | Chemistry   | Chaplin       |
| 4 | Fischer  | Douglas   | <a href="mailto:Douglas.Fischer@ourcampus.edu">Douglas.Fischer@ourcampus.edu</a> | Baker           | <a href="mailto:Linda.Baker@ourcampus.edu">Linda.Baker@ourcampus.edu</a>       | Accounting  | Smith         |
| 5 | Hwang    | Terry     | <a href="mailto:Terry.Hwang@ourcampus.edu">Terry.Hwang@ourcampus.edu</a>         | Taing           | <a href="mailto:Susan.Taing@ourcampus.edu">Susan.Taing@ourcampus.edu</a>       | Accounting  | Smith         |
| 6 | Lai      | Tzu       | <a href="mailto:Tzu.Lai@ourcampus.edu">Tzu.Lai@ourcampus.edu</a>                 | Valdez          | <a href="mailto:Richard.Valdez@ourcampus.edu">Richard.Valdez@ourcampus.edu</a> | Chemistry   | Chaplin       |
| 7 | Manno    | Chip      | <a href="mailto:Chip.Manno@ourcampus.edu">Chip.Manno@ourcampus.edu</a>           | Iran            | <a href="mailto:Ken.Iran@ourcampus.edu">Ken.Iran@ourcampus.edu</a>             | InfoSystems | Rogers        |
| 8 | Thompson | James     | <a href="mailto:James.Thompson@ourcampus.edu">James.Thompson@ourcampus.edu</a>   | Taing           | <a href="mailto:Susan.Taing@ourcampus.edu">Susan.Taing@ourcampus.edu</a>       | Accounting  | Smith         |
| 9 | ???      | ???       | ???                                                                              | ???             | ???                                                                            | Biology     | Kelly         |

Deleted row—Student, Adviser, and Department data lost

Inserted row—both Student and Adviser data missing



# Solution

- ▮ **Relational databases** are designed to address many of the information complexity issues that arise in business.

# What is a relational database?

- A Relational Database stores information in a tables. Each informational theme or business concept is stored in its own table.
- In essence, a relational database will break up a list into several parts
- One part for each theme in a list.

| EmployeeID | Name   | PhoneNumber  | DepartmentID |
|------------|--------|--------------|--------------|
| 101        | Dan    | 714-555-7270 | 1            |
| 102        | Fry    | 657-555-6561 | 2            |
| 103        | Leela  | 303-555-3247 | 2            |
| 104        | Bender | 480-555-4439 | 2            |

| DepartmentID | DepartmentName      |
|--------------|---------------------|
| 1            | Information Systems |
| 2            | Package Delivery    |



# Putting pieces back together

- In our relational database example ,we broke apart our table into several tables . somehow the tables must be joined back together.
- In relational databases tables are joined together using *matched pair of data values*.

# Putting pieces back together

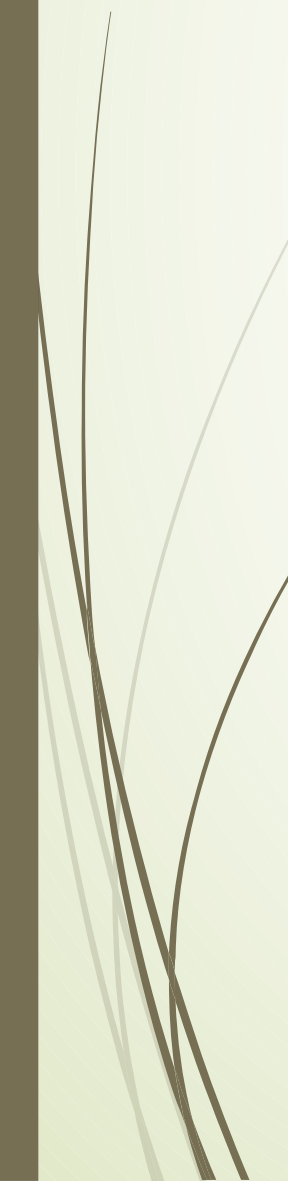
- For example: if a PROJECT has a customer the CUSTOMER\_ID can be stored in a column in a project table.
- *whenever we need information about the customer, we can use the CUSTOMER\_ID to look up the customer information in the customer table.*

| Project Name     | Project ID | Start Date  | Budget  | Customer ID |
|------------------|------------|-------------|---------|-------------|
| DB upgrade       | 110        | 19 Nov 2015 | \$8700  | 2           |
| New email server | 111        | 09 Jan 2016 | \$11900 | 2           |
| 3D printers      | 112        | 06 Jul 2016 | \$25000 | 2           |
| Network upgrade  | 113        | 24 Aug 2016 | \$33200 | 1           |

| Customer ID | Customer Name |
|-------------|---------------|
| 1           | Dan           |
| 2           | Priya         |



## Sounds like More Work, Not Less

- A relational database is more complicated than a list
  - However, a relational database minimizes data redundancy, preserves complex relationships among topics, and allows for partial data (*null* values)
  - Furthermore, a relational database provides a solid foundation for creating user interface forms and reports
- 



