

# SQL Session 2



## Table of Contents

- ▶ What are type of Aggregate Functions, why do we need them?
- ▶ Group By Clause
- ▶ What are type of Joins in SQL, why do we need them

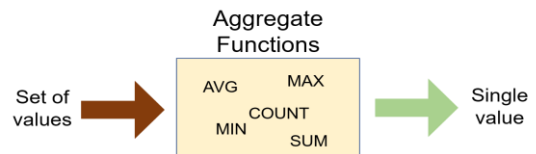
CLARUSWAY®

CLARUSWAY®

2

## Aggregate Functions

### SQL What is an aggregate function?



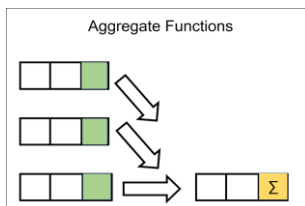
Aggregate functions are functions that take a collection of values as input and **return a single value**

CLARUSWAY®

CLARUSWAY®

4

### 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?

CLARUSWAY®

### 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.

ID	first_name	last_name	email	phone_number	hire_date	salary	commission_pct	department_id	job_id	manager_id	other_id
1	John	Deere	JD@DEERE	515 123 4567	1980-08-17	12000	0.1	10	Analyst		
2	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	1	
3	Lex	Deere	LEX@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	2	
4	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	3	
5	Lex	Deere	LEX@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	4	
6	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	5	
7	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	6	
8	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	7	
9	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	8	
10	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	9	
11	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	10	
12	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	11	
13	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	12	
14	Neena	Kochhar	NEENA.K@DEERE	515 123 4567	1989-09-01	15000	0.1	10	Analyst	13	

CLARUSWAY®

CLARUSWAY®

8

## 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

CLARUSWAY®

7

CLARUSWAY®

## 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;
```

CLARUSWAY®

8

CLARUSWAY®

## COUNT Function

How many employees does the company have?

