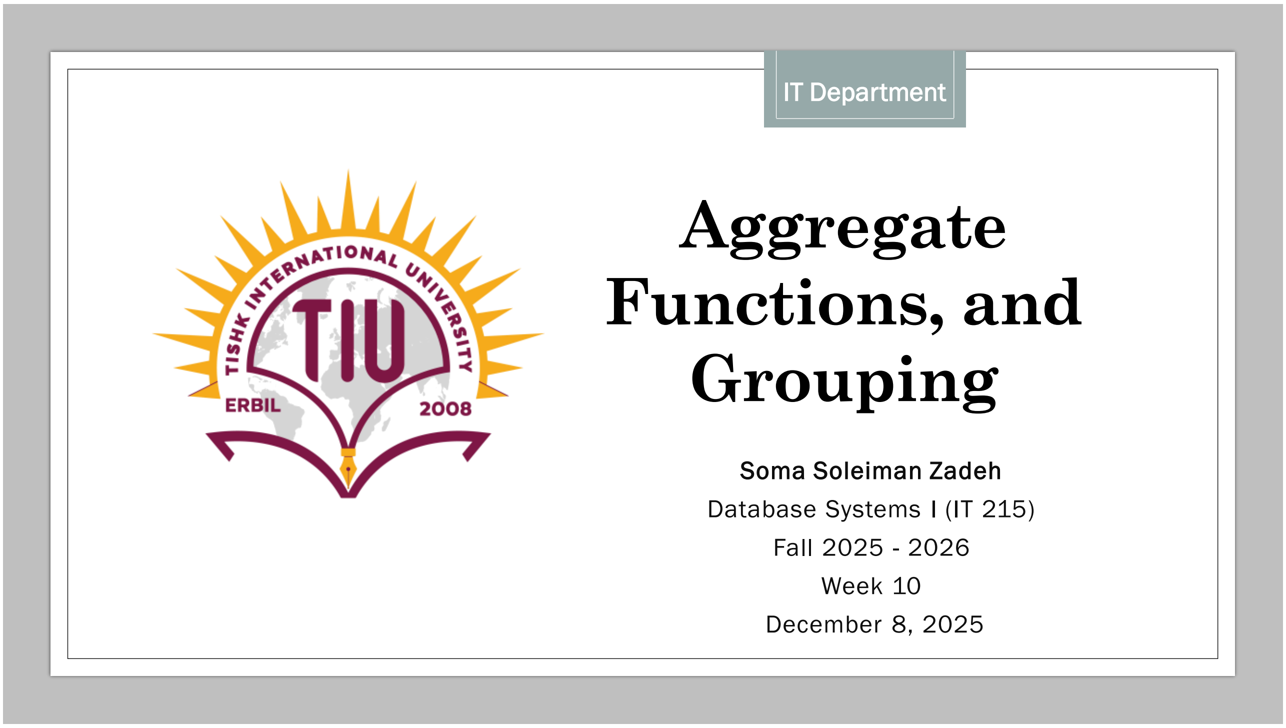


IT Department

Aggregate Functions, and Grouping

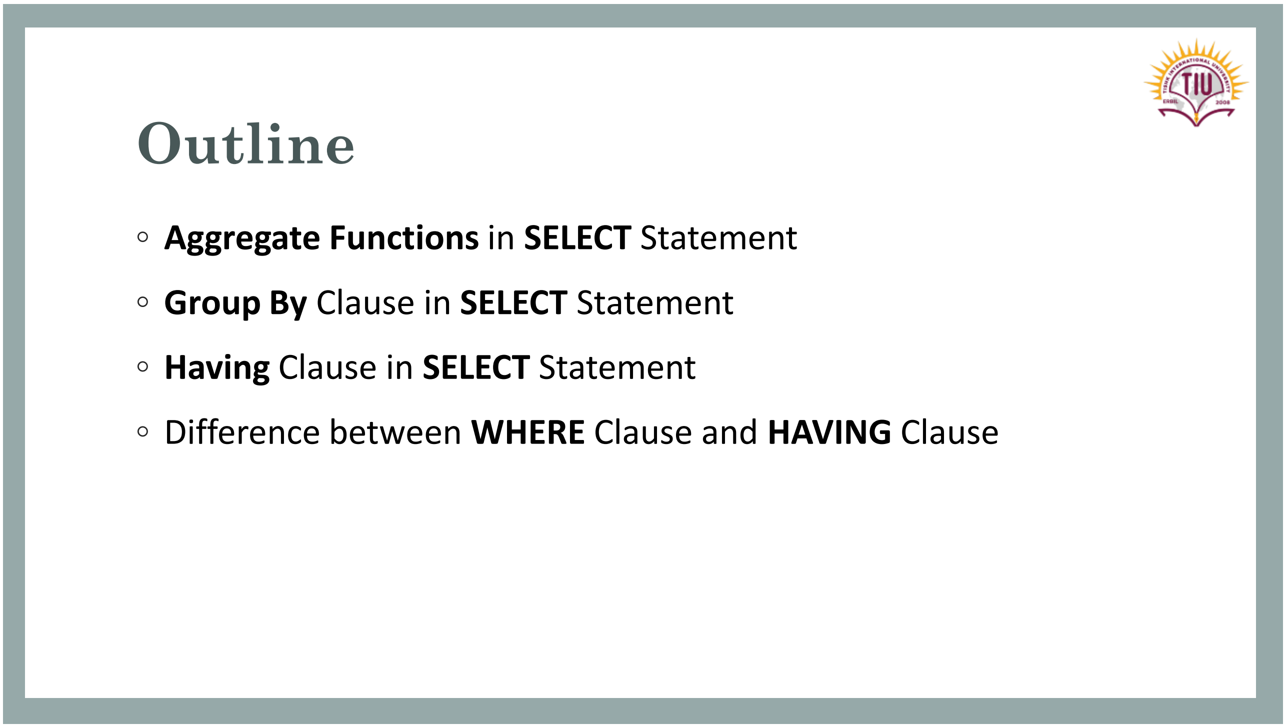
Soma Soleiman Zadeh

Database Systems I (IT 215)

