رواد مصر الرقمية

وزارة الاتصالات
وتكنولوجيا المعلومات

Module 2

Introduction to T-SQL Querying

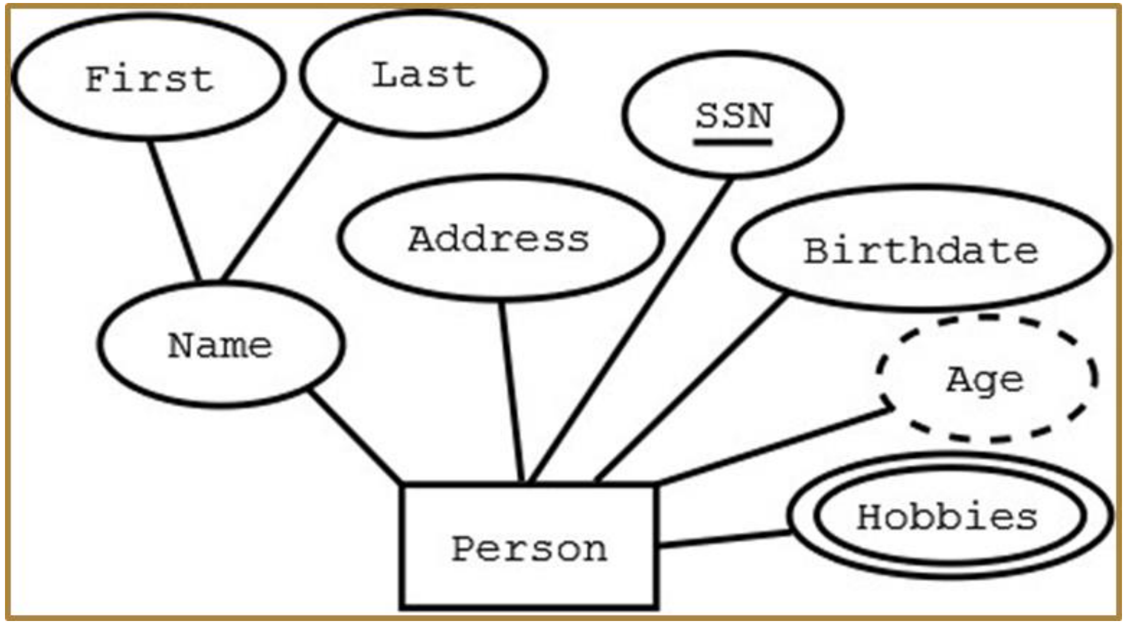
Session 2

Module Overview

- Introducing ERD
- Introducing T-SQL
- Understanding Sets
- Understanding Predicate Logic
- Understanding the Logical Order of Operations in SELECT Statements

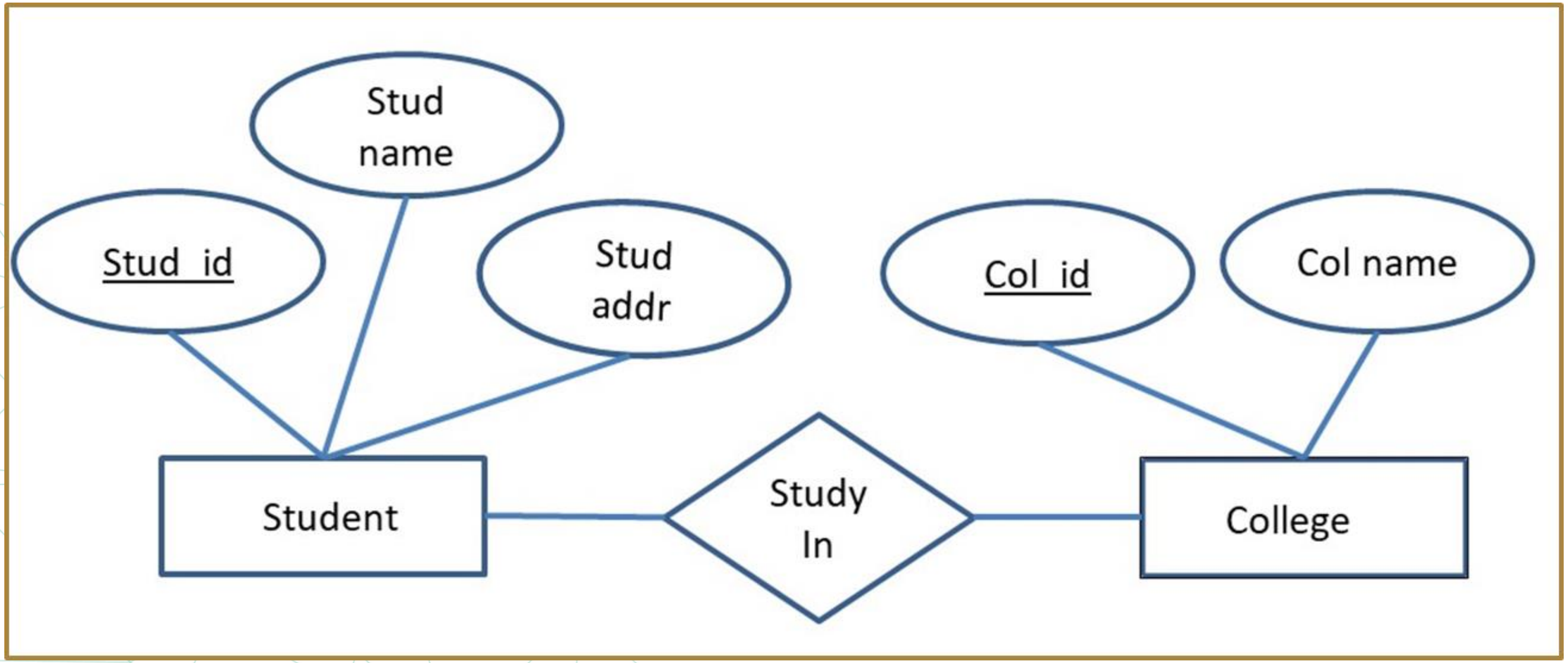
Entity Relationship Diagram (ERD)