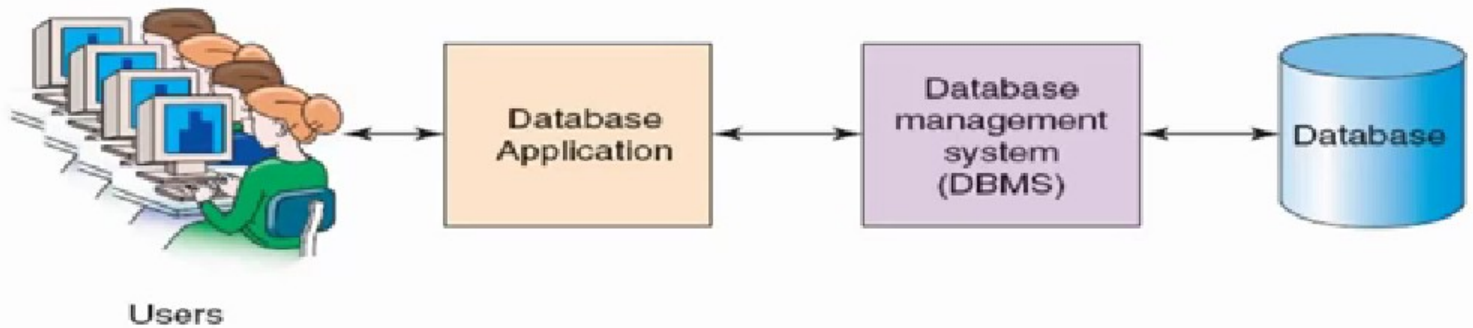
## The Structured Query Language (SQL)

- The Structured Query Language (SQL) is an international standard language for creating, processing, and querying databases and their tables
- The vast majority of data-driven applications and websites use SQL to retrieve, format, report, insert, delete, and/or modify data for users

# Database Systems

- The four components of a database system are:
  - Users
  - Database Application(s)
  - Database Management System (DBMS)
  - Database

# Components of a Database System





# Users

- A user of a database system will:
  - Use a database application to keep track of information
  - Use different user interface forms to enter, read, delete, and query data
  - Produce reports

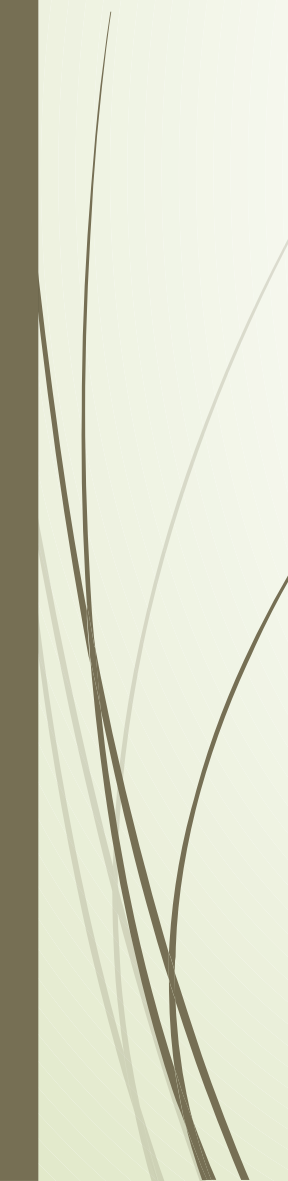


# The Database

- A database is a *self-describing* collection of *related* records
- Self-describing:
  - The database itself contains the definition of its structure
  - *Metadata* are data describing the structure of the data in the database
- Tables within a relational database are related to each other in some way

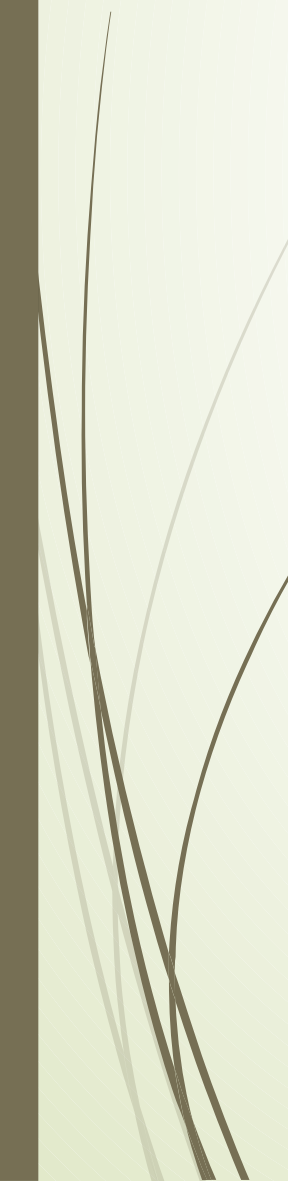


## Database Management System (DBMS)

- A database management system (DBMS) serves as an intermediary between database applications and the database
  - The DBMS manages and controls database activities
  - The DBMS creates, processes, and administers the databases it controls
- 



# Functions of a DBMS

- Create databases
  - Create tables
  - Create supporting structures
  - Read database data
  - Modify database data (insert, update, delete)
  - Maintain database structures
  - Enforce rules
  - Control concurrency
  - Provide security
  - Perform data backup and recovery
- 



# Commercial DBMS Products

- Example Desktop DBMS Products
  - Microsoft Access
- Example Organizational DBMS Products
  - SQL Server
  - Oracle
  - MySQL
  - DB2

# Referential Integrity Constraints

- A DBMS can enforce many constraints...
- Referential integrity constraints ensure that the values of a column in one table are valid based on the values in another table
  - For example, if a 5 was entered as a CustomerID in the PROJECT table, a Customer having a CustomerID value of 5 must exist in the CUSTOMER table