Fall 2025 - 2026

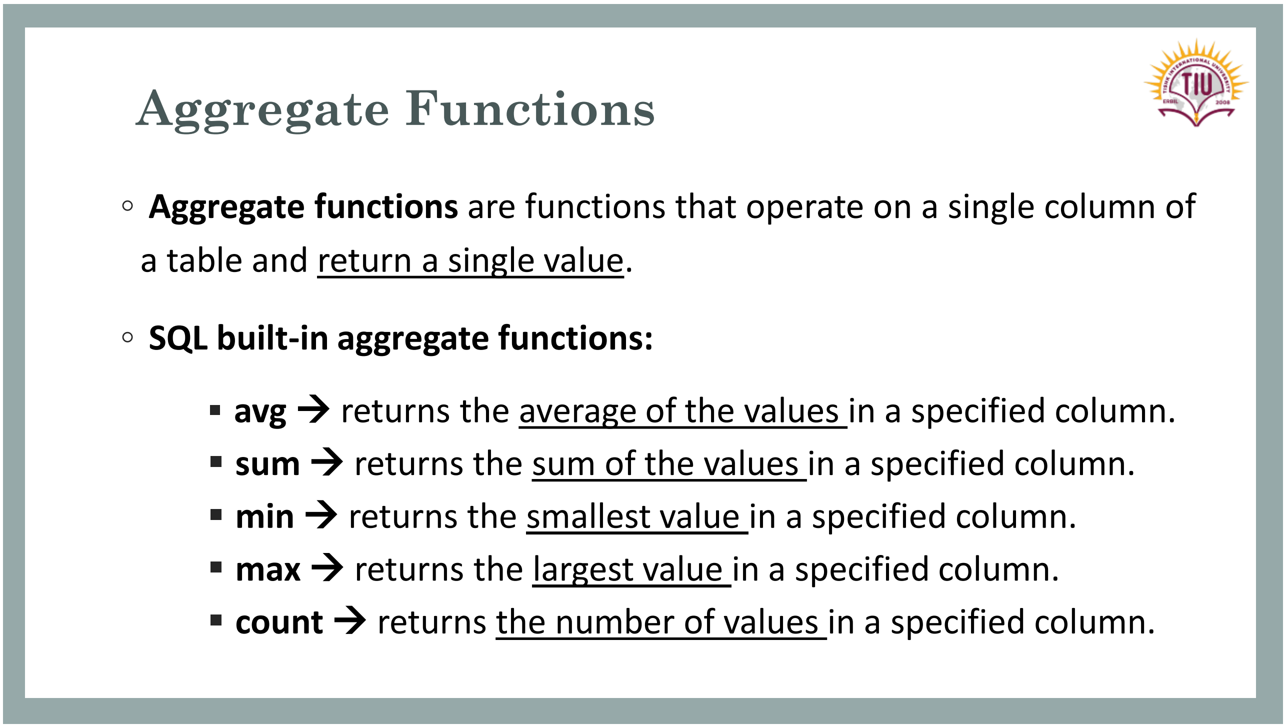
Week 10

December 8, 2025



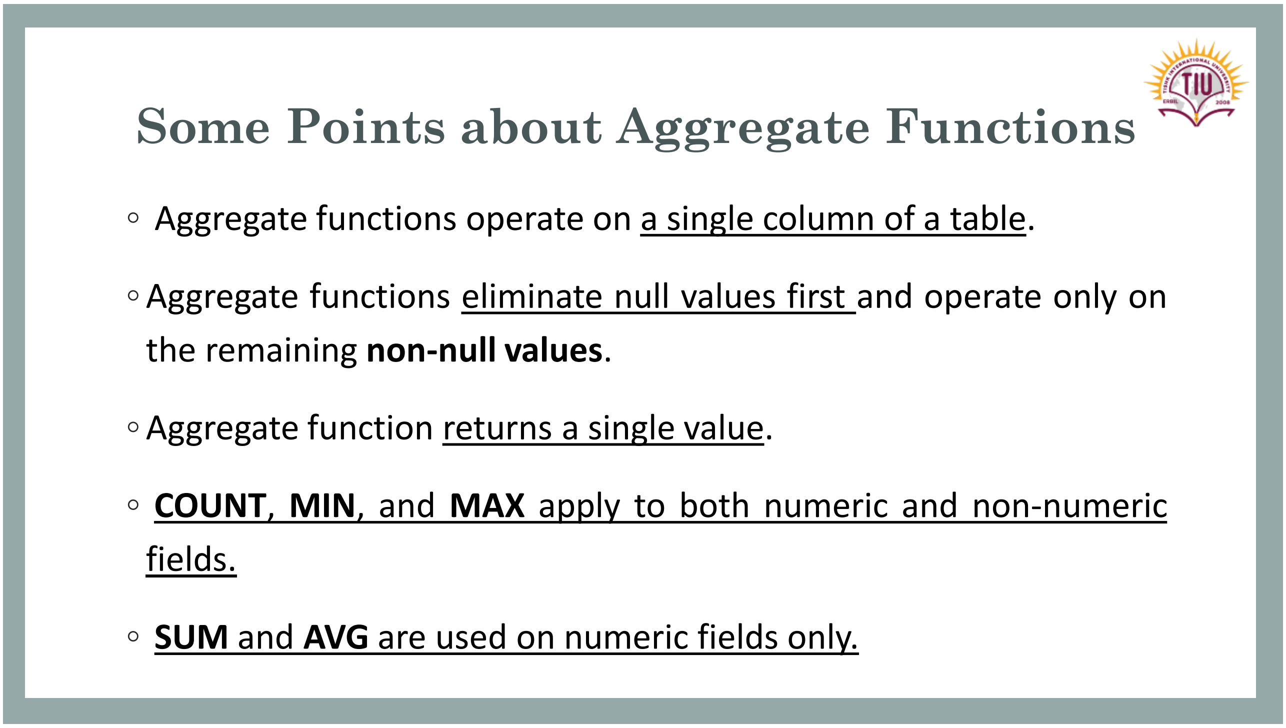
Outline

- **Aggregate Functions** in **SELECT** Statement
- **Group By** Clause in **SELECT** Statement
- **Having** Clause in **SELECT** Statement
- Difference between **WHERE** Clause and **HAVING** Clause



Aggregate Functions

- **Aggregate functions** are functions that operate on a single column of a table and return a single value.
- **SQL built-in aggregate functions:**
  - **avg** → returns the average of the values in a specified column.
  - **sum** → returns the sum of the values in a specified column.
  - **min** → returns the smallest value in a specified column.
  - **max** → returns the largest value in a specified column.
  - **count** → returns the number of values in a specified column.