- It identifies information required by the business by displaying the relevant Entities and Relationships between them.
- Entity: it’s a thing in the real world with independent existence and can be described with a set of attributes.



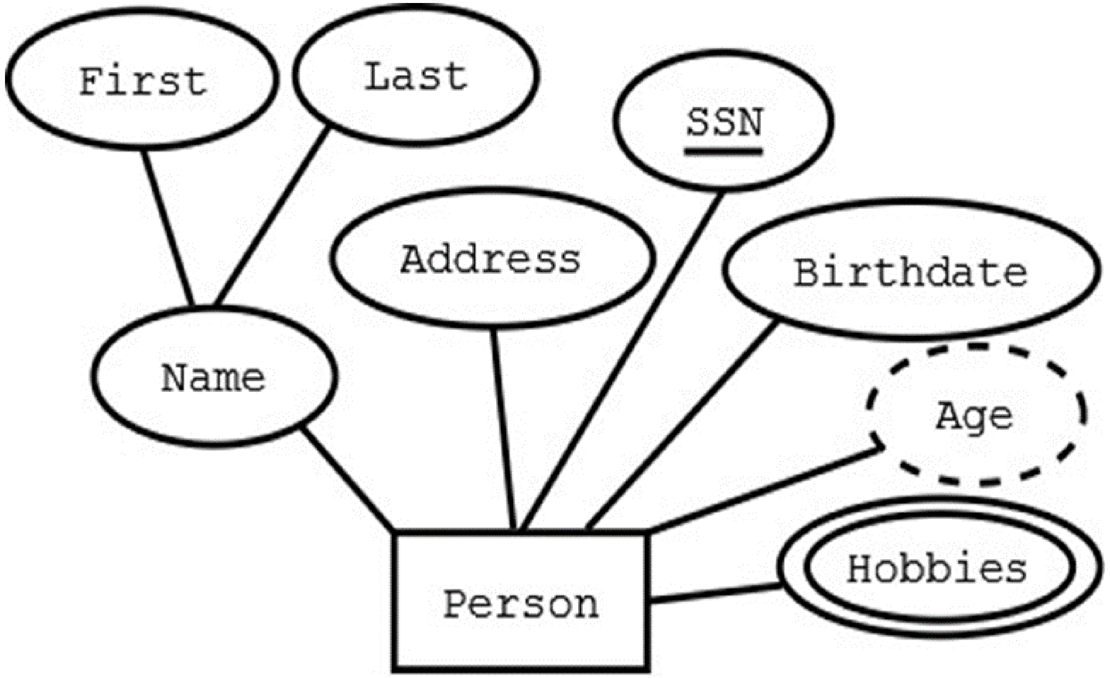
Entity Relationship Diagram (ERD)

- There should be always relationships between entities and there shouldn’t be an entity without any relationship.



