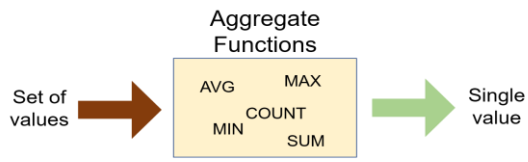
SQL
Session 2

Table of Contents

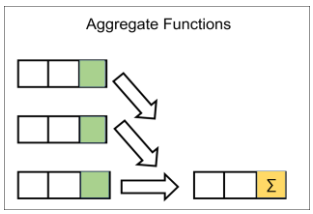
Aggregate Functions

SQL
What is an aggregate function?



Aggregate functions are functions that take a collection of values as input and return a single value

SQL
What is an aggregate function?



SUM and AVG → numeric values

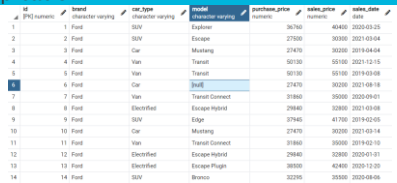
MIN, MAX, COUNT → numeric & non-numeric (strings, date, etc.)

We will learn GROUP BY clause and HAVING clause later.

What is NULL?

SQL
What is NULL?

`NULL` means no data and is a special value in SQL.
It shows us that a piece of information is unknown or missing or not applicable.



SQL
What is NULL?

- NULL value represents the unknown value or missing value or not applicable.
- NULL is not equal to zero or empty string.
- NULL is not equal to itself.

2 COUNT Function

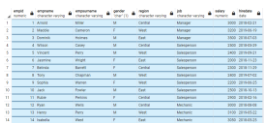
SQL
COUNT Function

We use COUNT function to count the numbers of records (a.k.a row) in a table.

SELECT COUNT (column_name)
FROM table_name;

COUNT Function

How many employees does the company have?



SELECT COUNT (*)
FROM employees;

	count bigint
1	14

COUNT Function

There is another special character returning the number of rows in a table. That is * character. Use it inside the COUNT function as **COUNT(*)**.

COUNT Function

An important point for **COUNT(*)** function is that the result table includes **NULL**. If you want the number of non-null values, use the syntax: **COUNT(column_name)**.

AS (Alias) Keyword

We can customize the column name or table name using **AS** keyword. AS is used to rename a column or table with an alias.
This is the syntax for aliasing a column name:
column_name [AS] alias_name

This is the syntax for aliasing a table name:
table_name [AS] alias_name

AS (Alias) Keyword

PRO TIP

AS keyword is optional. Most programmers specify the *AS* keyword when aliasing a column name, but not when aliasing a table name.

3 COUNT DISTINCT

COUNT DISTINCT

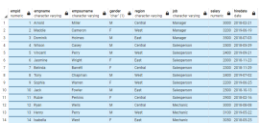
In some cases, we may want unique values. In those cases, we use COUNT DISTINCT function.

Syntax

COUNT (DISTINCT column_name)

COUNT DISTINCT

How many unique fields are there in the employees table?



SELECT COUNT (DISTINCT job)
FROM employees;

count
bigint
3

MIN and MAX

MIN Function

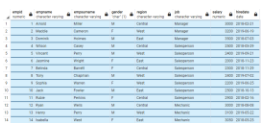
MIN function returns the minimum value in the selected column. The MIN function ignores the NULL values.

Syntax

SELECT MIN (column_name)
FROM table_name;

MIN Function

What is the lowest wage in the company?



SELECT MIN (salary)
FROM employees;

count
bigint
3

MAX Function

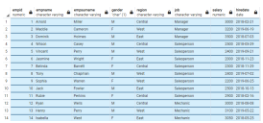
MAX function returns the maximum value in the selected column.

Syntax

SELECT MAX (column_name)
FROM table_name;

MAX Function

What is the last hired employees's date?



SELECT MAX (hiredate)
FROM employees;

SUM and AVG

SUM Function

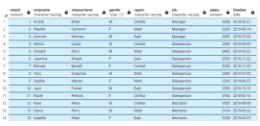
SUM function returns the sum of a numeric column.