## Referential Integrity Constraints

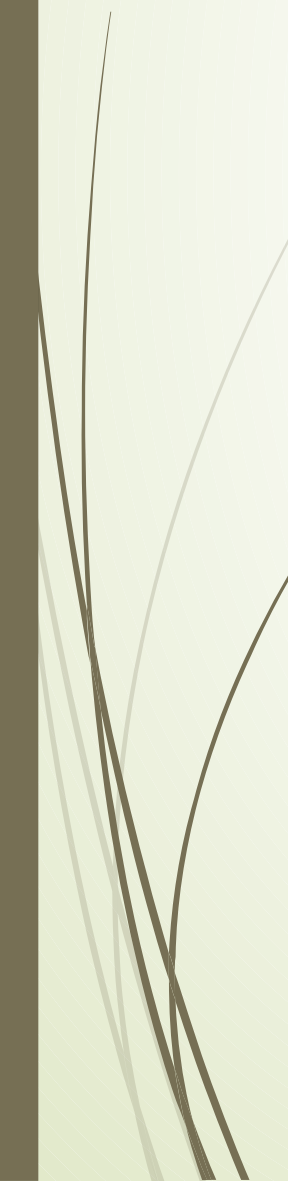
| Project Name     | Project ID | Start Date  | Budget  | Customer ID |
|------------------|------------|-------------|---------|-------------|
| DB upgrade       | 110        | 19 Nov 2015 | \$8700  | 2           |
| New email server | 111        | 09 Jan 2016 | \$11900 | 2           |
| 3D printers      | 112        | 06 Jul 2016 | \$25000 | 1           |
| Network upgrade  | 113        | 24 Aug 2016 | \$33200 | 5           |

| Customer ID | Customer Name |
|-------------|---------------|
| 1           | Dan           |
| 2           | Priya         |

Does Customer ID 5 exist?



# Database Applications

- A database application is a set of one or more computer programs or websites that serve as an intermediary between the user and the DBMS
- 

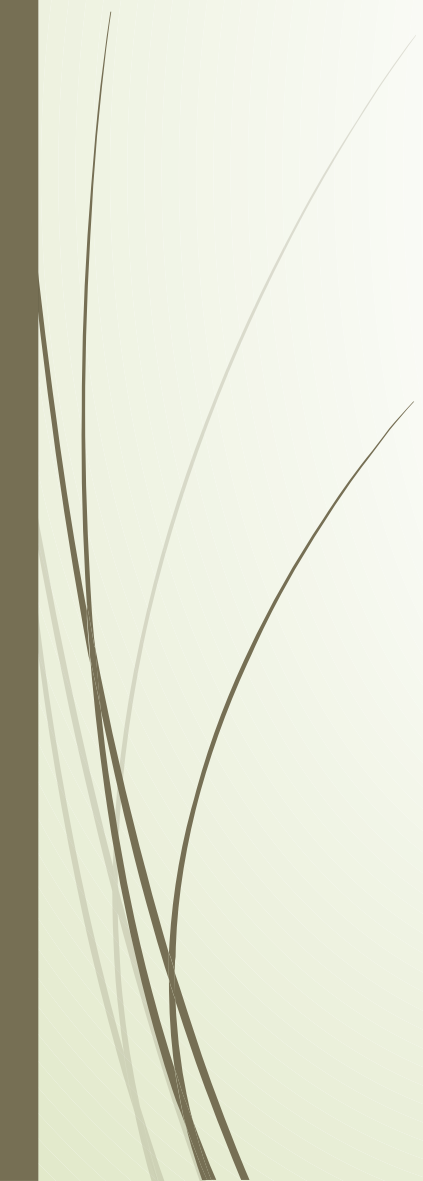


# quiz

- What are the three list modification issues?
- How do we address the problem of lists?
- What are the four components of a database system?
- What is a DBMS? functions of DBMS? Example of a DBMS?



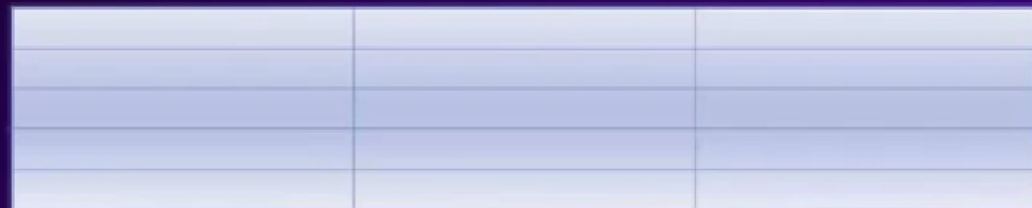
# Entity



- An entity represents on theme ,topic or business concept.
- Eg:- employee , department.....

# Relation

- A relation is a two-dimensional table that has specific characteristics
- The table dimensions, like a matrix, consist of rows and columns



|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



# Characteristics of a Relation

- Rows contain data about instances of an entity
- Columns contain data about attributes of the entity
- Cells of the table hold a single value
- All values in a column are of the same kind (data type)
- Each column has a unique name
- The order of the columns is unimportant
- The order of the rows is unimportant
- No two rows can be identical

# A Sample Relation

| EmployeeNumber | FirstName | LastName |
|----------------|-----------|----------|
| 100            | Yousef    | Hassan   |
| 101            | Jerry     | Cadley   |
| 104            | Yi        | Chang    |
| 107            | Shir      | Anavi    |



# A Non-relation Example

| EmployeeNumber | Phone                 | LastName |
|----------------|-----------------------|----------|
| 101            | 335-6421,<br>454-9744 | Hassan   |
| 100            | 215-7789              | Cadley   |
| 104            | 610-9850              | Chang    |
| 107            | 299-9090              | Anavi    |
| 100            | 215-7789              | Cadley   |



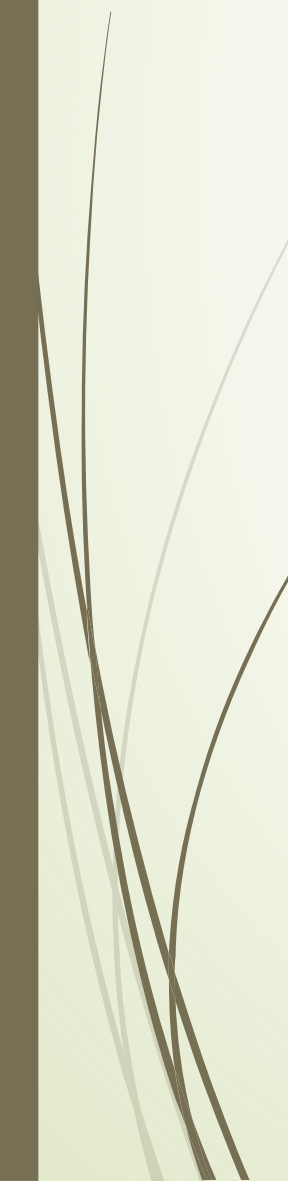
# Terminology

Synonyms...