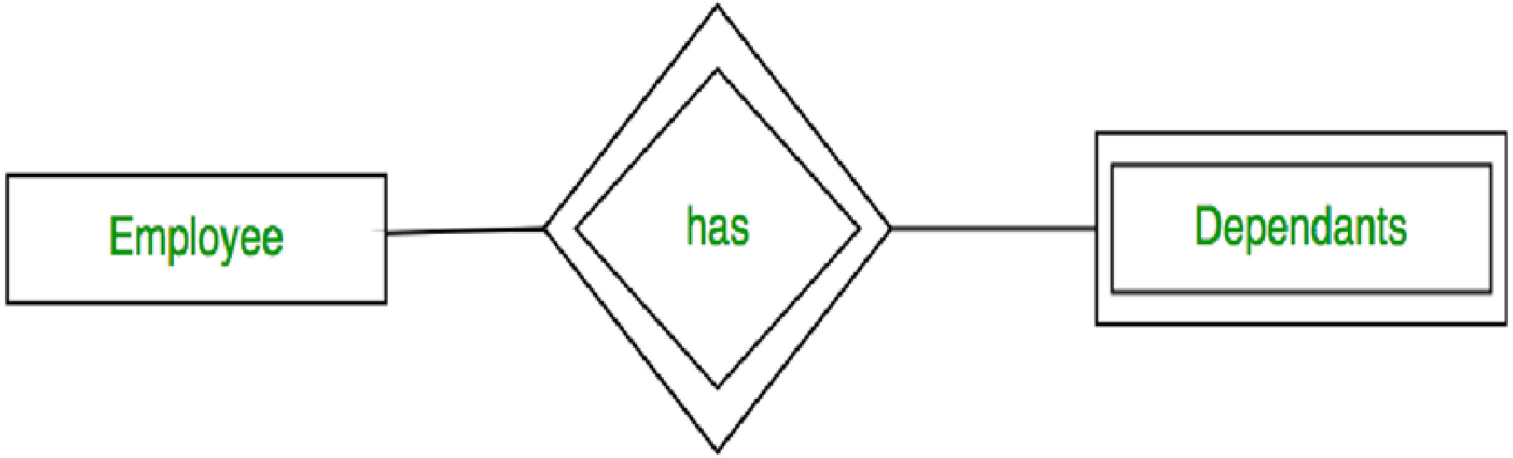
Attributes

- Single or simple Attributes. (Ex: ID, SSN).
- Multi Valued Attribute (Ex: Hobbies). 3- Composite Attribute (Ex: Name).
- Derived Attribute (Ex: Age) can be calculated from another attribute.
- Candidate Keys: the unique identifiers attributes for each entity (SSN)



Entity TYPES

- **Strong Entity**: It has a unique identifier or a key attribute. It’s represented in a single rectangle.
- **Weak Entity**: It doesn’t have a key attribute. It must be fully dependent on another entity. It’s represented in a double rectangle.



Relationships

- For each relationship, We need to identify:

Degree of relationship

It's the number of participating entities.

1-Binary Relationship (Two Entities)
2-Unary Relationship (Entity and itself)
3-Ternary Relationship (Three Entities)
4-N-ary Relationship

Cardinality

It specifies the maximum number of relationships.

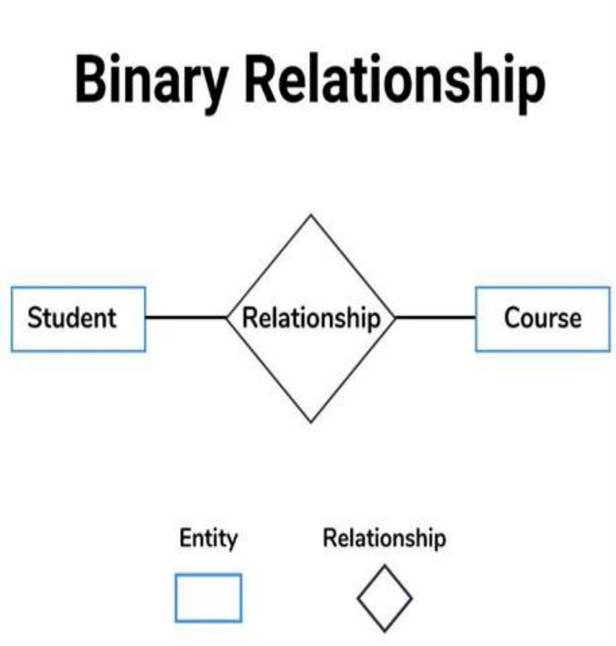
1 to 1
1 to Many
Many to Many
Many to 1

Participation

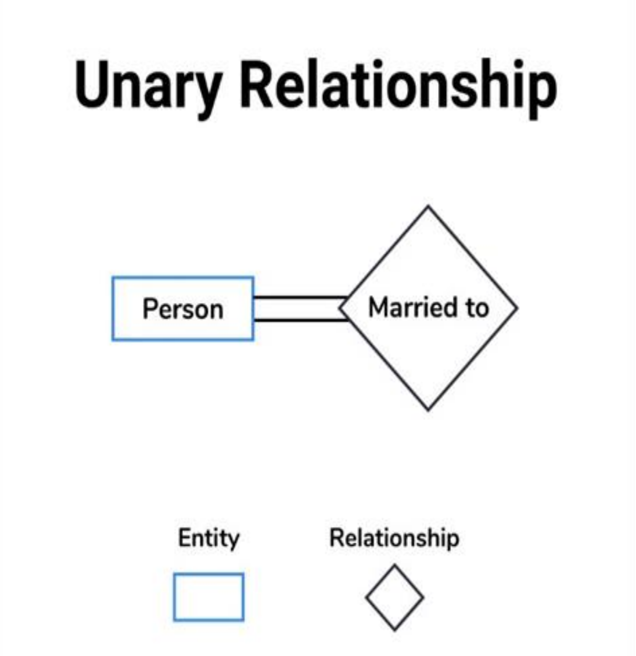
It specifies the minimum number of relationships instances that each entity can participate with.

Must (Double Lines)
May (Single Line)