```
SELECT COUNT (*)
FROM employees;
```

	empid	employee	employee	gender	region	job	salary	commission	manager	department
	1	James	Blake	M	West	Manager	9000	0	100	90
	2	Steven	Knight	M	West	Manager	8000	0	100	90
	3	Deena	Deena	F	East	Manager	7000	0	100	90
	4	Neena	Kochhar	F	Central	Analyst	4000	0	100	90
	5	Lex	Kochhar	M	Central	Analyst	3900	0	100	90
	6	Shelley	Kochhar	F	Central	Analyst	4500	0	100	90
	7	Pooja	Kochhar	F	Central	Analyst	3800	0	100	90
	8	Delia	Tavner	F	Central	Analyst	4200	0	100	90
	9	Anthony	John	M	Central	Analyst	4300	0	100	90
	10	Paul	Hern	M	Central	Analyst	4000	0	100	90
	11	Andrew	Werner	M	Central	Analyst	4200	0	100	90
	12	Jean	DeLo	F	Central	Analyst	4100	0	100	90
	13	Wendy	Willamson	F	Central	Analyst	4000	0	100	90
	14	April	Reynolds	F	Central	Analyst	3900	0	100	90
	15	Jacob	Furman	M	Central	Analyst	4000	0	100	90
	16	Guy	DeLo	M	Central	Analyst	3900	0	100	90
	17	Clay	DeLo	M	Central	Analyst	3800	0	100	90
	18	John	Smith	M	Central	Analyst	4000	0	100	90
	19	Chris	Watson	M	Central	Analyst	4000	0	100	90
	20	Bruce	Smith	M	Central	Analyst	4000	0	100	90
	21	David	Watson	M	Central	Analyst	4100	0	100	90
	22	Allen	Ward	M	Central	Analyst	4200	0	100	90
	23	Ward	Ward	M	Central	Analyst	4200	0	100	90
	24	Amanda	Ward	F	Central	Analyst	4200	0	100	90
	25	Ward	Ward	M	Central	Analyst	4200	0	100	90
	26	Ward	Ward	M	Central	Analyst	4200	0	100	90
	27	Ward	Ward	M	Central	Analyst	4200	0	100	90
	28	Ward	Ward	M	Central	Analyst	4200	0	100	90
	29	Ward	Ward	M	Central	Analyst	4200	0	100	90
	30	Ward	Ward	M	Central	Analyst	4200	0	100	90
	31	Ward	Ward	M	Central	Analyst	4200	0	100	90
	32	Ward	Ward	M	Central	Analyst	4200	0	100	90
	33	Ward	Ward	M	Central	Analyst	4200	0	100	90
	34	Ward	Ward	M	Central	Analyst	4200	0	100	90
	35	Ward	Ward	M	Central	Analyst	4200	0	100	90
	36	Ward	Ward	M	Central	Analyst	4200	0	100	90
	37	Ward	Ward	M	Central	Analyst	4200	0	100	90
	38	Ward	Ward	M	Central	Analyst	4200	0	100	90
	39	Ward	Ward	M	Central	Analyst	4200	0	100	90
	40	Ward	Ward	M	Central	Analyst	4200	0	100	90
	41	Ward	Ward	M	Central	Analyst	4200	0	100	90
	42	Ward	Ward	M	Central	Analyst	4200	0	100	90
	43	Ward	Ward	M	Central	Analyst	4200	0	100	90
	44	Ward	Ward	M	Central	Analyst	4200	0	100	90
	45	Ward	Ward	M	Central	Analyst	4200	0	100	90
	46	Ward	Ward	M	Central	Analyst	4200	0	100	90
	47	Ward	Ward	M	Central	Analyst	4200	0	100	90
	48	Ward	Ward	M	Central	Analyst	4200	0	100	90
	49	Ward	Ward	M	Central	Analyst	4200	0	100	90
	50	Ward	Ward	M	Central	Analyst	4200	0	100	90
	51	Ward	Ward	M	Central	Analyst	4200	0	100	90
	52	Ward	Ward	M	Central	Analyst	4200	0	100	90
	53	Ward	Ward	M	Central	Analyst	4200	0	100	90
	54	Ward	Ward	M	Central	Analyst	4200	0	100	90
	55	Ward	Ward	M	Central	Analyst	4200	0	100	90
	56	Ward	Ward	M	Central	Analyst	4200	0	100	90
	57	Ward	Ward	M	Central	Analyst	4200	0	100	90
	58	Ward	Ward	M	Central	Analyst	4200	0	100	90
	59	Ward	Ward	M	Central	Analyst	4200	0	100	90
	60	Ward	Ward	M	Central	Analyst	4200	0	100	90
	61	Ward	Ward	M	Central	Analyst	4200	0	100	90
	62	Ward	Ward	M	Central	Analyst	4200	0	100	90
	63	Ward	Ward	M	Central	Analyst	4200	0	100	90
	64	Ward	Ward	M	Central	Analyst	4200	0	100	90
	65	Ward	Ward	M	Central	Analyst	4200	0	100	90
	66	Ward	Ward	M	Central	Analyst	4200	0	100	90
	67	Ward	Ward	M	Central	Analyst	4200	0	100	90
	68	Ward	Ward	M	Central	Analyst	4200	0	100	90
	69	Ward	Ward	M	Central	Analyst	4200	0	100	90
	70	Ward	Ward	M	Central	Analyst	4200	0	100	90
	71	Ward	Ward	M	Central	Analyst	4200	0	100	90
	72	Ward	Ward	M	Central	Analyst	4200	0	100	90
	73	Ward	Ward	M	Central	Analyst	4200	0	100	90
	74	Ward	Ward	M	Central	Analyst	4200	0	100	90
	75	Ward	Ward	M	Central	Analyst	4200	0	100	90
	76	Ward	Ward	M	Central	Analyst	4200	0	100	90
	77	Ward	Ward	M	Central	Analyst	4200	0	100	90
	78	Ward	Ward	M	Central	Analyst	4200	0	100	90
	79	Ward	Ward	M	Central	Analyst	4200	0	100	90
	80	Ward	Ward	M	Central	Analyst	4200	0	100	90
	81	Ward	Ward	M	Central	Analyst	4200	0	100	90
	82	Ward	Ward	M	Central	Analyst	4200	0	100	90
	83	Ward	Ward	M	Central	Analyst	4200	0	100	90
	84	Ward	Ward	M	Central	Analyst	4200	0	100	90
	85	Ward	Ward	M	Central	Analyst	4200	0	100	90
	86	Ward	Ward	M	Central	Analyst	4200	0	100	90
	87	Ward	Ward	M	Central	Analyst	4200	0	100	90
	88	Ward	Ward	M	Central	Analyst	4200	0	100	90
	89	Ward	Ward	M	Central	Analyst	4200	0	100	90
	90	Ward	Ward	M	Central	Analyst	4200	0	100	90
	91	Ward	Ward	M	Central	Analyst	4200	0	100	90
	92	Ward	Ward	M	Central	Analyst	4200	0	100	90
	93	Ward	Ward	M	Central	Analyst	4200	0	100	90
	94	Ward	Ward	M	Central	Analyst	4200	0	100	90
	95	Ward	Ward	M	Central	Analyst	4200	0	100	90
	96	Ward	Ward	M	Central	Analyst	4200	0	100	90
	97	Ward	Ward	M	Central	Analyst	4200	0	100	90
	98	Ward	Ward	M	Central	Analyst	4200	0	100	90
	99	Ward	Ward	M	Central	Analyst	4200	0	100	90
	100	Ward	Ward	M	Central	Analyst	4200	0	100	90
	101	Ward	Ward	M	Central	Analyst	4200	0	100	90
	102	Ward	Ward	M	Central	Analyst	4200	0	100	90
	103	Ward	Ward	M	Central	Analyst	4200	0	100	90
	104	Ward	Ward	M	Central	Analyst	4200	0	100	90
	105	Ward	Ward	M	Central	Analyst	4200	0	100	90
	106	Ward	Ward	M	Central	Analyst	4200	0	100	90
	107	Ward	Ward	M	Central	Analyst	4200	0	100	90
	108	Ward	Ward	M	Central	Analyst	4200	0	100	90
	109	Ward	Ward	M	Central	Analyst	4200	0	100	90
	110	Ward	Ward	M	Central	Analyst	4200	0	100	90
	111	Ward	Ward	M	Central	Analyst	4200	0	100	90
	112	Ward	Ward	M	Central	Analyst	4200	0	100	90
	113	Ward	Ward	M	Central	Analyst	4200	0	100	90
	114	Ward	Ward	M	Central	Analyst	4200	0	100	90
	115	Ward	Ward	M	Central	Analyst	4200	0	100	90
	116	Ward	Ward	M	Central	Analyst	4200	0	100	90
	117	Ward	Ward	M	Central	Analyst	4200	0	100	90
	118	Ward	Ward	M	Central	Analyst	4200	0	100	90
	119	Ward	Ward	M	Central	Analyst	4200	0	100	90
	120	Ward	Ward	M	Central	Analyst	4200	0	100	90

## 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**

CLARUSWAY®  
ABOUT CLARUSWAY TECHNOLOGIES



## AS (Alias) Keyword



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

CLARUSWAY®  
ABOUT CLARUSWAY TECHNOLOGIES

13

14

## 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)**

CLARUSWAY®  
ABOUT CLARUSWAY TECHNOLOGIES

CLARUSWAY®  
ABOUT CLARUSWAY TECHNOLOGIES

16

## COUNT DISTINCT



How many unique fields are there in the employees table?

