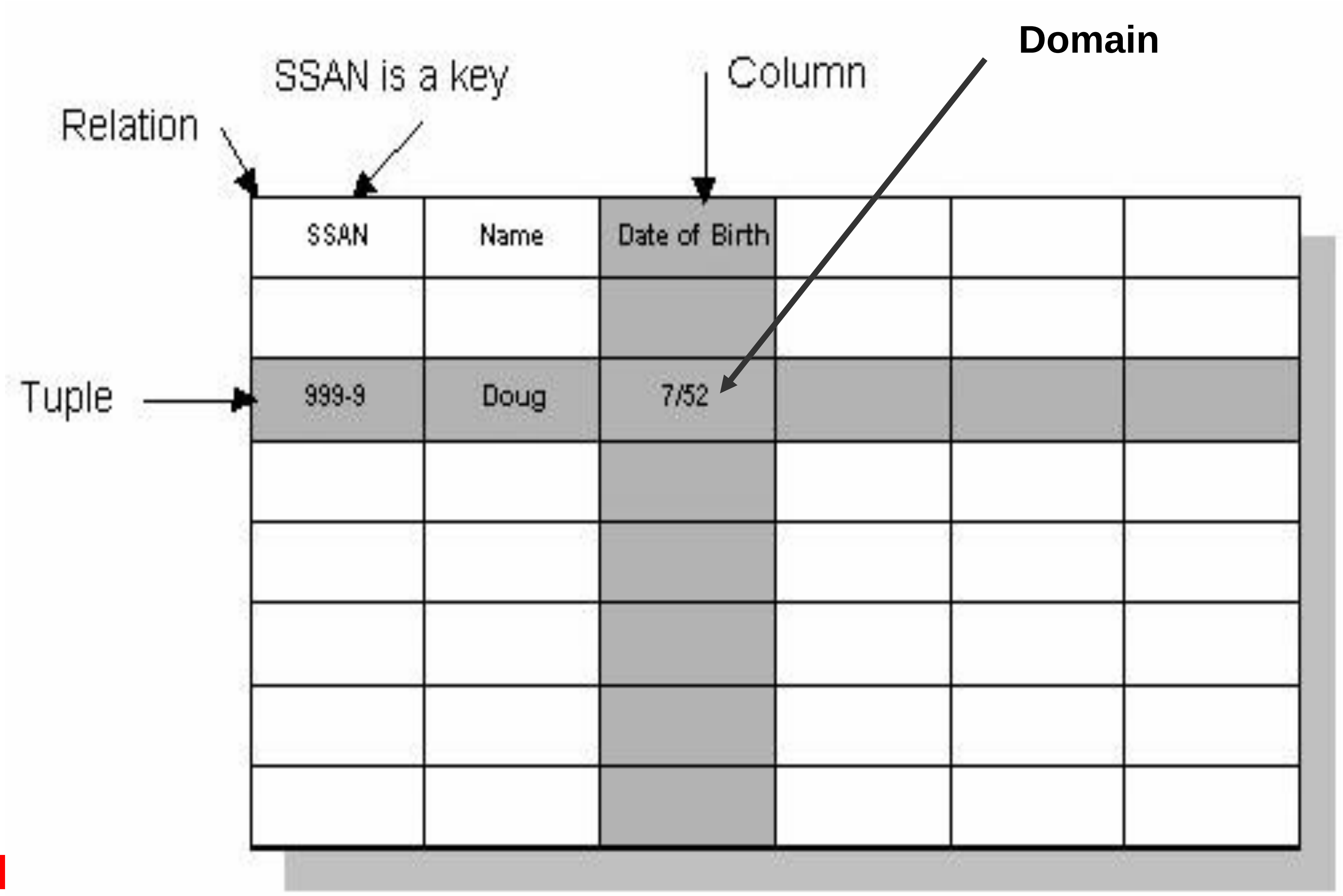
ITI
Database Fundamentals

Day 2

Mapping
SQL server
SQL (Select basic)

Relational Database



ER-to-Relational Mapping

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of Multi-valued attributes.

Step 7: Mapping of N-ary Relationship Types.