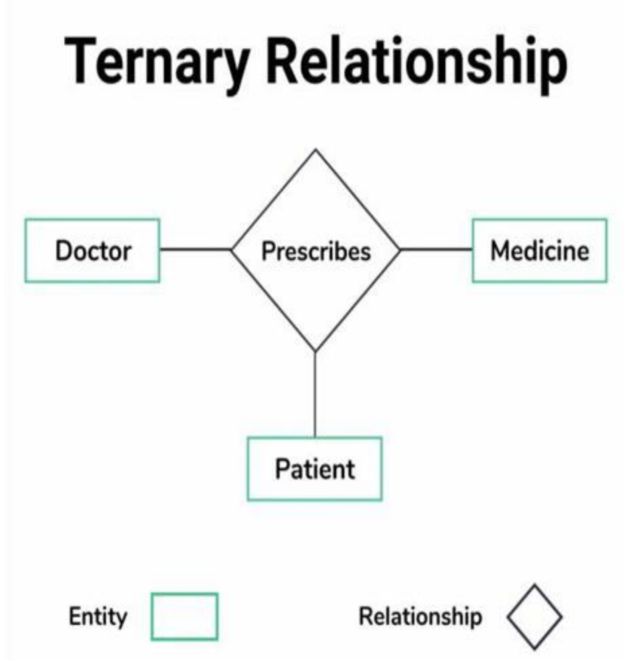
Degree of relationship



Binary Relationship



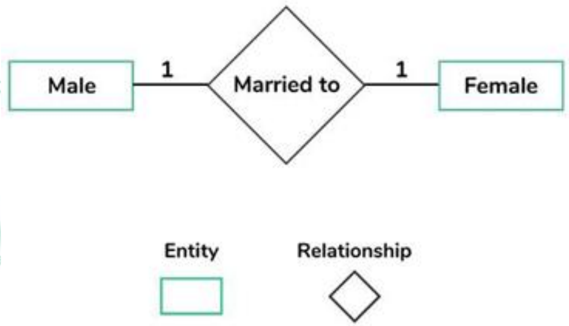
Unary Relationship



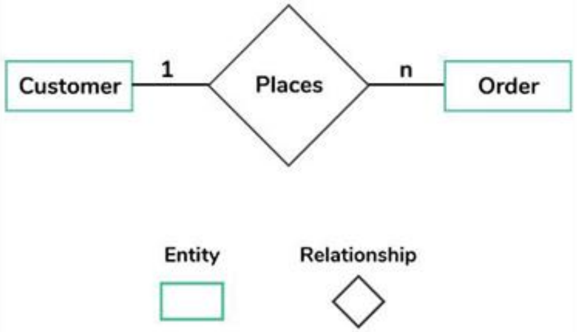
Ternary Relationship

Cardinality

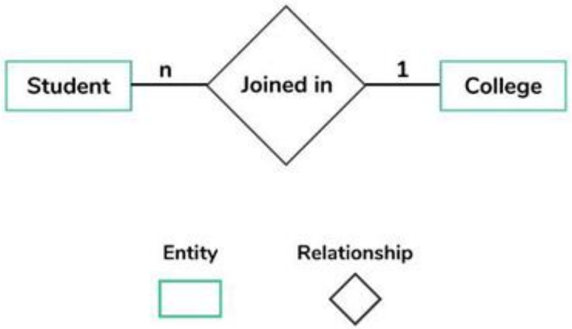
One to One Cardinality



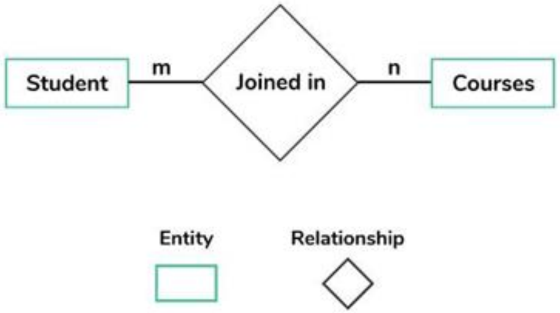
