

## Module 9 Advanced SQL Query

Course Learning Outcomes:

1. Learn how to write SQL SELECT statement and know the result involving Expression, Aggregate functions, Outer join, and Alias

### Overview

More on SQL SELECT queries

- SQL Expression
- Aggregate functions with grouping
- More table join types and options
- Alias

### Expression

It is a combination of symbols and operators that returns a single value.

An Expression can be:

- A single constant, column, variable, or scalar function
  - 12
  - 3.14
  - 'Anne Curtis'
  - '05/18/16'
  - CustomerName
  - GetDate()
- A number, columns, literals, functions connected by operators
  - $10 * 4 + 3$
  - 'Anne' + 'Curtis'
  - Address + ',' + City + ',' + Zipcode
  - Quantity \* Price
  - $3.5 * \text{Price}$

#### 1.1 Expression used as Columns

```
SELECT ProductName, Price * 2 AS 'New Price'  
FROM Products;
```

```
SELECT FirstName+ ' ' + LastName  
FROM Students
```

```
SELECT GETDATE();
```

## 1.2 Expression used for Comparison

```
SELECT *  
FROM Products  
WHERE Stocks < 5;
```

```
SELECT ProductName, Price * 3 AS Discount  
FROM Products  
WHERE Price * 3 <= 15;
```

## 2. Aggregate Functions

Using aggregate functions for row calculation

- MAX – maximum of all or selected values
- MIN – minimum of all or selected values
- COUNT – number of all or selected values
- SUM – sum of all or selected values
- AVG – average of all or selected values

```
SELECT COUNT(ProductID) AS NumberOfProducts  
FROM Products;
```

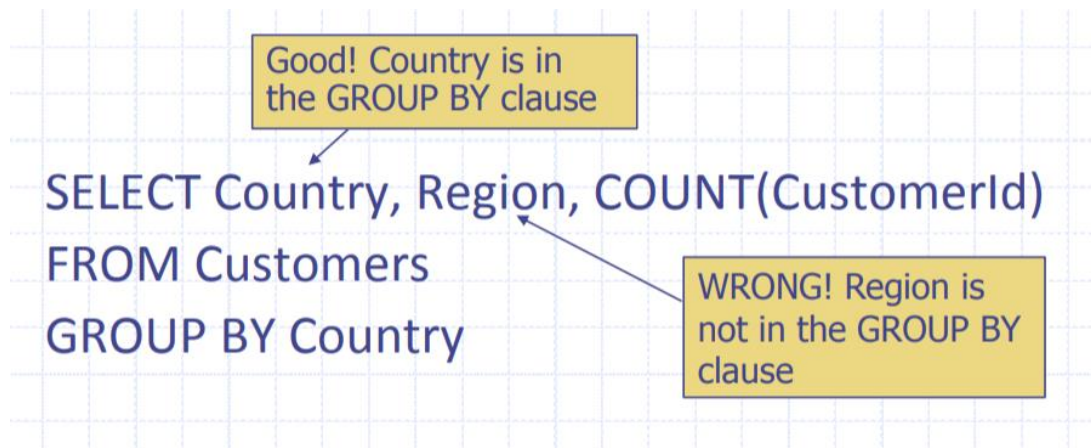
## 2.2 Grouping

**GROUP BY** – aggregation with groups. Used to get aggregation results for different groups of records.

```
SELECT CategoryID, AVG(Price)
FROM Products
GROUP BY CategoryID;
```

### Limitations of GROUP BY

Columns or expressions can be in the SELECT clause only if they are in the GROUP BY clause.