Some Points about Aggregate Functions

- Aggregate functions operate on a single column of a table.
- Aggregate functions eliminate null values first and operate only on the remaining **non-null values**.
- Aggregate function returns a single value.
- **COUNT**, **MIN**, and **MAX** apply to both numeric and non-numeric fields.
- **SUM** and **AVG** are used on numeric fields only.

COUNT Function – Example

- ○ **count()** function counts the number of values in a specified column.
- ○ **Example –** Write a query to find the **total number of staff** whose salary is more than $15000.

```
SELECT    COUNT(staffNo)  AS  myCount
FROM      Staff
WHERE     salary > 15000 ;
```

myCount
3

Staff

staffNo	sName	position	gender	salary
SL21	John	Manager	Male	**30000**
SG37	Ann	Assistant	Female	12000
SG14	David	Supervisor	Male	**18000**
SA9	Mary	Assistant	Female	9000
SG5	Susan	Manager	Female	**24000**
SL41	Julie	Assistant	Female	9000

COUNT Function – Example

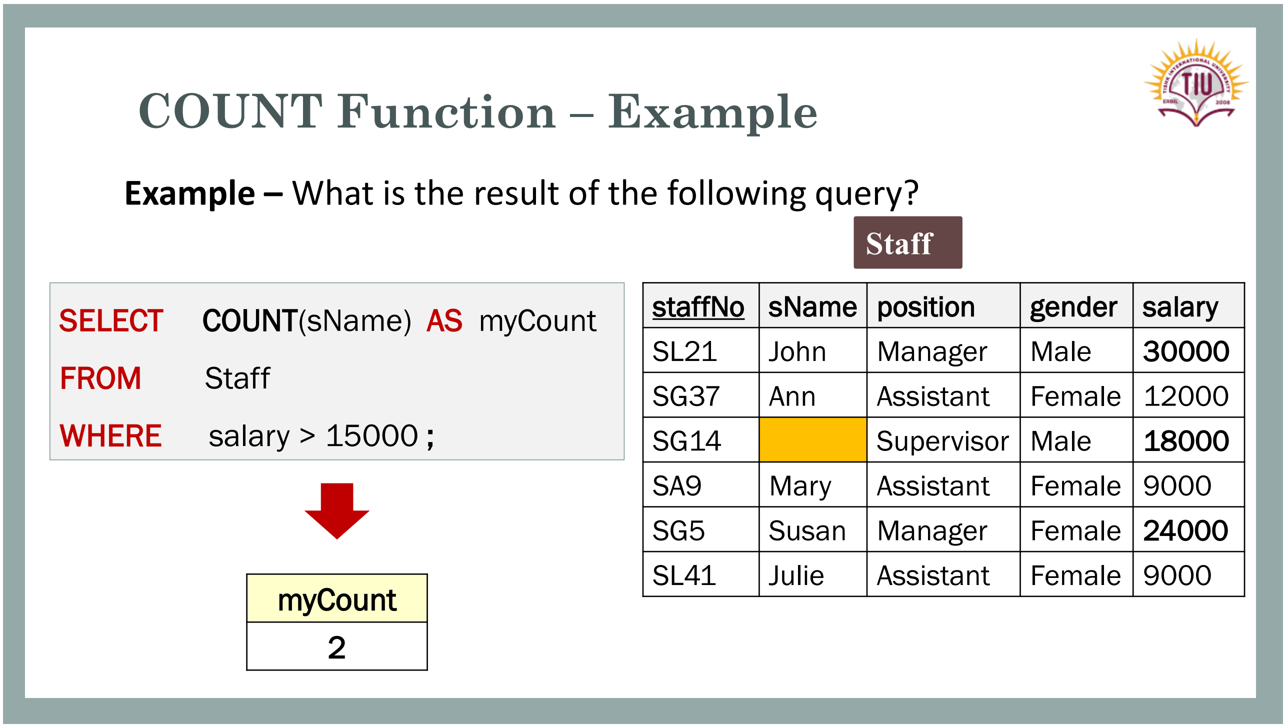
Example – What is the result of the following query?

```
SELECT    COUNT(sName)  AS  TotalStaff
FROM      Staff;
```

TotalStaff
5

Staff

staffNo	sName	position	gender	salary
SL21	John	Manager	Male	**30000**
SG37	Ann	Assistant	Female	12000
SG14	🟧	Supervisor	Male	**18000**
SA9	Mary	Assistant	Female	9000
SG5	Susan	Manager	Female	**24000**
SL41	Julie	Assistant	Female	9000