One to Many Cardinality



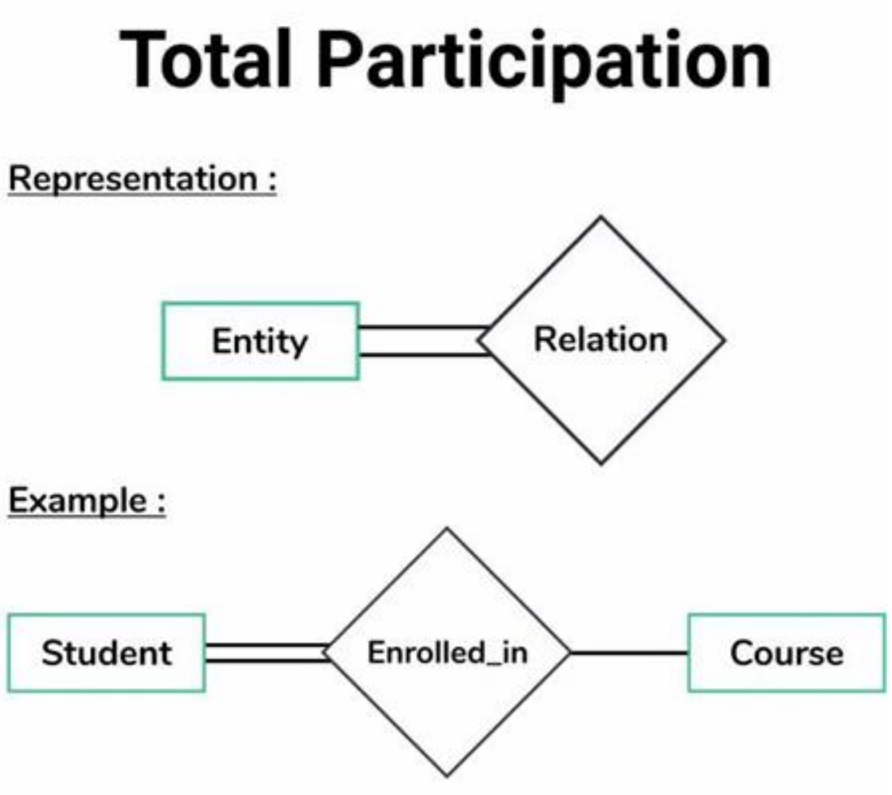
Many to One Cardinality



Many to Many Cardinality



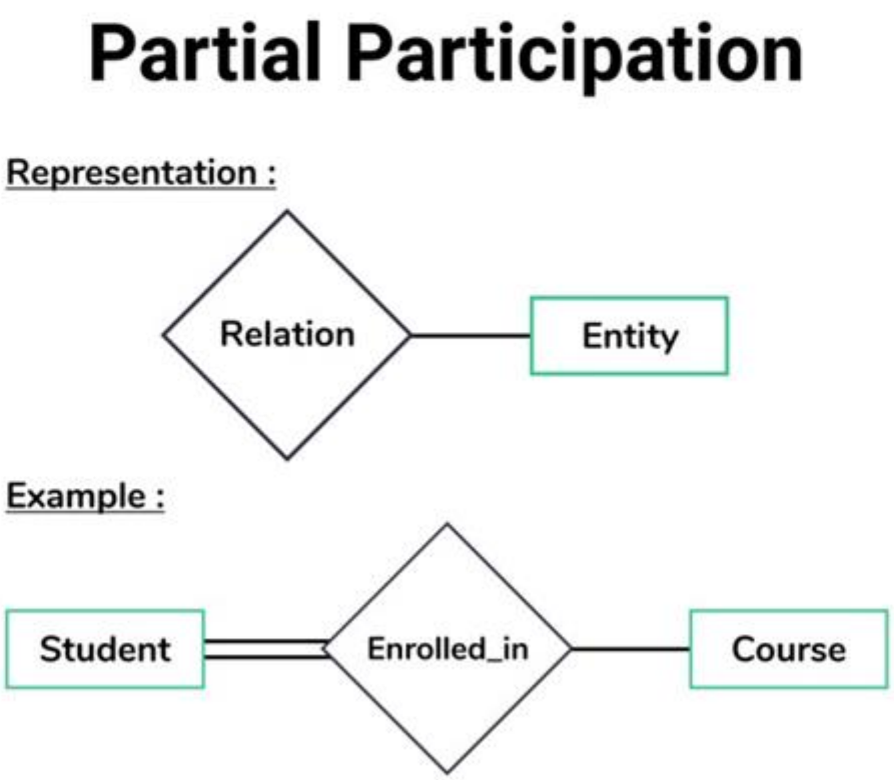
Participation



Total Participation

Representation :

Example :



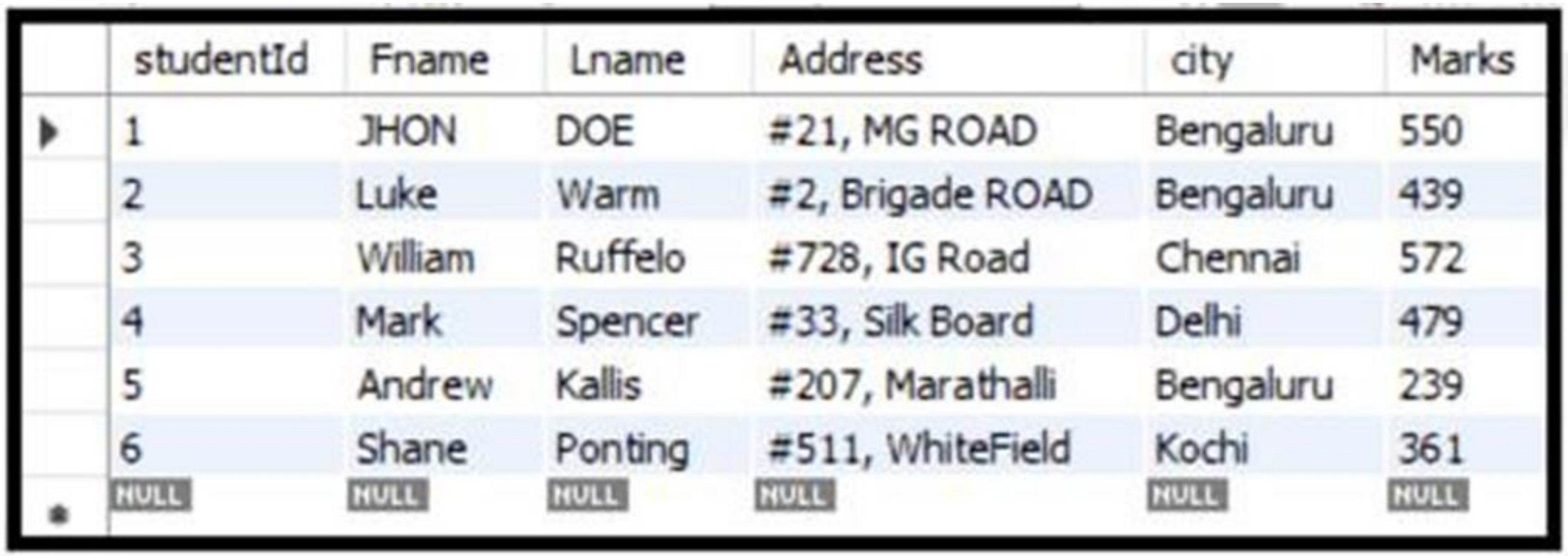
Partial Participation

Representation :