Step 1: Mapping of Regular Entity Types

- Create table for each entity type

- Choose one of key attributes to be the primary key

Step 2: Mapping of Weak Entity Types

- Create table for each weak entity.
- Add foreign key that correspond to the owner entity type.
- Choose the primary key : (FK + weak entity Partial PK if any)

Step 3: Mapping of Binary 1:1 Relation Types

- Merged two tables if both sides are Mandatory.
- Add FK into table with the total participation relationship to represent optional side.
- Create third table if both sides are optional.

Step 4: Mapping of Binary 1:N Relationship Types.

- Add FK to N-side table

- Add any simple attributes of relationship as column to N-side table.

Step 5: Mapping of Binary M:N Relationship Types.

- Create a new third table

- Add FKs to the new table for both parent tables

- Add simple attributes of relationship to the new table if any .

Step 6: Mapping of Multi-valued attributes.

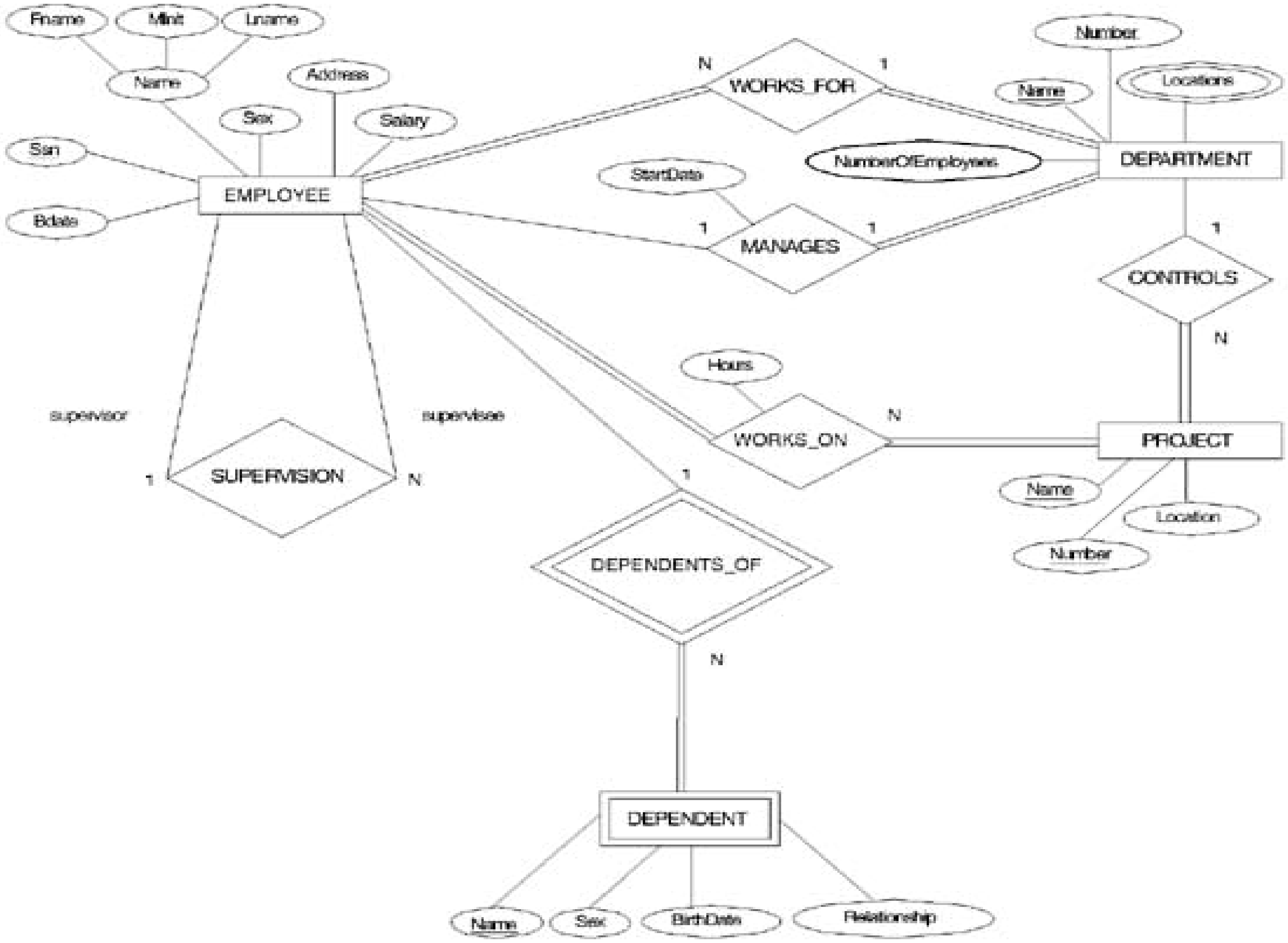
- Create new table for each multi-valued attribute