COUNT Function – Example

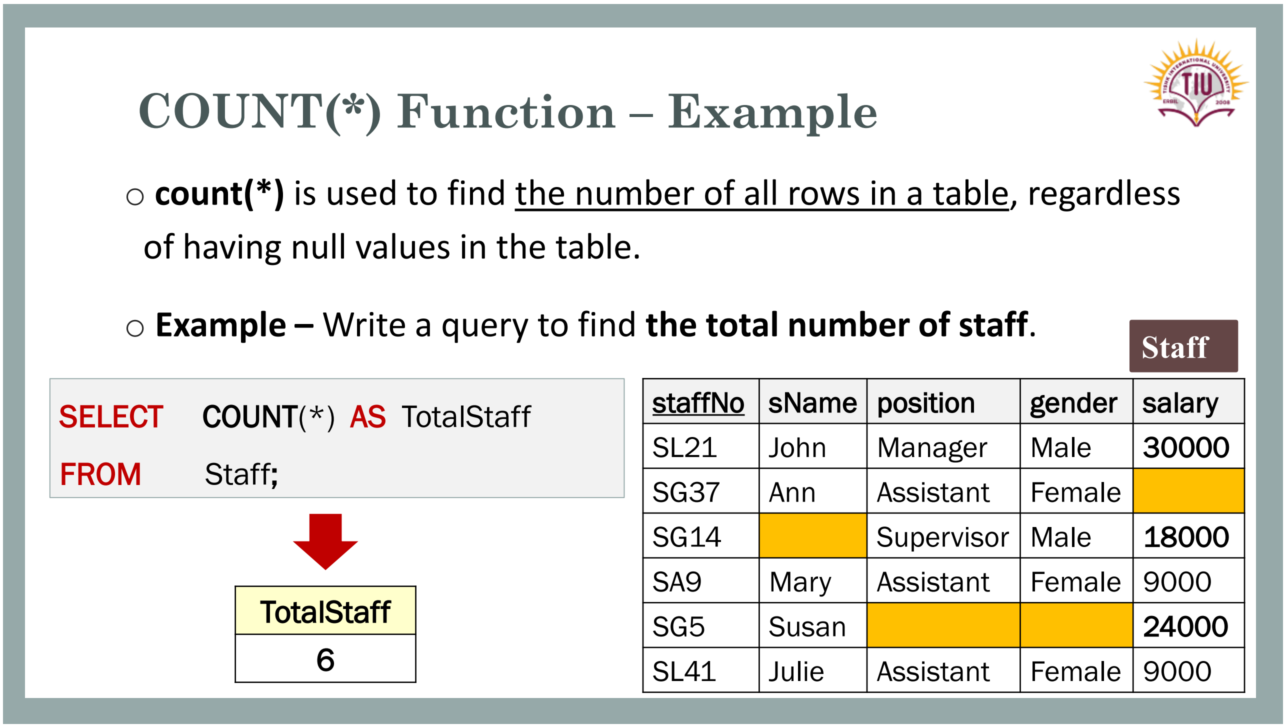
Example – What is the result of the following query?

```
SELECT    COUNT(sName)  AS  myCount
FROM      Staff
WHERE     salary > 15000 ;
```

myCount
2

Staff

staffNo	sName	position	gender	salary
SL21	John	Manager	Male	**30000**
SG37	Ann	Assistant	Female	12000
SG14	🟨	Supervisor	Male	**18000**
SA9	Mary	Assistant	Female	9000
SG5	Susan	Manager	Female	**24000**
SL41	Julie	Assistant	Female	9000



COUNT(*) Function – Example

- **count(*)** is used to find the number of all rows in a table, regardless of having null values in the table.
- **Example –** Write a query to find **the total number of staff**.

```
SELECT    COUNT(*)  AS  TotalStaff
FROM      Staff;
```

TotalStaff
6

Staff

staffNo	sName	position	gender	salary
SL21	John	Manager	Male	**30000**
SG37	Ann	Assistant	Female	🟨
SG14	🟨	Supervisor	Male	**18000**
SA9	Mary	Assistant	Female	9000
SG5	Susan	🟨	🟨	**24000**
SL41	Julie	Assistant	Female	9000

