

19CSE202 - DATABASE MANAGEMENT SYSTEMS

SQL - AGGREGATE FUNCTIONS, SET OPERATIONS, JOINS

SQL - GROUPING

- ❖ The GROUP BY clause groups records into summary rows.
- ❖ GROUP BY returns one record for each group.
- ❖ GROUP BY queries often include aggregates: COUNT, MAX, SUM, AVG, etc.
- ❖ GROUP BY can group by one or more columns.

```
SELECT column-name  
FROM table-name  
WHERE condition  
GROUP BY column-names
```

```
SELECT column-names  
FROM table-name  
WHERE condition  
GROUP BY column-names  
ORDER BY column-names
```

SQL GROUP BY Example

- ❖ List the number of customers in each country.

Count	Country
3	Argentina
2	Austria

```
SELECT COUNT(Id), Country  
FROM Customer  
GROUP BY Country
```

CUSTOMER
Id 
FirstName
LastName
City
Country
Phone

- ❖ List the number of customers in each country sorted high to low

```
SELECT COUNT(Id), Country  
FROM Customer  
GROUP BY Country  
ORDER BY COUNT(Id) DESC
```

Count	Country
13	USA
11	France

SQL - HAVING

- ❖ HAVING is like WHERE but operates on grouped records returned by a GROUP BY.
- ❖ HAVING applies to summarized group records, whereas WHERE applies to individual records.
- ❖ Only the groups that meet the HAVING criteria will be returned.
- ❖ HAVING requires that a GROUP BY clause is present.
- ❖ WHERE and HAVING can be used in the same query.

GROUP BY - HAVING

```
SELECT column-names  
FROM table-name  
WHERE condition  
GROUP BY column-names  
HAVING condition
```

```
SELECT column-names  
FROM table-name  
WHERE condition  
GROUP BY column-names  
HAVING condition  
ORDER BY column-names
```

List the number of customers in each country. Only include countries with more than 10 customers.

```
SELECT COUNT(Id), Country  
FROM Customer  
GROUP BY Country  
HAVING COUNT(Id) > 10
```

Count	Country
11	France
11	Germany
13	USA

GROUP BY - HAVING - ORDER BY

- ❖ List the number of customers in each country, except the USA, sorted high to low. Only include countries with 9 or more customers.

Count	Country
11	France
11	Germany
9	Brazil

```
SELECT COUNT(Id), Country  
FROM Customer  
WHERE Country <> 'USA'  
GROUP BY Country  
HAVING COUNT(Id) >= 9  
ORDER BY COUNT(Id) DESC
```

CUSTOMER	
Id	🔑
FirstName	
LastName	
City	
Country	
Phone	

GROUP BY - JOIN

- ❖ List all customer with average orders between \$1000 and \$1200.

```
SELECT AVG(TotalAmount), FirstName, LastName  
FROM Order O JOIN Customer C ON O.CustomerId = C.Id  
GROUP BY FirstName, LastName  
HAVING AVG(TotalAmount) BETWEEN 1000 AND 1200
```

Average	FirstName	LastName
1081.215000	Miguel	Angel Paolino
1063.420000	Isabel	de Castro
1008.440000	Alexander	Feuer

ORDER
Id 
OrderDate
OrderNumber
CustomerId
TotalAmount

CUSTOMER
Id 
FirstName
LastName
City
Country
Phone

SQL Alias

- ❖ An Alias is a shorthand for a table or column name.
- ❖ Aliases reduce the amount of typing required to enter a query.
- ❖ Complex queries with Aliases are generally easier to read.
- ❖ Aliases are useful with JOINs and aggregates: SUM, COUNT, etc.
- ❖ An Alias only exists for the duration of the query.

```
SELECT column-name AS alias-name  
FROM table-name alias-name  
WHERE condition
```

SQL - Alias Eg:

- ❖ List total customers in each country. Display results with easy to understand column headers.

```
SELECT COUNT(C.Id) AS TotalCustomers, C.Country AS Nation  
FROM Customer C  
GROUP BY C.Country
```

TotalCustomers and Nation are column Aliases. The table Alias is (C)

TotalCustomers	Nation
3	Argentina
2	Austria
2	Belgium

CUSTOMER	
Id	PK
FirstName	
LastName	
City	
Country	
Phone	

SQL JOIN

The **SQL JOIN** clause is used whenever we have to select data from 2 or more tables.