```
SELECT COUNT (DISTINCT job)
FROM employees;
```

EMPID	EMPNAME	DEPARTMENT	JOBTITLE	MANAGER	STARTDATE	ENDDATE	STATUS	EXPIRYDATE
1	Scott	AD	Analyst	1000	2000-04-19	2000-04-19	ACTIVE	2000-04-19
2	Deena	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
3	Alex	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
4	Julia	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
5	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
6	Jack	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
7	Pat	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
8	Neena	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
9	Lex	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
10	Dena	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
11	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
12	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
13	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
14	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
15	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
16	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
17	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
18	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
19	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
20	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
21	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
22	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
23	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
24	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
25	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
26	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
27	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
28	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
29	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
30	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
31	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
32	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
33	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
34	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
35	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
36	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
37	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
38	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
39	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
40	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
41	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
42	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
43	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
44	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
45	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
46	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
47	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
48	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
49	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
50	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
51	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
52	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
53	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
54	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
55	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
56	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
57	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
58	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
59	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
60	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
61	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
62	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
63	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
64	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
65	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
66	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
67	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
68	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
69	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
70	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
71	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
72	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
73	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
74	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
75	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
76	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
77	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
78	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
79	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
80	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
81	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
82	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
83	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
84	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
85	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
86	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
87	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
88	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
89	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
90	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
91	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
92	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
93	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
94	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
95	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
96	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
97	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
98	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
99	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01
100	John	AD	Analyst	1000	2000-05-01	2000-05-01	ACTIVE	2000-05-01

count  
bigint  
3

CLARUSWAY®  
ABOUT CLARUSWAY TECHNOLOGIES

CLARUSWAY®  
ABOUT CLARUSWAY TECHNOLOGIES

17

## 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;
```