SUM Function – Example

Example – Write a query to find **the total number** of **managers** and the **sum of their salaries**.

```
SELECT    COUNT(*)  AS  totalManagers , SUM(salary)  AS  TotalSalary
FROM      Staff
WHERE     position = 'Manager' ;
```

totalManagers	TotalSalary
2	54000

Staff

staffNo	sName	position	gender	salary
SL21	John	Manager	Male	**30000**
SG37	Ann	Assistant	Female	12000
SG14	David	Supervisor	Male	**18000**
SA9	Mary	Assistant	Female	9000
SG5	Susan	Manager	Female	**24000**
SL41	Julie	Assistant	Female	9000

MIN, MAX, AVG Functions – Example

Example – Write a query to find the **minimum**, **maximum**, and **average** staff salary.

```
SELECT    MIN(salary)  AS  lowestSalary , MAX(salary)  AS  highestSalary , AVG(salary)  AS  averageSalary
FROM      Staff ;
```

lowestSalary	highestSalary	averageSalary
9000	30000	17000

Staff

staffNo	sName	position	gender	salary
SL21	John	Manager	Male	**30000**
SG37	Ann	Assistant	Female	12000
SG14	David	Supervisor	Male	**18000**
SA9	Mary	Assistant	Female	9000
SG5	Susan	Manager	Female	**24000**
SL41	Julie	Assistant	Female	9000

Class Activity

Example – Write an SQL query to find **ID** and **name** of student(s) with the highest GPA.

```
SELECT    stuID, stuName
FROM      Student
WHERE     GPA = ( SELECT  max(GPA)
                  FROM  Student );
```

The result of this query is the highest GPA. Then the IDs and names of the students with the highest GPA are retrieved.

Student

stuID	stuName	deptName	GPA
IT011	Smith	IT	67
EN007	Tom	English	**94**
EN189	Jones	English	52
CI209	Owen	Civil Eng.	72
IT112	Ann	IT	**94**
IT056	Edward	IT	60

GROUP BY Clause

- The **GROUP BY** clause in SQL is a powerful function used to group rows from a table based on the values of one or more columns.
- It is often used in conjunction with aggregate functions (such as COUNT, SUM, AVG, MAX, or MIN) to perform calculations on these grouped data.

GROUP BY Clause

○ Find the total number of students in each department.

```
SELECT      deptName, COUNT(stuID)  AS  TotalStudent
FROM        Student
GROUP BY    deptName;
```

deptName	TotalStudent
IT	3
English	2
Civil Eng.	1

Student

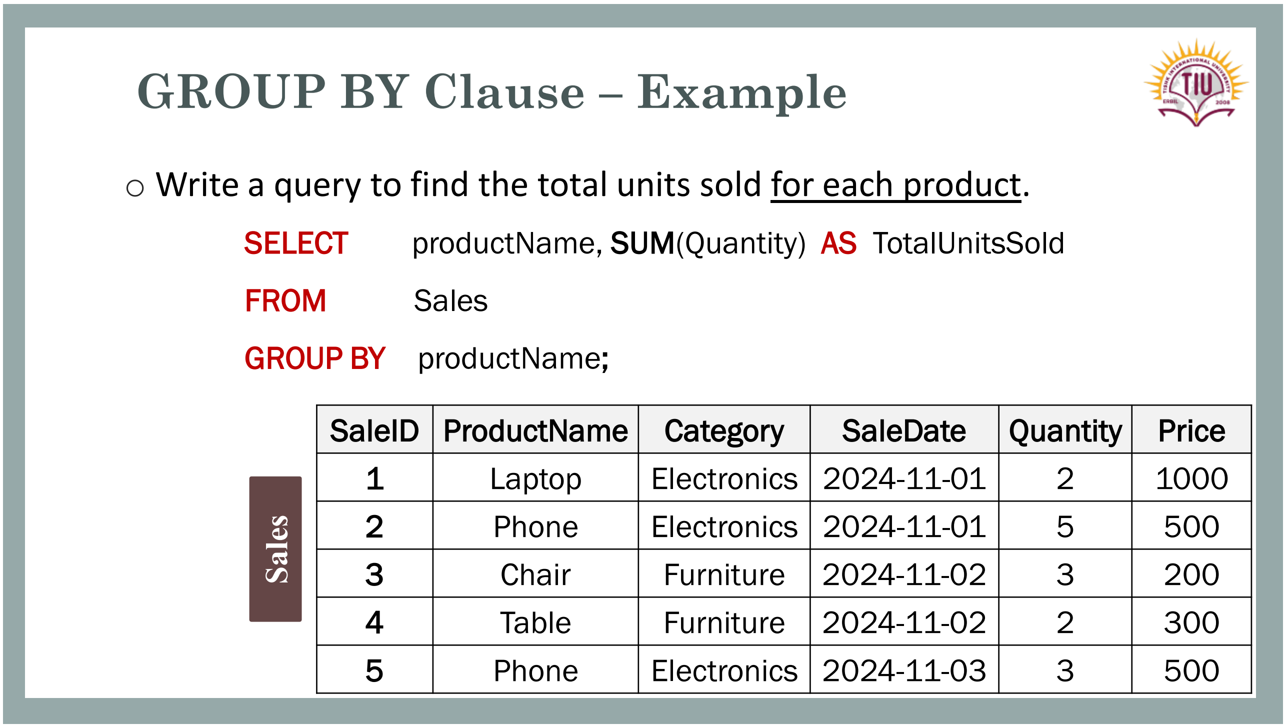
stuID	stuName	deptName	GPA
IT011	Smith	IT	67
EN007	Tom	English	94
EN189	Jones	English	52
CI209	Owen	Civil Eng.	72
IT112	Ann	IT	94
IT056	Edward	IT	60