## 3. Table Join Types

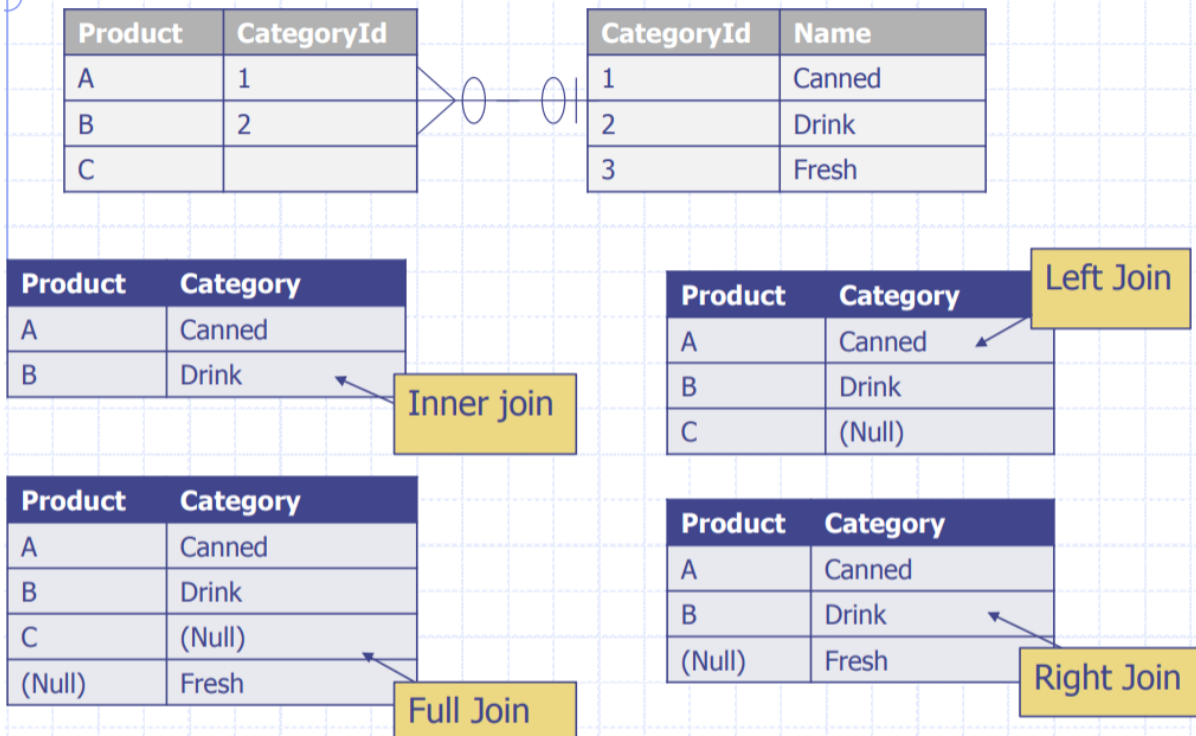
### Inner Join

Records that have matching records from two (2) tables

### Outer Join

- It is needed when minimum cardinality is optional on a table.
- **Left Join** – include all qualified records from the left table in the join condition even if they do not have matching records in the right table.
- **Right Join** - include all qualified records from the right table in the join condition even if they do not have matching records in the left table.
- **Full Join** – include all qualified records from both tables in the join condition.

# Table Join Effect



## OUTER JOIN Example

- ◆ Get customers and their orders; also include customers who have never placed an order

```
SELECT CompanyName, OrderID
FROM
Customers LEFT JOIN Orders ON
Customers.CustomerID = Orders.CustomerID
ORDER BY OrderID
```

- ◆ Execution result

	CompanyName	OrderID
1	Paris spécialités	NULL
2	FISSA Fabrica Inter. Salchichas S.A.	NULL
3	Wilman Kala	10248
4	Tradição Hipermercados	10249
5	Henri Carnes	10250

The first two rows will not be included for an inner join.

### 3. Alias

#### ◆ Column alias: representing derived and constant columns

```
SELECT CategoryID, AVG(UnitPrice) Price  
FROM Products  
GROUP BY CategoryID  
ORDER BY Price;
```

Column alias can be used in  
ORDER BY

```
SELECT ProductName, UnitPrice * 0.9 Discount  
FROM Products  
WHERE UnitPrice * 0.9 > 20;
```

Column alias can NOT be  
used in WHERE or HAVING  
clause (SQL Server)

#### ◆ Table alias: commonly used in table joins

```
SELECT ProductName, CategoryName  
FROM Products AS p, Categories c  
Where p.CategoryID = c.CategoryID
```

"AS" is optional.

If an alias is assigned, it must be used  
instead of the original table name

## References and Supplementary Materials

### Books and Journals

1. Ramez Elmasri and Shamkant B. Navathe; 2016; Fundamentals of Database Systems; USA; Pearson
2. [Dr. Kashif Qureshi; 2018; Advanced concepts of information technology; educreation publishing; India.](#)

### Online Supplementary Reading Materials

1. RelationalDBDesing;; March 31, 2020
2. Advantages of Database Management System;  
<https://www.tutorialspoint.com/Advantages-of-Database-Management-System>;  
March 31, 2020
3. DesigningandManagingData;  
[https://www.academia.edu/36712448/Entity Relationship Diagram ERD Basics CIS 3730 Designing and Managing Data](https://www.academia.edu/36712448/Entity_Relationship_Diagram_ERD_Basics_CIS_3730_Designing_and_Managing_Data); April 01, 2020
4. DesigningandManagingData; <http://jackzheng.net/teaching/archive/cis3730-2010-fall/>; April 03,2020

### Online Instructional Videos

1. Introduction to Database; [https://www.youtube.com/watch?v=8e-wgQnsFxE&list=PLJ5C\\_6qdAvBHKccG0ZyOxcf\\_2YO6r4Q4I](https://www.youtube.com/watch?v=8e-wgQnsFxE&list=PLJ5C_6qdAvBHKccG0ZyOxcf_2YO6r4Q4I); March 21, 2020
2. Three levels of Architecture/DBMS;  
<https://www.youtube.com/watch?v=j6xh8wKfjkY>; April 01,2020
3. Relational Data Model; <https://www.youtube.com/watch?v=TsSf1Z3g0Kk>; Arpil 06, 2020.
4. Basic Concept of Database Normalization;  
<https://www.youtube.com/watch?v=xoTyrdT9SZI>; April 08, 2020
5. SQL Basics; <https://www.youtube.com/watch?v=zbMHLJ0dY4w>; April 10, 2020
6. Joins|SQL|; <https://www.youtube.com/watch?v=efpFCd8iFAQ>; April 11, 2020