Syntax

SELECT SUM (column_name)
FROM table_name;

SUM Function

What is total amount salary of the employees?



SELECT SUM (salary)
FROM employees;

sum
numeric
38050

2 AVG Function

AVG Function

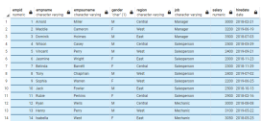
AVG function calculates the average of a numeric column.

Syntax

SELECT MAX (column_name)
FROM table_name;

AVG Function

What is the average salary of the employees?



SELECT AVG (salary)
FROM employees;

1 GROUP BY Clause

GROUP BY Clause

The GROUP BY clause groups the rows into summary rows. It returns one value for each group and is typically used with aggregate functions (COUNT, MAX, MIN, SUM, AVG).

		Gender	
		Male	
		Male	
		Female	
		Female	

COUNT(Gender) → 4

COUNT(Gender)
WHERE Gender = 'Male' → 2

COUNT(Gender)
WHERE Gender = 'Female' → 2

GROUP BY Clause

Syntax

```
SELECT column_1, aggregate_function(column_2)
FROM table_name
GROUP BY column_1;
```


GROUP BY Clause

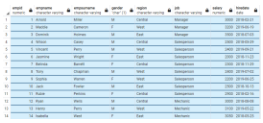
- GROUP BY returns only one result per group of data.
- GROUP BY Clause always follows the WHERE Clause.
- GROUP BY Clause always precedes the ORDER BY.

SELECT column1,aggregate_function(column2)
FROM tabel_name
GROUP BY column_1;

2 GROUP BY with COUNT Function

GROUP BY with COUNT Function

What is the number of employees per gender?



SELECT gender, **COUNT** (gender)
FROM employees
GROUP BY gender;

	gender "char" (1)	count bigint
1	F	6
2	M	8

GROUP BY Clause

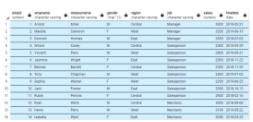
The GROUP BY clause groups results before calling the aggregate function. This allows you to apply aggregate function to groups than the entire query.

gender
Male
Male
Male
Male
Male
Male
Female
Female
Female
Female

gender	COUNT(gender)
Male	6
Female	4

GROUP BY with COUNT Function

What is the number of employees working as a salesperson broken by gender?



SELECT gender, **COUNT** (job)
FROM employees
WHERE job='Salesperson'
GROUP BY gender;

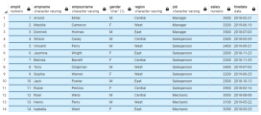
GROUP BY Clause

- WHERE clause operates on the data before the aggregation.
- WHERE clause happens before the GROUP BY clause.
- Only the rows that meet the conditions in the WHERE clause are grouped.

3 GROUP BY with MIN&MAX Functions

GROUP BY with MIN&MAX Functions

Let's find the minimum salaries of each gender group using the **MIN** function.

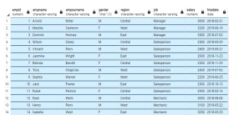


SELECT gender, **MIN** (salary)
FROM employees
GROUP BY gender;

gender 'char' (1)	min numeric
F	2000
M	2400

GROUP BY with MIN&MAX Functions

Similarly, we can find the maximum salaries of each group using the `MAX` function. You may also use the `ORDER BY` clause to sort the salaries in descending or ascending order. The `ORDER BY` follows `GROUP BY`. For instance, sort the maximum salaries in descending order.



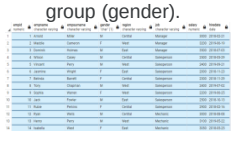
SELECT gender, **MAX** (salary) **AS** maxsalary
FROM employees
GROUP BY gender
ORDER BY maxsalary **DESC**;

gender 'char' (1)	maxsalary numeric
M	3500
F	3200

4 GROUP BY with SUM&AVG Functions

GROUP BY with SUM&AVG Functions

Let's calculate the total and avarage salaries of each group (gender).