Are used to extract data from 2 (or more) tables, when we need a relationship between certain columns in these tables.

Are used to relate information in different tables and used as a part of SQL query that retrieves rows from 2(or more) tables.

A SQL Join condition is always used in the WHERE clause of SELECT,UPDATE and DELETE statements

CARTESIAN JOIN

- ❖ If a SQL join condition is omitted or if it is invalid, the join operation will result in a CARTESIAN product.
- ❖ SYNTAX: SELECT col1, col2 FROM table1, table2;
- ❖ Eg: SELECT order_id, prod_name FROM Customer, Product

SELF JOINS

- ❖ A Self join is the type of SQL join where we join a particular table to itself. Here it is necessary to ensure that the join statement defines an ALIAS name for both the copies of the tables to avoid column ambiguity.
- ❖ Self Join on Course table:

```
SELECT a.course_name AS COURSE, b.course_name AS PREREQUISITE_COURSE  
      FROM COURSE a, COURSE b  
     WHERE a.pre_course = b.course_id
```

NATURAL/CARTESIAN JOIN

```
SELECT prod_ID, prod_name, order_id, quantity  
FROM product NATURAL JOIN customer  
WHERE prod_name LIKE('teak%') AND quantity = 10;
```

SQL JOIN

Cars

CarMake	Model	Year	Color
Toyota	Camry XLE	2005	Gray
Honda	Accord EX	2002	Black
Lexus	ES 350	2008	Gray
BMW	3 Series Coupe	2008	Red

CountryOfOrigin

CarMake	Country
Toyota	Japan
Honda	Japan
Lexus	Japan
BMW	Germany

```
SELECT Cars.Model, CountryOfOrigin.Country  
FROM Cars JOIN CountryOfOrigin ON  
Cars.CarMake = CountryOfOrigin.CarMake;
```

Model	Country
Camry XLE	Japan
Accord EX	Japan
ES 350	Japan
3 Series Coupe	Germany

What is the output?

```
SELECT Cars.Model  
FROM Cars JOIN CountryOfOrigin ON  
Cars.CarMake = CountryOfOrigin.CarMake  
WHERE CountryOfOrigin.Country = 'Japan';
```

Model

Camry XLE

Accord EX

ES 350

JOIN Eg

- ❖ List the total amount ordered for each customer

ORDER	
Id	🔑
OrderDate	
OrderNumber	
CustomerId	
TotalAmount	

CUSTOMER	
Id	🔑
FirstName	
LastName	
City	
Country	
Phone	

```
SELECT SUM(O.TotalAmount) AS SUM, C.FirstName, C.LastName  
FROM Order O JOIN Customer C ON O.CustomerId = C.Id  
GROUP BY C.FirstName, C.LastName  
ORDER BY SUM(O.TotalAmount) DESC
```

Sum	FirstName	LastName
117483.39	Horst	Kloss
115673.39	Jose	Pavarotti
113236.68	Roland	Mendel

SET Operators

```
SELECT * FROM InfoTable  
WHERE age = 40  
UNION  
SELECT * FROM InfoTable  
WHERE age = 45
```

```
SELECT * FROM InfoTable  
WHERE age = 40  
INTERSECT  
SELECT * FROM InfoTable  
WHERE age = 45
```

```
SELECT * FROM InfoTable  
WHERE age = 40  
MINUS  
SELECT * FROM InfoTable  
WHERE age = 45
```