Example :

Tables:

- In relational database terms, a table is responsible for storing data in the database. Database tables consist of rows(records) and columns.
- Every Table should have a primary key (may be one column or more than one column).
- **Primary Key**: A column which values are unique. Can’t contain null values.



	studentId	Fname	Lname	Address	city	Marks
▶	1	JHON	DOE	#21, MG ROAD	Bengaluru	550
	2	Luke	Warm	#2, Brigade ROAD	Bengaluru	439
	3	William	Ruffelo	#728, IG Road	Chennai	572
	4	Mark	Spencer	#33, Silk Board	Delhi	479
	5	Andrew	Kallis	#207, Marathalli	Bengaluru	239
	6	Shane	Ponting	#511, WhiteField	Kochi	361
*	NULL	NULL	NULL	NULL	NULL	NULL

Database Constraints:

- Primary Key Constraints:
  - Not Null.
  - Unique.
- Not Null. (Enforces the field to have a value).
- Unique Key. (All values in the column are different).
- Check Constraints. (Check constraints on a salary column Minimum 1000, Max 2000).

About T-SQL

- Structured Query Language (SQL)
  - Developed by IBM in the 1970s
  - Adopted by ANSI and ISO standards bodies
  - Widely used in the industry
    - PL/SQL (Oracle), SQL Procedural Language (IBM), Transact-SQL (Microsoft)
- Transact-SQL is commonly referred to as T-SQL
  - The querying language of SQL Server 2016
- SQL is declarative
  - Describe what you want, not the individual steps

Categories of T-SQL Statements

DDL

- Data Definition Language
- Used to define database objects
- CREATE, ALTER, DROP

DML*

- Data Manipulation Language
- Used to query and manipulate data
- SELECT, INSERT, UPDATE, DELETE

DCL

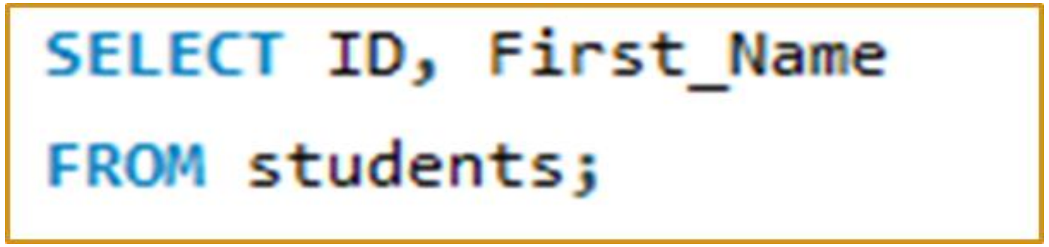
- Data Control Language
- Used to manage security permissions
- GRANT, REVOKE, DENY

*DML with SELECT is the focus of this course

SELECT statement:

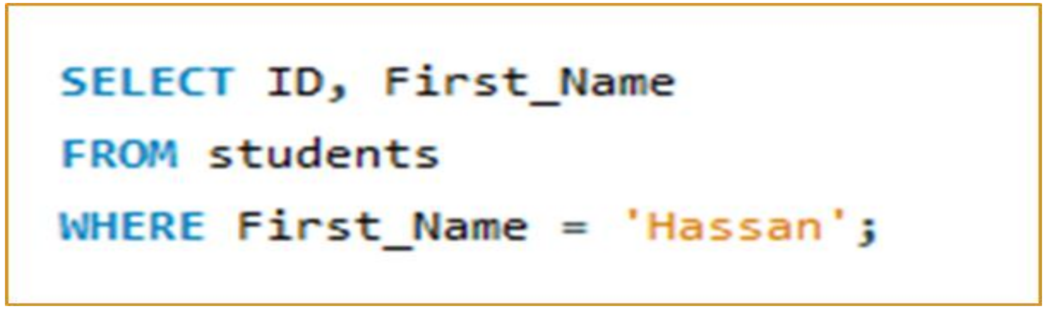
It’s used to select data from a database.

- It selects Columns ID and First_Name Only.



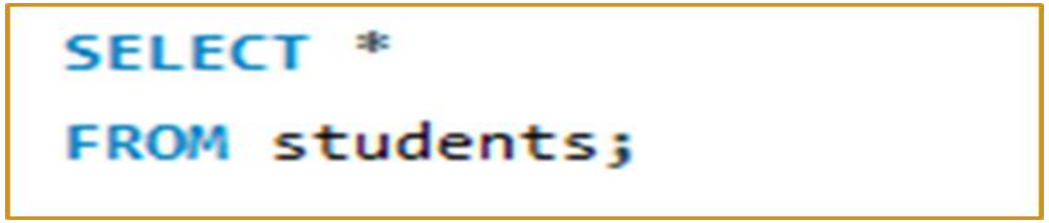
```
SELECT ID, First_Name
FROM students;
```

- It selects all Columns that have Hassan.