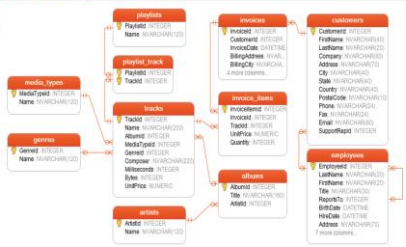
SELECT gender, **SUM** (salary) **AS** totalsalary,
AVG (salary) **AS** avgsalary
FROM employees
GROUP BY gender;

gender 'char' (1)	totalsalary numeric	avgsalary numeric
F	15650	2608
M	22400	2800

Query Time

CLARUSWAY©

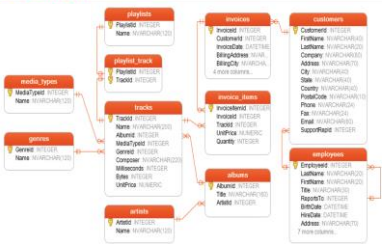
How many invoices are in the digital music store?



CLARUSWAY©

4
4

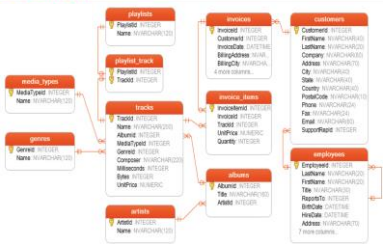
How many composers are there in the digital music store?



CLARUSWAY

6

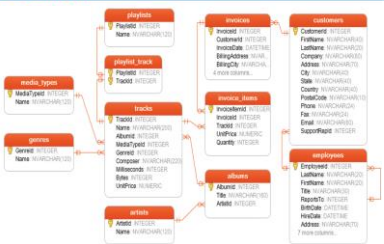
Find the track name having the minimum duration.



CLARUSWAY

48

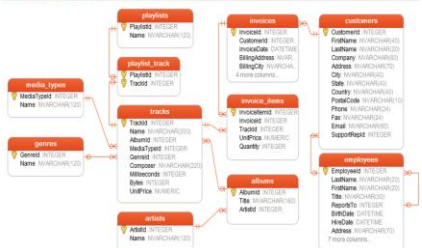
Find the track name having the maximum duration.



CLARUSWAY

47

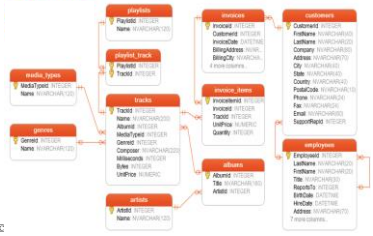
How much money did our store earn?



CLARUSWAY©

48

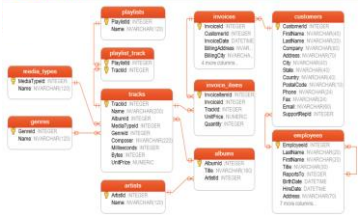
Find the tracks having duration bigger than the average duration.



CLARUSWAY©

49

Find the total number of each composer's track. Your result will include name of the composer and number.



CLARUSWAY©

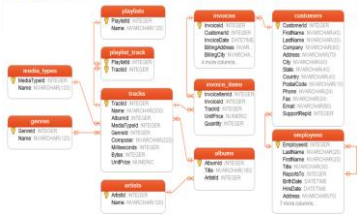
50

How many customers do we have from each country? Your result will include name of the country and number.

CLARUSWAY©

51

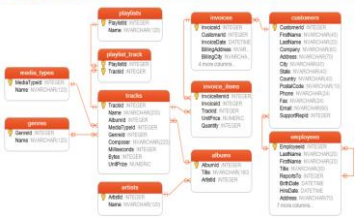
Find the minimum duration of track for each album. Your result will include album id and min duration.



CLARUSWAY©

52

Find the total amount of invoice for each country. Your result will include country name and total amount.



CLARUSWAY©

53

JOINs

CLARUSWAY©

Important Concepts

Primary Key (PK):
The primary key is a column in our table that makes each row (aka, record) unique.