- Table will include two columns.

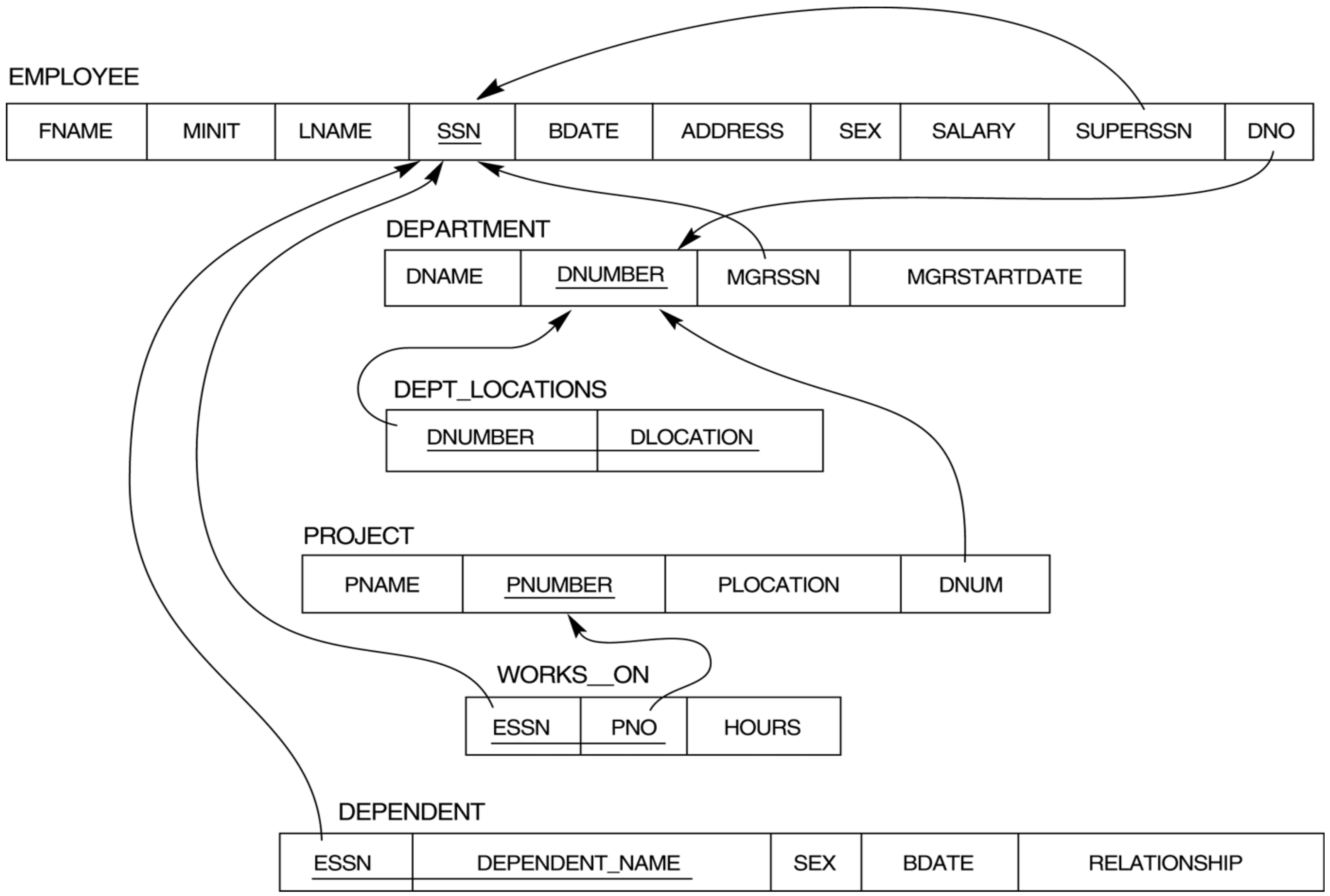
  one for multi-valued attribute + FK column.