CLARUSWAY®  
ANY IT CHALLENGE IS SOLVABLE



## MIN Function

What is the lowest wage in the company?

```
SELECT MIN (salary)
FROM employees;
```

id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct
1	Scott	Tyler	STYLER	515.123.4567	1987-06-08	Analyst	1100	0.10
2	Deena	Fawcett	DFAWCETT	515.123.4568	1987-06-13	Analyst	1200	0.10
3	Neena	Kochhar	NKOCHHAR	515.123.4569	1987-06-22	Analyst	1300	0.10
4	Lex	DeHaan	LDEHAAN	515.123.4570	1987-07-17	Analyst	1400	0.10
5	Rena	Lot	RL	515.123.4571	1987-07-21	Analyst	1500	0.10
6	Kevin	Timothy	KTIMOTHY	515.123.4572	1987-07-27	Analyst	1600	0.10
7	John	Smith	JS	515.123.4573	1987-08-03	Analyst	1700	0.10
8	Joshua	Patel	JAPATEL	515.123.4574	1987-08-09	Analyst	1800	0.10
9	Timothy	Gaughan	TGAUGHAN	515.123.4575	1987-08-17	Analyst	1900	0.10
10	David	Turner	DTURNER	515.123.4576	1987-08-28	Analyst	2000	0.10
11	Walter	King	WKING	515.123.4577	1987-09-07	Analyst	2100	0.10
12	Ismael	Sciarra	ISCIARRA	515.123.4578	1987-09-16	Analyst	2200	0.10
13	Helena	Sullivan	HSULLIVAN	515.123.4579	1987-09-19	Analyst	2300	0.10
14	Gordon	Everett	GE	515.123.4580	1987-09-28	Analyst	2400	0.10
15	Adam	Smith	AS	515.123.4581	1987-10-07	Analyst	2500	0.10
16	Janina	Barber	JB	515.123.4582	1987-10-13	Analyst	2600	0.10
17	Shelley	Stevens	SS	515.123.4583	1987-10-19	Analyst	2700	0.10
18	Wendell	Olson	WO	515.123.4584	1987-10-28	Analyst	2800	0.10
19	Timothy	Gietz	TG	515.123.4585	1987-11-07	Analyst	2900	0.10
20	Jack	Fey	JF	515.123.4586	1987-11-16	Analyst	3000	0.10
21	Greg	King	GK	515.123.4587	1987-11-27	Analyst	3100	0.10
22	Cheryl	Winters	CW	515.123.4588	1987-12-08	Analyst	3200	0.10
23	David	Adams	DA	515.123.4589	1987-12-14	Analyst	3300	0.10
24	Julia	Abel	JA	515.123.4590	1987-12-15	Analyst	3400	0.10
25	Keith	Johnson	KJ	515.123.4591	1987-12-18	Analyst	3500	0.10
26	Timothy	Maxam	TMAXAM	515.123.4592	1987-12-27	Analyst	3600	0.10
27	Wendell	Olson	WO	515.123.4593	1988-01-05	Analyst	3700	0.10
28	Phyllis	Wheeler	PWHEELER	515.123.4594	1988-01-09	Analyst	3800	0.10
29	Debra	Fisk	DFISK	515.123.4595	1988-01-22	Analyst	3900	0.10
30	Michael	Harvey	MHARVEY	515.123.4596	1988-02-03	Analyst	4000	0.10
31	Robert	DeNino	RDENINO	515.123.4597	1988-02-04	Analyst	4100	0.10
32	Shelley	Stevens	SS	515.123.4598	1988-02-05	Analyst	4200	0.10
33	Wendell	Olson	WO	515.123.4599	1988-02-06	Analyst	4300	0.10
34	Phyllis	Wheeler	PWHEELER	515.123.4600	1988-02-07	Analyst	4400	0.10
35	Debra	Fisk	DFISK	515.123.4601	1988-02-08	Analyst	4500	0.10
36	Michael	Harvey	MHARVEY	515.123.4602	1988-02-09	Analyst	4600	0.10
37	Robert	DeNino	RDENINO	515.123.4603	1988-02-10	Analyst	4700	0.10
38	Shelley	Stevens	SS	515.123.4604	1988-02-11	Analyst	4800	0.10
39	Wendell	Olson	WO	515.123.4605	1988-02-12	Analyst	4900	0.10
40	Phyllis	Wheeler	PWHEELER	515.123.4606	1988-02-13	Analyst	5000	0.10
41	Debra	Fisk	DFISK	515.123.4607	1988-02-14	Analyst	5100	0.10
42	Michael	Harvey	MHARVEY	515.123.4608	1988-02-15	Analyst	5200	0.10
43	Robert	DeNino	RDENINO	515.123.4609	1988-02-16	Analyst	5300	0.10
44	Shelley	Stevens	SS	515.123.4610	1988-02-17	Analyst	5400	0.10