Foreign Key (FK):
Foreign key is a column in a table that uniquely identifies each row of another table. That column refers to a primary key of another table. This creates a kind of link between the tables.

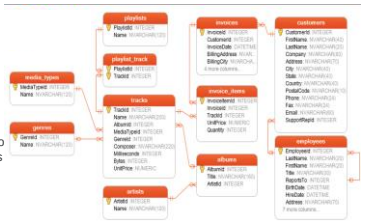


Table of Contents

1 Introduction

Introduction

A JOIN clause is used to combine two or more tables into a single table.

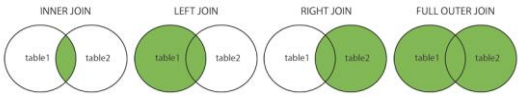
Joins are usually applied based on the keys that define the relationship between those tables or on common fields.

Introduction

In most cases this joins are created using the primary key of one table and the foreign key of the other table we want to join it with.

2 JOIN Types

JOIN Types

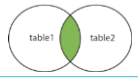


- **INNER JOIN:** Returns the common records in both tables.
- **LEFT OUTER JOIN:** Returns all records from the left table and matching records from the right table.
- **RIGHT OUTER JOIN:** Returns all records from the right table and matching records from the left table.
- **FULL OUTER JOIN:** Returns all records of both left and right tables.

3 INNER JOIN

INNER JOIN

INNER JOIN is the most common type of JOINs. The INNER JOIN selects records that have matching values in both tables. INNER keyword is optional for this type of JOIN.



Syntax

```
SELECT columns
  FROM table_A
  INNER JOIN table_B ON join_conditions
```

join_conditions

table_A. common_field = table_B.common_field

students

name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60

tests

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60


```
SELECT students.name, students.exam,
       students.score, tests.passing_score
FROM students
INNER JOIN tests ON students.exam = tests.exam;
```

students

name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60

tests

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60


```
SELECT students.name, students.exam,
       students.score, tests.passing_score
FROM students
INNER JOIN tests ON students.exam = tests.exam;
```

students **tests**

name	exam	score	exam	passing_score
John	SQL	75	SQL	70
Mary	AWS	80	AWS	80
Clark	Python	60	Python	70
			Network	60

```
SELECT students.name, students.exam,
       students.score, tests.passing_score
FROM students
INNER JOIN tests ON students.exam = tests.exam;
```

students

name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60

tests

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60


```
SELECT students.name, students.exam,
       students.score, tests.passing_score
FROM students
INNER JOIN tests ON students.exam = tests.exam;
```

students

name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60

tests

exam	passing_score
SQL	70
AWS	80
Python	70


```
SELECT students.name, students.exam,
       students.score, tests.passing_score
FROM students
INNER JOIN tests ON students.exam = tests.exam;
```


output of the query

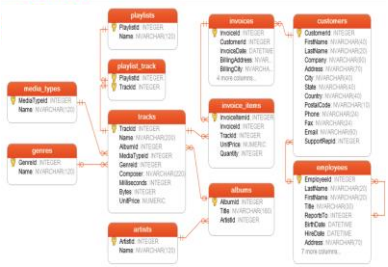
name	exam	score	passing_score
John	SQL	75	70
Mary	AWS	80	80
Clark	Python	60	70

INNER JOIN

Syntax of Join of Multiple Tables

```
SELECT columns
  FROM table_A
  INNER JOIN table_B
    ON join_conditions1 AND join_conditions2
  INNER JOIN table_C
    ON join_conditions3 OR join_conditions4
...
```


Find the genre of each track.