GROUP BY Clause – Example

○ Imagine you manage a store and have a table named **Sales** that tracks daily sales transactions:

Sales

SaleID	ProductName	Category	SaleDate	Quantity	Price
1	Laptop	Electronics	2024-11-01	2	1000
2	Phone	Electronics	2024-11-01	5	500
3	Chair	Furniture	2024-11-02	3	200
4	Table	Furniture	2024-11-02	2	300
5	Phone	Electronics	2024-11-03	3	500



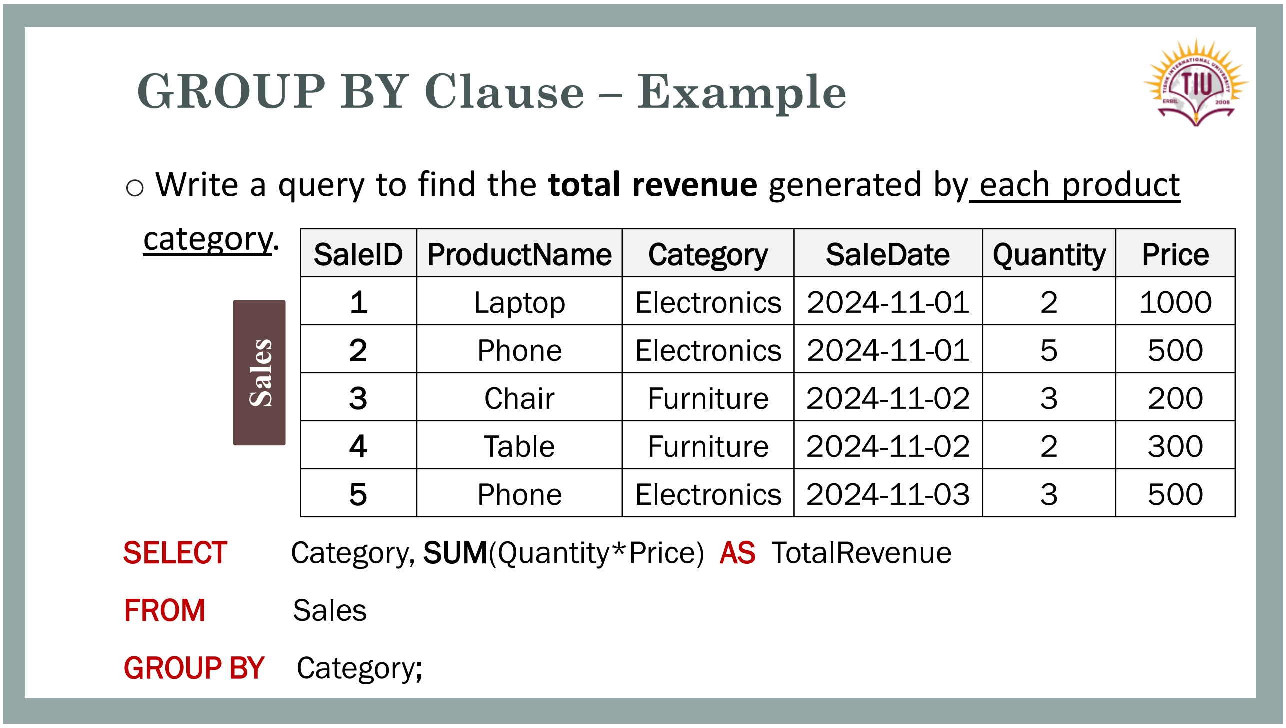
GROUP BY Clause – Example

○ Write a query to find the total units sold for each product.

```
SELECT      productName, SUM(Quantity)  AS  TotalUnitsSold
FROM        Sales
GROUP BY    productName;
```

Sales

SaleID	ProductName	Category	SaleDate	Quantity	Price
1	Laptop	Electronics	2024-11-01	2	1000
2	Phone	Electronics	2024-11-01	5	500
3	Chair	Furniture	2024-11-02	3	200
4	Table	Furniture	2024-11-02	2	300
5	Phone	Electronics	2024-11-03	3	500



GROUP BY Clause – Example

○ Write a query to find the **total revenue** generated by each product category.

Sales

SaleID	ProductName	Category	SaleDate	Quantity	Price
1	Laptop	Electronics	2024-11-01	2	1000
2	Phone	Electronics	2024-11-01	5	500
3	Chair	Furniture	2024-11-02	3	200
4	Table	Furniture	2024-11-02	2	300
5	Phone	Electronics	2024-11-03	3	500

```
SELECT      Category, SUM(Quantity*Price)  AS  TotalRevenue
FROM        Sales
GROUP BY    Category;
```

Queries with Group By Clause

- When an SQL query uses grouping,
  - the only attributes that appear in the **select** statement without being aggregated are those that are present in the **group by** clause.

Question – Is the following query allowed in SQL?

```
SELECT      deptName, stuID, COUNT(stuID)
FROM        Student
GROUP BY    deptName;
```

Student

stuID	stuName	deptName	GPA
IT011	Smith	IT	67
EN007	Kate	English	94
EN189	Jones	English	52
CI209	Owen	Civil Eng.	72