45	Wendell	Olson	WO	515.123.4611	1988-02-18	Analyst	5500	0.10
46	Phyllis	Wheeler	PWHEELER	515.123.4612	1988-02-19	Analyst	5600	0.10
47	Debra	Fisk	DFISK	515.123.4613	1988-02-20	Analyst	5700	0.10
48	Michael	Harvey	MHARVEY	515.123.4614	1988-02-21	Analyst	5800	0.10
49	Robert	DeNino	RDENINO	515.123.4615	1988-02-22	Analyst	5900	0.10
50	Shelley	Stevens	SS	515.123.4616	1988-02-23	Analyst	6000	0.10
51	Wendell	Olson	WO	515.123.4617	1988-02-24	Analyst	6100	0.10
52	Phyllis	Wheeler	PWHEELER	515.123.4618	1988-02-25	Analyst	6200	0.10
53	Debra	Fisk	DFISK	515.123.4619	1988-02-26	Analyst	6300	0.10
54	Michael	Harvey	MHARVEY	515.123.4620	1988-02-27	Analyst	6400	0.10
55	Robert	DeNino	RDENINO	515.123.4621	1988-02-28	Analyst	6500	0.10
56	Shelley	Stevens	SS	515.123.4622	1988-02-29	Analyst	6600	0.10
57	Wendell	Olson	WO	515.123.4623	1988-03-01	Analyst	6700	0.10
58	Phyllis	Wheeler	PWHEELER	515.123.4624	1988-03-02	Analyst	6800	0.10
59	Debra	Fisk	DFISK	515.123.4625	1988-03-03	Analyst	6900	0.10
60	Michael	Harvey	MHARVEY	515.123.4626	1988-03-04	Analyst	7000	0.10
61	Robert	DeNino	RDENINO	515.123.4627	1988-03-05	Analyst	7100	0.10
62	Shelley	Stevens	SS	515.123.4628	1988-03-06	Analyst	7200	0.10
63	Wendell	Olson	WO	515.123.4629	1988-03-07	Analyst	7300	0.10
64	Phyllis	Wheeler	PWHEELER	515.123.4630	1988-03-08	Analyst	7400	0.10
65	Debra	Fisk	DFISK	515.123.4631	1988-03-09	Analyst	7500	0.10
66	Michael	Harvey	MHARVEY	515.123.4632	1988-03-10	Analyst	7600	0.10
67	Robert	DeNino	RDENINO	515.123.4633	1988-03-11	Analyst	7700	0.10
68	Shelley	Stevens	SS	515.123.4634	1988-03-12	Analyst	7800	0.10
69	Wendell	Olson	WO	515.123.4635	1988-03-13	Analyst	7900	0.10
70	Phyllis	Wheeler	PWHEELER	515.123.4636	1988-03-14	Analyst	8000	0.10
71	Debra	Fisk	DFISK	515.123.4637	1988-03-15	Analyst	8100	0.10
72	Michael	Harvey	MHARVEY	515.123.4638	1988-03-16	Analyst	8200	0.10
73	Robert	DeNino	RDENINO	515.123.4639	1988-03-17	Analyst	8300	0.10
74	Shelley	Stevens	SS	515.123.4640	1988-03-18	Analyst	8400	0.10
75	Wendell	Olson	WO	515.123.4641	1988-03-19	Analyst	8500	0.10
76	Phyllis	Wheeler	PWHEELER	515.123.4642	1988-03-20	Analyst	8600	0.10
77	Debra	Fisk	DFISK	515.123.4643	1988-03-21	Analyst	8700	0.10
78	Michael	Harvey	MHARVEY	515.123.4644	1988-03-22	Analyst	8800	0.10
79	Robert	DeNino	RDENINO	515.123.4645	1988-03-23	Analyst	8900	0.10
80	Shelley	Stevens	SS	515.123.4646	1988-03-24	Analyst	9000	0.10
81	Wendell	Olson	WO	515.123.4647	1988-03-25	Analyst	9100	0.10
82	Phyllis	Wheeler	PWHEELER	515.123.4648	1988-03-26	Analyst	9200	0.10
83	Debra	Fisk	DFISK	515.123.4649	1988-03-27	Analyst	9300	0.10
84	Michael	Harvey	MHARVEY	515.123.4650	1988-03-28	Analyst	9400	0.10
85	Robert	DeNino	RDENINO	515.123.4651	1988-03-29	Analyst	9500	0.10
86	Shelley	Stevens	SS	515.123.4652	1988-03-30	Analyst	9600	0.10
87	Wendell	Olson	WO	515.123.4653	1988-03-31	Analyst	9700	0.10
88	Phyllis	Wheeler	PWHEELER	515.123.4654	1988-04-01	Analyst	9800	0.10
89	Debra	Fisk	DFISK	515.123.4655	1988-04-02	Analyst	9900	0.10
90	Michael	Harvey	MHARVEY	515.123.4656	1988-04-03	Analyst	10000	0.10