|               |                |           |
|---------------|----------------|-----------|
| Table         | Row            | Column    |
| File (rarely) | Record         | Field     |
| Relation      | Tuple (rarely) | Attribute |



# Keys

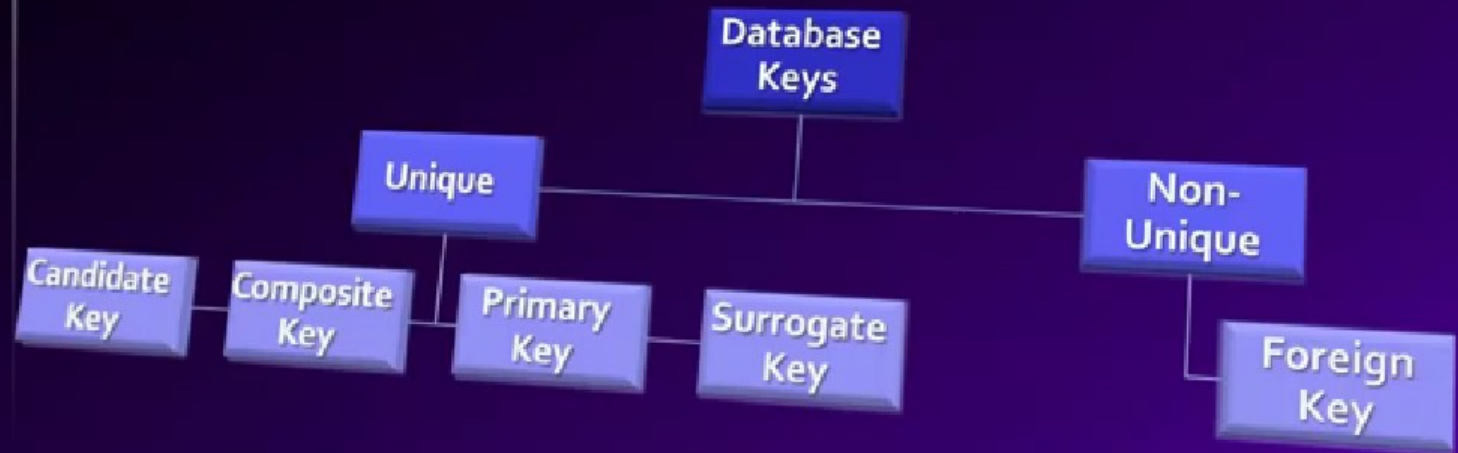
- A key is one (or more) columns of a relation whose values are used to identify a row
- 



# Uniqueness of Keys

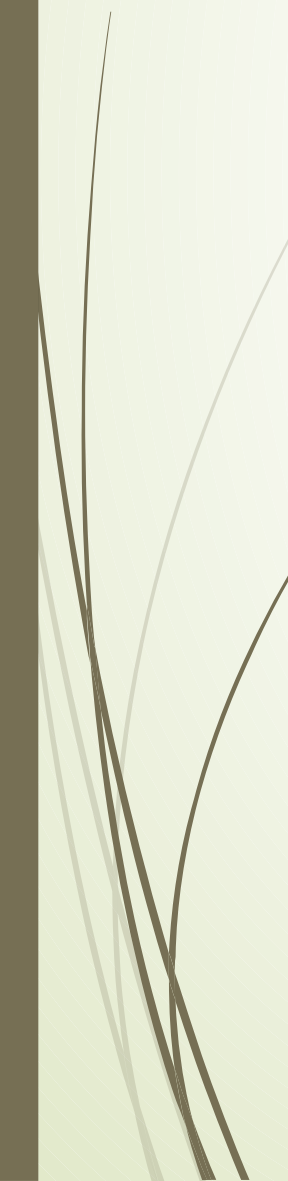
| Unique Key                                                                                | Nonunique Key                                                                                      |
|-------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| Data value is unique for each row.<br>Consequently, the key will uniquely identify a row. | Data value may be shared among several rows.<br>Consequently, the key will identify a set of rows. |

# Dan's Typology of Database Keys





## A Composite Key

- A composite key is a key that is composed of two or more attributes
  - For a key to be unique, it must often become a composite key
- 

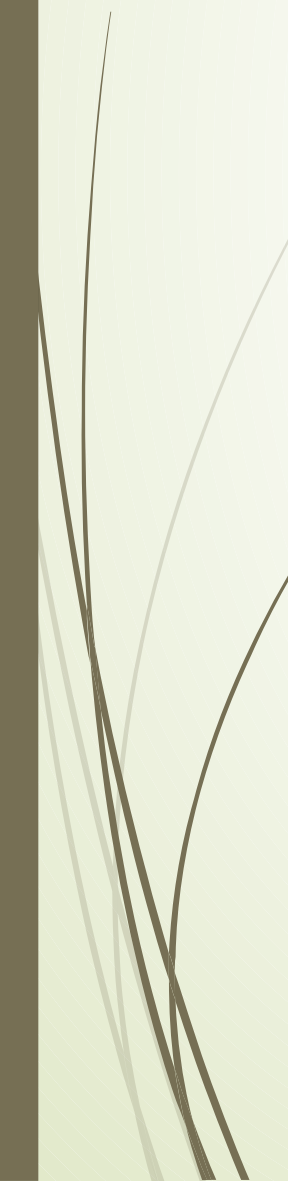
# Composite Key Example

| Flight Number | Date        |
|---------------|-------------|
| UA 36         | 03 Sep 2016 |
| AA 704        | 31 Oct 2016 |
| UA 36         | 04 Sep 2016 |
| BA 9          | 31 Oct 2016 |