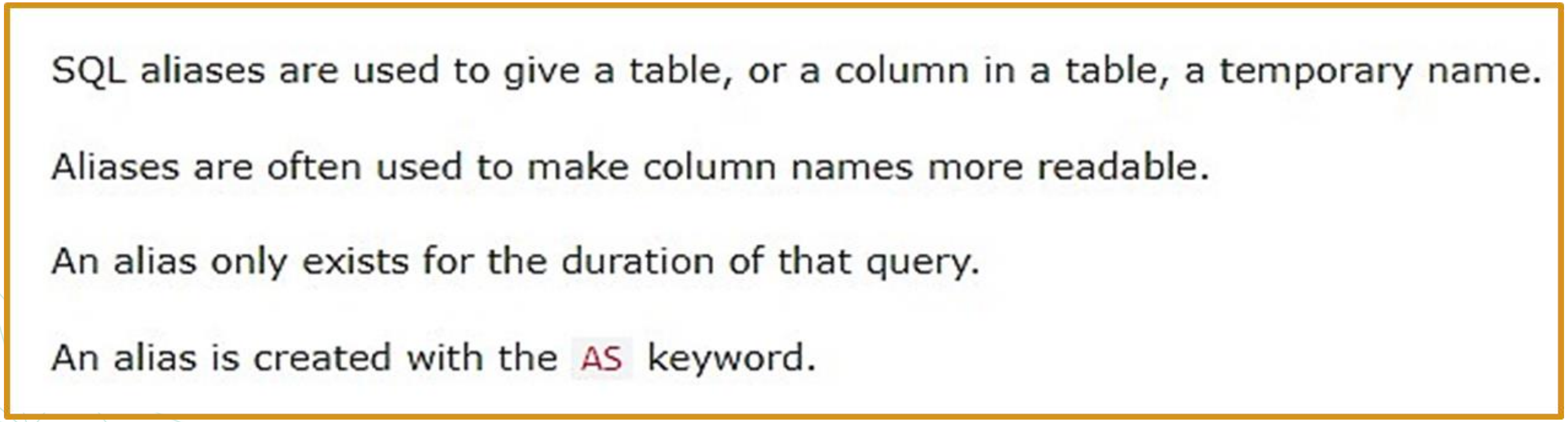
```
SELECT ID, First_Name
FROM students
WHERE First_Name = 'Hassan';
```

- It selects all the table content.



```
SELECT *
FROM students;
```

ALIAS

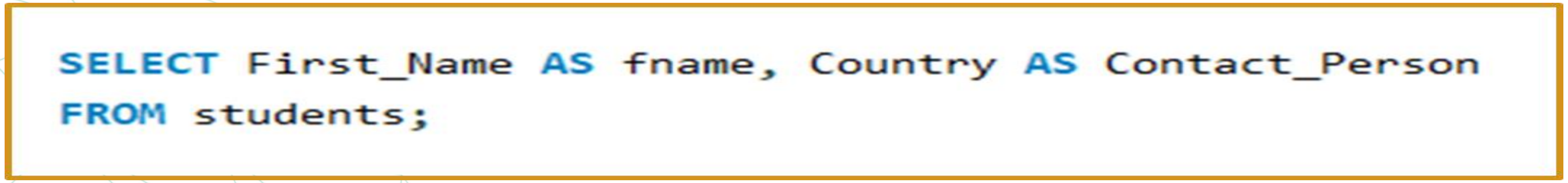


SQL aliases are used to give a table, or a column in a table, a temporary name.

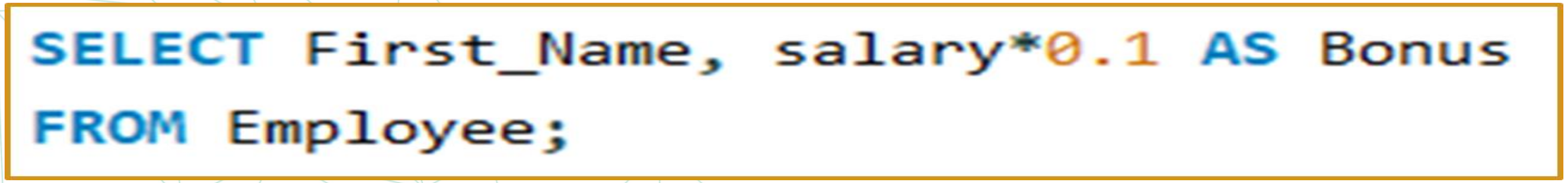
Aliases are often used to make column names more readable.

An alias only exists for the duration of that query.

An alias is created with the `AS` keyword.



```sql
SELECT First_Name AS fname, Country AS Contact_Person
FROM students;
```



```sql
SELECT First_Name, salary*0.1 AS Bonus
FROM Employee;
```

Understanding the Logical Order of Operations in SELECT Statements

Elements of a SELECT Statement

Element	Expression	Role
SELECT	<select list>	Defines which columns to return
FROM	<table source>	Defines table(s) to query
WHERE	<search condition>	Filters returned data using a predicate
GROUP BY	<group by list>	Arranges rows by groups
HAVING	<search condition>	Filters groups by a predicate
ORDER BY	<order by list>	Sorts the results

Logical Query Processing

5.	SELECT	<select list>
1.	FROM	<table source>
2.	WHERE	<search condition>
3.	GROUP BY	<group by list>
4.	HAVING	<search condition>
6.	ORDER BY	<order by list>

The order in which a query is written is not the order in which it is evaluated by SQL Server