tracks

	TrackId	Name	AlbumId	MediaTypeId	GenreId
1	1	For Those About To Rock (We Salute You)	1	1	1
2	2	Balls to the Wall	2	2	1
3	3	Fast As a Shark	3	2	1
4	4	Restless and Wild	3	2	1
5	5	Princess of the Dawn	3	2	1
6	6	Put The Finger On You	1	1	1
7	7	Let's Get It Up	1	1	1
8	8	Inject The Venom	1	1	1
9	9	Snowballed	1	1	1
10	10	Evil Walks	1	1	1

genres

	GenreId	Name
1	1	Rock
2	2	Jazz
3	3	Metal
4	4	Alternative & Punk
5	5	Rock And Roll
6	6	Blues
7	7	Latin
8	8	Reggae
9	9	Pop
10	10	Soundtrack

tracks

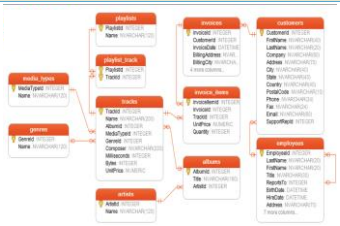
TrackId	Name	AlbumId	MediaTypeId	GenreId
1	For Those About To Rock (We Salute You)	1	1	1
2	Balls to the Wall	2	2	1
3	Fast As a Shark	3	2	1
4	Restless and Wild	3	2	1
5	Princess of the Dawn	3	2	1
6	Put The Finger On You	1	1	1
7	Let's Get It Up	1	1	1
8	Inject The Venom	1	1	1
9	Snowballed	1	1	1
10	Evil Walks	1	1	1

genres

GenreId	Name
1	Rock
2	Jazz
3	Metal
4	Alternative & Punk
5	Rock And Roll
6	Blues
7	Latin
8	Reggae
9	Pop
10	Soundtrack

	Name	Genre
1	For Those About To Rock (We Salute You)	Rock
2	Balls to the Wall	Rock
3	Fast As a Shark	Rock
4	Restless and Wild	Rock
5	Princess of the Dawn	Rock
6	Put The Finger On You	Rock
7	Let's Get It Up	Rock
8	Inject The Venom	Rock
9	Snowballed	Rock
10	Evil Walks	Rock

Find the customer name of each invoice.
Your result will include Invoice id and customer name.



CLARUSWAY©

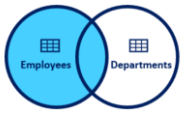
75

LEFT JOIN

CLARUSWAY©

LEFT JOIN

In this JOIN statement, all the records of the left table and the common records of the right table are returned in the query. If no matching rows are found in the right table during the JOIN operation, these values are assigned as NULL.



Visual Representation of Left JOIN

Syntax

```
SELECT columns
  FROM table_A
  LEFT JOIN table_B ON join_conditions
```

join_conditions

table_A. common_field = table_B.common_field

CLARUSWAY©

students

name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60

tests

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60

CLARUSWAY©

78

```
SELECT tests.exam, tests.passing_score,
       students.name, students.score
FROM tests
LEFT JOIN students ON tests.exam = students.exam;
```

tests

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60

students

name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60

CLARUSWAY©

79

```
SELECT tests.exam, tests.passing_score,
       students.name, students.score
FROM tests
LEFT JOIN students ON tests.exam = students.exam;
```

tests

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60

students

name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60


```
SELECT tests.exam, tests.passing_score,
       students.name, students.score
FROM tests
LEFT JOIN students ON tests.exam = students.exam;
```

tests

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60

students

name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60
Null	Null	Null


```
SELECT tests.exam, tests.passing_score,
       students.name, students.score
FROM tests
LEFT JOIN students ON tests.exam = students.exam;
```

tests

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60

students

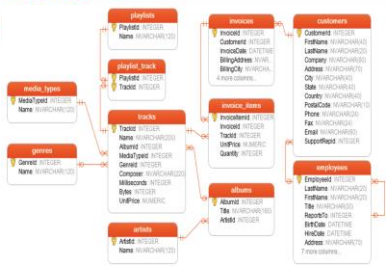
name	exam	score
John	SQL	75
Mary	AWS	80
Clark	Python	60
Null	Null	Null


```
SELECT tests.exam, tests.passing_score,
       students.name, students.score
FROM tests
LEFT JOIN students ON tests.exam = students.exam;
```

output of the query

exam	passing_score	name	score
SQL	70	John	75
AWS	80	Mary	80
Python	70	Clark	60
Network	60	Null	Null

Find the artists' album info



THANKS!

Any questions?