**Flight Number + Date = Unique!**

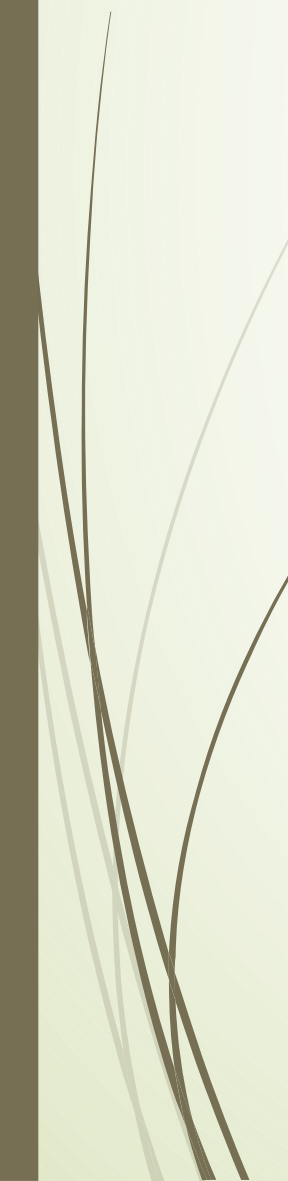


## A Candidate Key

- A candidate key is called “candidate” because it has the potential to become the primary key
  - A candidate key is a unique key
- 



## A Primary Key

- A primary key is a candidate key chosen to be the main key for the relation
  - If you know the value of the primary key, you will be able to uniquely identify a single row within the table
- 

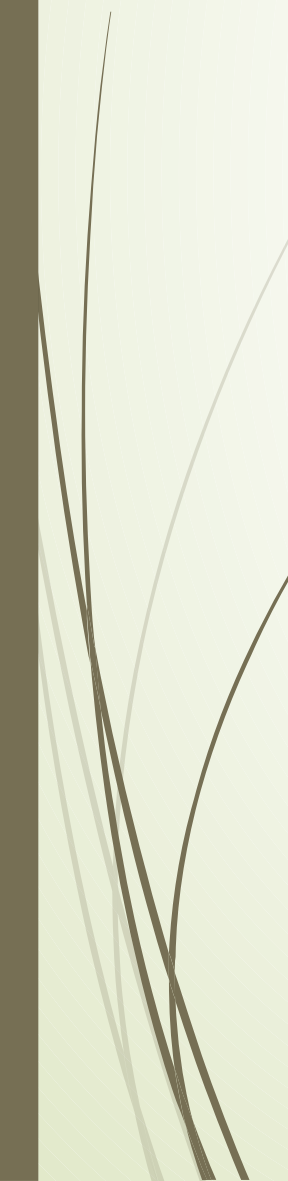


# Primary Key Example

| EmployeeNumber | FirstName | LastName |
|----------------|-----------|----------|
| 100            | Yousef    | Hassan   |
| 101            | Jerry     | Cadley   |
| 102            | Yi        | Chang    |
| 107            | Shir      | Anavi    |



## A Surrogate Key

- A surrogate key is a unique, numeric value that is added to a relation to serve as the primary key
  - Surrogate key values have no meaning to users and are usually hidden on forms, queries, and reports
  - A surrogate key is often used in place of a composite primary key
- 

# Primary Key Example



| EmployeeNumber | FirstName | LastName |
|----------------|-----------|----------|
| 100            | Yousef    | Hassan   |
| 101            | Jerry     | Cadley   |
| 102            | Yi        | Chang    |
| 107            | Shir      | Anavi    |

*EmployeeNumber is the primary key, but this type of primary key is called a Surrogate Key*



# Surrogate Key Example

| Flight ID | Flight Number | Date        |
|-----------|---------------|-------------|
| 1         | UA 36         | 03 Sep 2016 |
| 2         | AA 704        | 31 Oct 2016 |
| 3         | UA 36         | 04 Sep 2016 |
| 4         | BA 9          | 31 Oct 2016 |

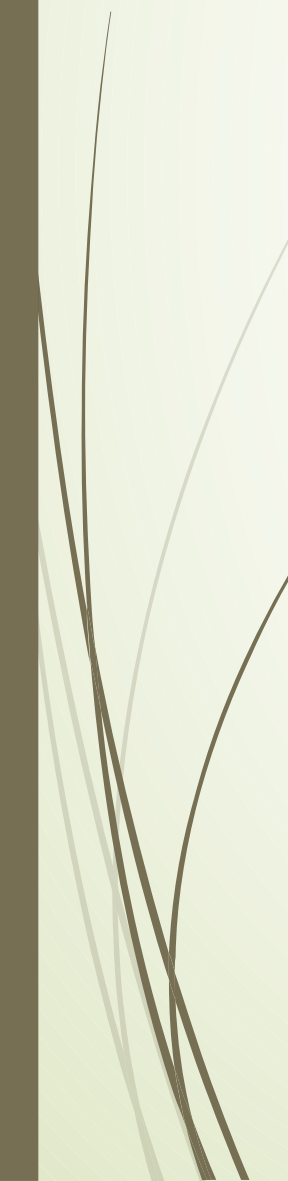


## Relationships Between Tables

- A table may be related to other tables
- For example
  - An Employee works in a Department
  - A Manager controls a Project