IT112	Ann	IT	94
IT056	Edward	IT	60

GROUP BY … HAVING

- Sometimes we want to select certain groups from the result of **GROUP BY**. In such cases, we use the **HAVING** clause.
- The **HAVING** clause always comes directly after the **GROUP BY** clause:
  - **HAVING** clause sets a condition that applies to groups, not rows of the original table.

GROUP BY … HAVING – Example

○ For each department **that has at least 3 students**, find the **total number of students** at that department and **their average GPA**.

```
SELECT      deptName, COUNT(stuID) AS totalStudents, AVG(GPA) AS AverageGPA
FROM        Student
GROUP BY    deptName
HAVING      COUNT(stuID)>=3;
```

Student

stuID	stuName	deptName	GPA
IT011	Smith	IT	50
EN007	Kate	English	90
EN189	Jones	English	70
CI209	Owen	Civil Eng.	95
IT112	Ann	IT	70
IT056	Edward	IT	60
IT101	Mike	IT	20

General Syntax of SELECT Query

```
SELECT      column/columns and/or aggregate functions
FROM        table/tables
WHERE       condition on rows
GROUP BY    one column
HAVING      condition on groups
ORDER BY    column/columns;
```

WHERE vs. HAVING

- Using of both a having clause and a where clause:
  - **Where** clause:
    - Applying conditions on all rows of a table
  - **Having** clause:
    - Applying conditions on groups
    - **Having** clause always comes after **group by** clause

WHERE vs. HAVING

Example – By having Employee table, for each position **that its minimum salary is greater than $2000**, find the minimum amount of salary of employees **hired after 2015**.

EmpID	HiringDate	Position	Salary
011	2014	Engineer	4000
012	2016	Technician	1500
013	2020	Technician	1600
014	2017	Manager	7000
015	2020	Engineer	2500
016	2013	Engineer	3000
017	2011	Manager	4500
018	2018	Manager	5000
019	2018	Engineer	3000