Applying the Logical Order of Operations to Writing SELECT Statements

```sql
USE TSQL;

SELECT EmployeeId, YEAR(OrderDate) AS OrderYear
FROM Sales.Orders
WHERE CustomerId = 71
GROUP BY EmployeeId, YEAR(OrderDate)
HAVING COUNT(*) > 1
ORDER BY EmployeeId, OrderYear;
```

Demonstration: Logical Query Processing

In this demonstration, you will see how to:

- View query output that illustrates logical processing order

T-SQL Language Elements

- Predicates and Operators
- Functions
- Variables
- Expressions
- Batch Separators
- Control of Flow
- Comments

T-SQL Language Elements: Predicates and Operators

Elements:	**Predicates and Operators:**
Predicates	ALL, ANY, BETWEEN, IN, LIKE, OR, SOME
Comparison Operators	=, >, <, >=, <=, <>, !=, !>, !<
Logical Operators	AND, OR, NOT
Arithmetic Operators	*, /, %, +, -,
Concatenation	+

T-SQL Language Elements: Functions

String Functions

- SUBSTRING
- LEFT, RIGHT
- LEN
- REPLACE
- REPLICATE
- UPPER, LOWER
- LTRIM, RTRIM
- STUFF
- SOUNDEX

Date and Time Functions

- GETDATE
- SYSDATETIME
- GETUTCDATE
- DATEADD
- DATEDIFF
- YEAR
- MONTH
- DAY
- DATENAME
- DATEPART
- ISDATE

Aggregate Functions

- SUM
- MIN
- MAX
- AVG
- COUNT
- COUNT_BIG
- STDEV
- STDEVP
- VAR

T-SQL Language Elements: Variables

- Local variables in T-SQL temporarily store a value of a specific data type
- Name begins with single @ sign
  - @@ reserved for system functions
- Assigned a data type
- Must be declared and used within the same batch
- In SQL Server 2016, you can declare and initialize a variable in the same statement

```
DECLARE @search varchar(30) = 'Match%';
```

T-SQL Language Elements: Expressions

- Combination of identifiers, values, and operators evaluated to obtain a single result
- Can be used in SELECT statements
  - SELECT clause
  - WHERE clause
- Can be single constant, single-valued function, or variable
- Can be combined if expressions have the same data type

```sql
SELECT YEAR(orderdate) + 1 ...
SELECT qty * unitprice ...
```

T-SQL Language Elements: Control of Flow, Errors, and Transactions

Control of Flow	Error Handling	Transaction Control
• IF … ELSE • WHILE • BREAK • CONTINUE • BEGIN … END • WAITFOR	• TRY • CATCH • THROW	• BEGIN TRANSACTION • ROLLBACK TRANSACTION • COMMIT TRANSACTION • ROLLBACK WORK • SAVE TRANSACTION

The above are used in programmatic code objects

T-SQL Language Elements: Comments

- Two methods for marking text as comments
  - A block comment, surround text with /* and */

```
/*
      All the text in this paragraph will be treated as
      comments by SQL Server.
*/
```

  - An inline comment, precede text with --

```
-- This is an inline comment
```

  - Many T-SQL editors will color comments as above

T-SQL Language Elements: Batch Separators

- Batches are sets of commands sent to SQL Server as a unit
- Batches determine variable scope, name resolution
- To separate statements into batches, use a separator:
  - SQL Server tools use the GO keyword
  - GO is not an SQL Server T-SQL command
  - GO [count] executes the preceding batch [count] times

Demonstration: T-SQL Language Elements

In this demonstration, you will see how to:

- Use T-SQL language elements

Understanding Sets

- Set Theory and SQL Server
- Set Theory Applied to SQL Server Queries

Set Theory and SQL Server

Characteristics of a Set	Example
Elements of a set called Members	Customer as a member of set called Customers
Elements of a set are described by attributes	First name, Last name, Age
Elements must be unique	Customer ID

Set theory does not specify the order of its members

Set Theory Applied to SQL Server Queries

Application of Set Theory	Comments
Acts on all elements at once	Query the whole table
Use set-based processing	Tell the engine what you want to retrieve
Avoid cursors or loops	Do not process each item individually
Members of a set must be unique	Define unique keys in a table
No defined order to result set	Use ORDER BY clause if results need to be ordered

Understanding Predicate Logic

- Predicate Logic and SQL Server
- Predicate Logic Applied to SQL Server Queries

Predicate Logic and SQL Server

- Predicate logic is another mathematical basis for the relational database model
- In theory, a predicate is a property or expression that is either true or false
- Predicate is also referred to as a Boolean expression

Predicate Logic Applied to SQL Server Queries

Uses for Predicates

- Filtering data in queries
- Providing conditional logic to CASE expressions
- Joining tables
- Defining subqueries
- Enforcing data integrity
- Control of flow

Lab: Introduction to T-SQL Querying

- Exercise 1: Executing Basic SELECT Statements
- Exercise 2: Executing Queries That Filter Data Using Predicates
- Exercise 3: Executing Queries That Sort Data Using ORDER BY

Lab Scenario

You are an Adventure Works business analyst, who will be writing reports against corporate databases stored in SQL Server. To help you become more comfortable with SQL Server querying, the Adventure Works IT department has provided some common queries to run against their databases. You will review and execute these queries.

Module Review and Takeaways

- Review Question(s)