## A Foreign Key

- To establish relationships, we need to implement a foreign key
  - A foreign key is a primary key from one table that is placed into another table
  - The key is called a foreign key in the table that receives the key
- 

# Foreign Key Example 1

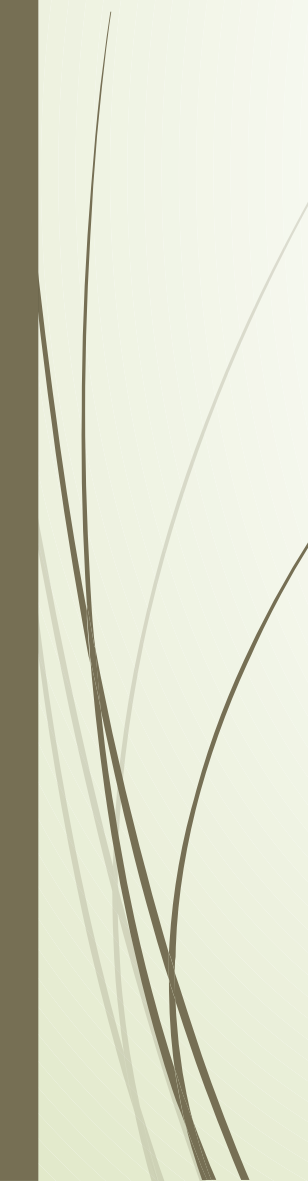
| Project      |
|--------------|
| Project ID   |
| Project Name |
| Manager ID   |

| Manager      |
|--------------|
| Manager ID   |
| Manager Name |





# Referential Integrity

- Referential integrity states that every value of a foreign key must match a value of an existing primary key
- 

# Referential Integrity Example

(...or lack thereof)

| Project Name     | Project ID | Start Date  | End Date | Customer ID |
|------------------|------------|-------------|----------|-------------|
| DB upgrade       | 110        | 19 Nov 2015 | \$8700   | 2           |
| New email server | 111        | 09 Jan 2016 | \$11900  | 2           |
| 3D printers      | 112        | 06 Jul 2016 | \$25000  | 1           |
| Network upgrade  | 113        | 24 Aug 2016 | \$33200  | 5           |

| Customer ID | Customer Name |
|-------------|---------------|
| 1           | Dan           |
| 2           | Priya         |

Does Customer ID #5 exist?



# quiz

- What is an entity?
- What are the two types of keys?
- How do we create relationships between tables?
- A foreign key is a primary key in another table.t/f



# Installing sql server 2008

...use the installation pdf



# What is SQL?

- ❑ SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational database.  
SQL is the standard language for Relation Database System.
- ❑ All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL Server use SQL as standard database language.



# Why SQL

- Allows users to access data in relational database management systems.
  - ¬∧ Allows users to describe the data.
  - ¬∧ Allows users to define the data in database and manipulate that data.
  - ¬∧ Allows to embed within other languages using SQL modules, libraries & pre-compilers.
  - ¬∧ Allows users to create and drop databases and tables.

## Objectives of SQL

- Can be used by range of users including **DBAs, management, application developers, and other types of end users.**
- **An ISO standard now exists for SQL,** making it both the formal and *de facto* standard language for relational databases.



## *Structured Query Language (SQL)*

- *The ANSI standard language for the definition and manipulation of relational database.*
- *Includes data definition language (DDL), statements that specify and modify database schemas.*
- *Includes a data manipulation language (DML), statements that manipulate database content.*



SQL Commands:

### DDL - Data Definition Language:

| Command | Description                                                                 |
|---------|-----------------------------------------------------------------------------|
| CREATE  | Creates a new table, a view of a table, or other object in database         |
| ALTER   | Modifies an existing database object, such as a table.                      |
| DROP    | Deletes an entire table, a view of a table or other object in the database. |

### DML - Data Manipulation Language:

| Command | Description      |
|---------|------------------|
| INSERT  | Creates a record |
| UPDATE  | Modifies records |
| DELETE  | Deletes records  |

### DCL - Data Control Language:

| Command | Description                             |
|---------|-----------------------------------------|
| GRANT   | Gives a privilege to user               |
| REVOKE  | Takes back privileges granted from user |

### DQL - Data Query Language:

| Command | Description                                       |
|---------|---------------------------------------------------|
| SELECT  | Retrieves certain records from one or more tables |



## *Some Facts on SQL*

- SQL data is case-sensitive, SQL commands are not.
- First Version was developed at IBM by Donald D. Chamberlin and Raymond F. Boyce. [SQL]
- Developed using Dr. E.F. Codd's paper, “A Relational Model of Data for Large Shared Data Banks.”
- SQL query includes references to tuples variables and the attributes of those variables

## Literals

- ▢ **Literals are constants used in SQL statements.**
- ▢ **All non-numeric literals must be enclosed in single quotes (e.g. 'London').**
- ▢ **All numeric literals must not be enclosed in quotes (e.g. 650.00).**

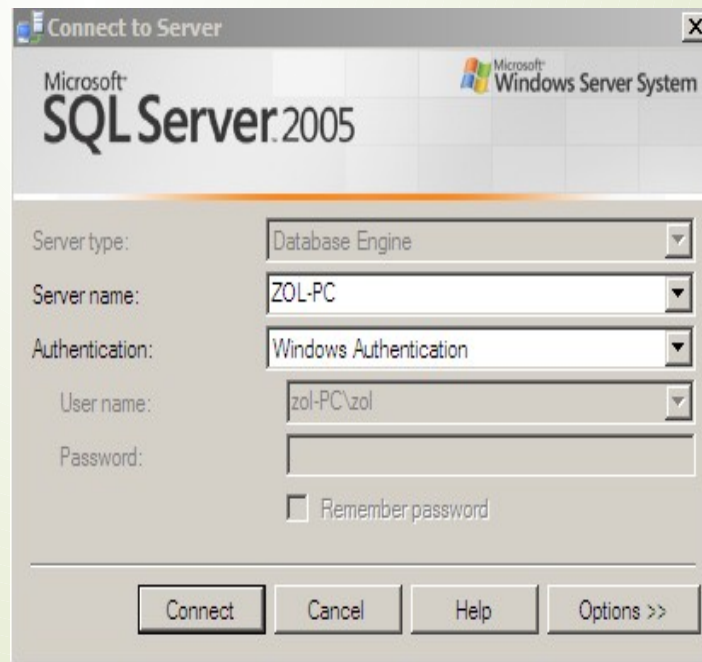
# What is Microsoft sql server?