Step 7: Mapping of N-ary Relationship Types.

- If n > 2 then :
- Create a new third table
- Add FKs to the new table for all parent tables
- Add simple attributes of relationship to the new table if any .


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
supervisor
supervisee
1
SUPERVISION
N
Hours
WORKS_ON
N
PROJECT
Name
Location
Number
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Mapping Result



Structured Query language
SQL

SQL Server authentication

- When SQL Server was installed, one of the decisions made was which of the following authentication
- methods to use:

■ Windows Authentication mode: Windows authentication only

■ Mixed mode: Both Windows authentication and SQL Server user authentication

SQL

- S-Q-L or sequel
- SQL (structured Query Language) is both an ANSI and ISO standard language based on the relational model, designed for querying and managing data in an RDBMS

T-SQL (Advanced SQL/ Transct -SQL)

- Microsoft’s and Sybase’s proprietary extension to SQL.
- T-SQL expands on the SQL standard to include:

1-Procedural Programming

2-Local Variables

3-various support functions for string processing

4-changes to the UPDATE and DELETE statements.

SQL Categories

- **DDL:** stands for **Data Definition Language**, deals with object definition and include statements such as CREATE, ALTER, DROP.
- **DML**: stands for **Data Manipulation Language** DML allows you to query and modify data and includes statements such as *SELECT, INSERT, UPDATE, DELETE.*
- **DCL:** stands for **data control language**, deals with permissions and includes statements such as *GRANT* and *REVOKE*

Database Transaction

- A transaction is an executing program that forms a logical unit of database actions.
- It includes one or more database access operations such as insert, delete and update.
- The database operations that form a transaction can either be embedded within an application program or they can be specified interactively via a high-level query language such as SQL.

Database Transaction Properties

- Transactions should possess several properties, often called the ACID properties:

  1. **Atomicity**
  2. **Consistency**
  3. **Isolation**
  4. **Durability or permanency**

Database Schema

A schema is a group of related objects in a database. There is one owner of a schema who has access to manipulate the structure of any object in the schema. A schema does not represent a person, although the schema is associated with a user that resides in the database.

Data types

A data type determines the type of data that can be stored in a database column. The most commonly used data types are:

1. Alphanumeric: data types used to store characters, numbers, special characters, or nearly any combination.
2. Numeric
3. Date and Time

Database Constraints

- Primary Key (Not Null + Unique)
- Not Null
- Unique Key
- Referential Integrity (FK)
- Check

Select statement

- The purpose of a *SELECT* statement is to query tables, and return a result.

```
SELECT [DISTINCT] [TOP (n)] *, columns, or expressions
[FROM data source(s)]
[into newtable]
[JOIN data source
ON condition](may include multiple joins)
[WHERE conditions]
[GROUP BY columns]
[HAVING conditions]
[ORDER BY Columns];
```

[Table Name]

- If the name of a database object, such as a table or column name, conflicts with a SQL reserved keyword, you can let SQL know that it’s the name of an object by placing it inside square brackets.
- Note that the square brackets [ ] are specific to SQL Server and not part of the ANSI SQL standard.