- Is a relational database management system(RDBMS) developed by Microsoft.
- Like major RDBMS ,sql server supports ANSI SQL, the standard sql language ,however it also contains T\_SQL its own SQL implementation.
- SQL server management studio(SSMS) is sql servers main interface tool.

| Name                      | Number | Release Date |
|---------------------------|--------|--------------|
| Sybase                    | -      | -            |
| SQL Server 6.5            | 6.5    | 1996         |
| SQL Server 7              | 7.0    | 1998         |
| SQL Server 2000           | 8.0    | 2000         |
| SQL Server 2005           | 9.0    | 2005         |
| SQL Server 2008           | 10.0   | 2008         |
| SQL Server 2008 Release 2 | 10.5   | 2010         |

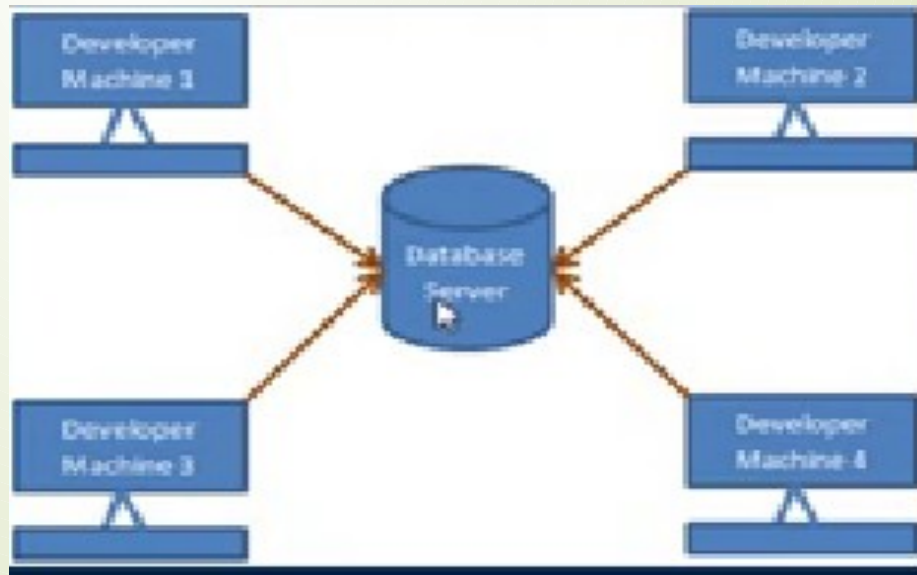
# Connecting to sql server

- ❑ You need to have sql server management studio already installed in your machine.
- ❑ To open the management studio click on start->all programs->sql server 2008/5-> select sql server management studio

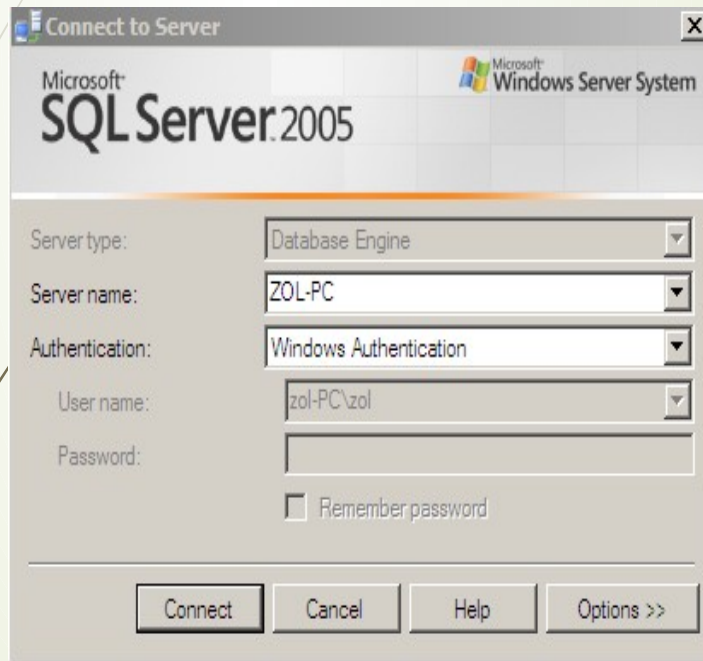


# Connecting to sql server

- The ssms and sql server database are two different things.
- Ssms:- is a **client tool which is used to connect to sql server**  
**and not the server by itself.**
- **Developer machines or clients connect to sql server using ssms**



# Connecting to sql server



- Server Type=Database Engine
- Server name=(local) or . Or 127.0.0.1(loopback ip address)
- Authentication=Windows or sql server
- If sql server authentication

Login

Pasword



# Creating and Altering a Database

- An Sql Server Database can be Created, Altered and Dropped
  - Using SSMS
  - Using a QUERY

# Creating and Altering a Database

- The SQL **CREATE DATABASE** statement is used to create new SQL database.

## Syntax:

Basic syntax of CREATE DATABASE statement is as follows:

```
CREATE DATABASE DatabaseName;
```

- Always database name should be unique within the RDBMS.

Example:

If you want to create new database <testDB>, then  
CREATE DATAB

```
SQL> CREATE DATABASE testDB;
```



# Creating a Database

- ❑ Create a database by the name **Sample\_1?**
- ❑ Create a database by the name **Sample\_2?**

# Creating and Altering a Database

- ❑ Altering a database
- ❑ To alter a database once its created
  - Alter database Databasename
  - Modify name=New Database Name
  - eg. Alter database hilco1 modify name=hilco



# Exercise 2

- ❑ Change the name of **Sample1** database to **hr1**.
- ❑ Change the name of **Sample2** database to **hr2**.



# SQL SELECT Database

- When you have multiple databases in your SQL Schema, then before starting your operation, you would need to select a database where all the operations would be performed. The SQL **USE** statement is used to select any existing database in SQL schema.

Syntax:

Basic syntax of USE statement is as follows:

- **Use DATABASE**



# Deleting or Dropping a Database

- To delete or Drop a Database use
- **Drop Database Databasename**  
eg. Drop Database hilco
- You cannot drop a database if its currently in use

**Note:-System databases cannot be dropped.**



# Exercise 3

- ❑ Delete database **hr1**?
- ❑ Delete database **hr2**?

# Creating and working with tables

- Creating a basic table involves naming the table and defining its columns and each column's data type.

The SQL **CREATE TABLE** statement is used to create a new table.

## Syntax:

Basic syntax of CREATE TABLE statement is as follows:

```
CREATE TABLE table_name(  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    .....  
    columnN datatype,  
    PRIMARY KEY( one or more columns )  
);
```

# Creating and working with tables

- You can create tables using SSMS or by executing query

Data types used when creating tables,

## Exact Numeric Data Types:

| DATA TYPE  | FROM                       | TO                        |
|------------|----------------------------|---------------------------|
| Bigint     | -9,223,372,036,854,775,808 | 9,223,372,036,854,775,807 |
| Int        | -2,147,483,648             | 2,147,483,647             |
| Smallint   | -32,768                    | 32,767                    |
| Tinyint    | 0                          | 255                       |
| Bit        | 0                          | 1                         |
| Decimal    | $-10^{38} + 1$             | $10^{38} - 1$             |
| Numeric    | $-10^{38} + 1$             | $10^{38} - 1$             |
| Money      | -922,337,203,685,477.5808  | +922,337,203,685,477.5807 |
| Smallmoney | -214,748.3648              | +214,748.3647             |

# Creating and working with tables

## Date and Time Data Types:

| DATA TYPE     | FROM                                 | TO           |
|---------------|--------------------------------------|--------------|
| Datetime      | Jan 1, 1753                          | Dec 31, 9999 |
| Smalldatetime | Jan 1, 1900                          | Jun 6, 2079  |
| Date          | Stores a date like June 30, 1991     |              |
| Time          | Stores a time of day like 12:30 P.M. |              |

**Note:** Here, datetime has 3.33 milliseconds accuracy where as smalldatetime has 1 minute accuracy.

## Character Strings Data Types:

| DATA TYPE    | FROM         | TO                                                                                        |
|--------------|--------------|-------------------------------------------------------------------------------------------|
| Char         | Char         | Maximum length of 8,000 characters.( Fixed length non-Unicode characters)                 |
| Varchar      | Varchar      | Maximum of 8,000 characters.(Variable-length non-Unicode data).                           |
| varchar(max) | varchar(max) | Maximum length of 231characters, Variable-length non-Unicode data (SQL Server 2005 only). |
| Text         | text         | Variable-length non-Unicode data with a maximum length of 2,147,483,647 characters.       |

# Creating and working with tables

□ We want to create the following two tables

Student

| Sid | sname | Email          | Aid  |
|-----|-------|----------------|------|
| 1   | Abebe | ab@gmail.com   | 3    |
| 2   | Girma | gir@gmail.com  | 2    |
| 3   | saba  | saba@gmail.com | null |

Advisor

| Aid | Aname     |
|-----|-----------|
| 1   | Dr Alemu  |
| 2   | Dr biruck |
| 3   | Dr gebre  |

# Creating and working with tables

- To create an a table we use a create table statement  
eg.

```
create table Advisor  
(  
Aid int not null primary key,  
sname varchar(50)  
)
```

This statement will create a table called advisor with a primary key column as Aid and a name column.

Note that if you don't specify the not null statement ,the default statement null will be executed, which means that the column can accept null values.

# Exercise 4

- Create a database with a name hilco
- Create the following two tables inside the **database hilco**?

| SID     | Int, primary key not null |
|---------|---------------------------|
| sname   | Text, not null1           |
| Email   | Text,null                 |
| advisor | Int null                  |

| Aid   | int, not null,primary key |
|-------|---------------------------|
| Aname | Text, not null            |



# Exercise 5

- ❑ Create a database named shop
- ❑ Create a table named groceries which have a primary key based on the following list
  - Bananas(4) aisle 2,
  - Butter(1) aisle 5,
  - Chocolate bars(2) aisle 6
- ❑ Change the database name to shopping

# SQL DROP or DELETE Table

- The SQL **DROP TABLE** statement is used to remove a table definition and all data, indexes, triggers, constraints, and permission specifications for that table.

- **NOTE:** You have to be careful while using this command because once a table is deleted then all the information available in the table would also be lost forever.

Syntax:

Basic syntax of DROP TABLE statement is as follows:

- **DROP TABLE** TABLE NAME



# Exercise 6

- ❑ Delete the table groceries found in shopping database.
- ❑ Delete the database shopping.