Selecting All Columns

- Example For Retrieving All Columns Of Table

```
Select *
From Employees
```

Selecting Specific Columns

- Example For Retrieving Specific Columns Of Table

```
Select  Employeeid, Salary, Address
From Employees
```

Column aliases

- Renames Column Heading
- Useful With Calculation
- Follows Column Name (Optional **As Keyword** Between CN and Alias)
- Require Double quotation If it contains spaces Or Special Characters Or Case Sensitive

SELECT **Employeeid, Salary as “Employee Salary”**

FROM Employees

Concatenation Operator

- Links Columns or Character Strings To Other Columns
- Represented By (+)
- Creates Resultant column That is Character Expression

Select EmployeeName + address

From Employees

Literal Column Strings

- Specify Own quotation Mark delimiter
- Choose delimiter
- Increase Readability

Select EmployeeName + ‘ The Address is ’   + address
From Employees

Select Distinct

- eliminates duplicate rows from the result set of the query.
- SELECT **DISTINCT Salary**
  From Employee

Using Arithmetic Operators

- SELECT Employeeid, Salary + 3000
  From Employee

Operators Precedence

- SELECT Employeeid,300+Salary*10 From Employee
- SELECT Employeeid,10*(Salary+3000) From Employee

Defining Null Value

- A null is value that is unavailable,unknown,unassigned
- A null is not zero or blank space

- Select Employeeid , MobileNo
  From Employee

Null Values In Arithmatic Expressions

- Select Employeeid ,Salary * 10
  From Employee

Where Conditions

- The WHERE conditions filter the output of the FROM clause and restrict the rows that will be returned in the result set. The conditions can refer to the data within the tables, expressions, built-in SQL Server scalar functions, or user-defined functions.

Using NULL Condition

- Select Employeename,MobileNo
  From Employee
  Where Mobileno is NULL

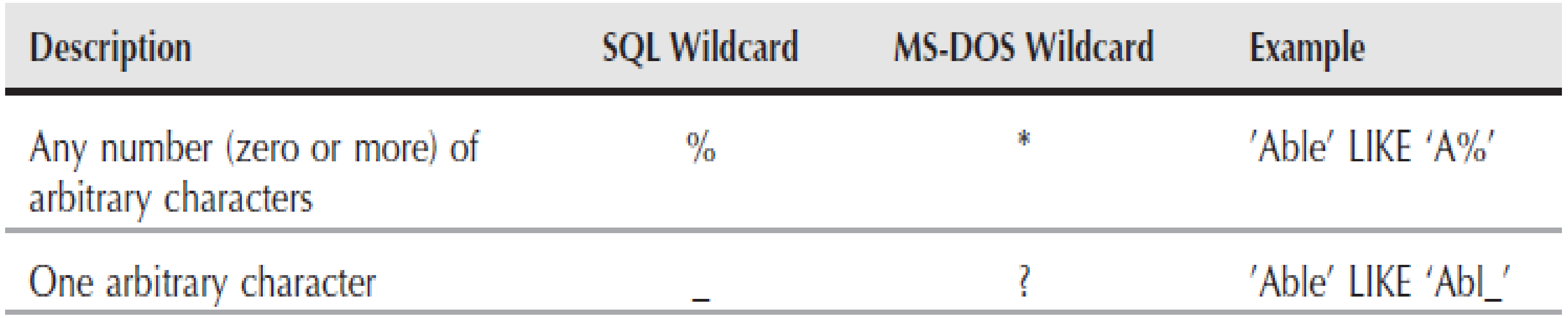
Comparison operator in Where Clause

Description	Operator	Example
Equals	=	Quantity = 12
Greater than	>	Quantity > 12
Greater than or equal to	>=	Quantity >= 12
Less than	<	Quantity < 12
Less than or equal to	<=	Quantity<= 12
Not equal to	<> , !=	Quantity <> 12 , Quantity != 12
Not less than	!<	Quantity !< 12
Not greater than	!>	Quantity !> 12

```
Select fname + ‘ ‘ + lname, salary
  From Employee
  Where Salary >=1000
```

Like operator in where clause

- The LIKE search condition uses **wildcards** to search for patterns within a string.



Description	SQL Wildcard	MS-DOS Wildcard	Example
Any number (zero or more) of arbitrary characters	%	*	'Able' LIKE 'A%'
One arbitrary character	_	?	'Able' LIKE 'Abl_'

```
SELECT fname
FROM dbo.employee
WHERE fname LIKE 'A%';
```