count  
bigint  
3

CLARUSWAY®  
ANY IT CHALLENGE IS SOLVABLE

## MAX Function

**MAX** function returns the maximum value in the selected column.

### Syntax

```
SELECT MAX (column_name)
FROM table_name;
```

CLARUSWAY®  
ANY IT CHALLENGE IS SOLVABLE



## MAX Function

What is the last hired employees's date?

```
SELECT MAX (hiredate)
FROM employees;
```

empid	employee	employee	gender	region	job	salary	bonus
number	name	character varying	code	code	character varying	numeric	numeric
1	Scott	Tyler	M	West	Manager	1100	120.00
2	Deena	Fawcett	F	West	Manager	1200	130.00
3	Neena	Kochhar	F	West	Manager	1300	140.00
4	Lex	DeHaan	M	West	Manager	1400	150.00
5	Rena	Lot	F	Central	Supervisor	1500	160.00
6	Kevin	Timothy	M	Central	Supervisor	1600	170.00
7	John	Smith	F	Central	Supervisor	1700	180.00
8	Joshua	Patel	F	Central	Supervisor	1800	190.00
9	Timothy	Gaughan	M	Central	Supervisor	1900	200.00
10	David	Turner	M	Central	Supervisor	2000	210.00
11	Walter	King	F	Central	Supervisor	2100	220.00
12	Ismael	Sciarra	M	Central	Supervisor	2200	230.00
13	Helena	Sullivan	F	Central	Supervisor	2300	240.00
14	Gordon	Everett	M	Central	Supervisor	2400	250.00
15	Adam	Smith	M	Central	Supervisor	2500	260.00
16	Janina	Barber	F	Central	Supervisor	2600	270.00
17	Shelley	Stevens	F	Central	Supervisor	2700	280.00
18	Wendell	Olson	M	Central	Supervisor	2800	290.00
19	Timothy	Gietz	M	Central	Supervisor	2900	300.00
20	Jack	Fey	M	Central	Supervisor	3000	310.00
21	Greg	King	M	Central	Supervisor	3100	320.00
22	Cheryl	Winters	F	Central	Supervisor	3200	330.00
23	David	Adams	M	Central	Supervisor	3300	340.00
24	Julia	Abel	F	Central	Supervisor	3400	350.00
25	Keith	Johnson	M	Central	Supervisor	3500	360.00
26	Timothy	Maxam	M	Central	Supervisor	3600	370.00
27	Wendell	Olson	M	Central	Supervisor	3700	380.00
28	Phyllis	Wheeler	F	Central	Supervisor	3800	390.00
29	Debra	Fisk	F	Central	Supervisor	3900	400.00
30	Michael	Harvey	M	Central	Supervisor	4000	410.00
31	Robert	DeNino	M	Central	Supervisor	4100	420.00
32	Shelley	Stevens	F	Central	Supervisor	4200	430.00
33	Wendell	Olson	M	Central	Supervisor	4300	440.00
34	Phyllis	Wheeler	F	Central	Supervisor	4400	450.00
35	Debra	Fisk	F	Central	Supervisor	4500	460.00

## SUM Function

What is total amount salary of the employees?

emp_id	last_name	first_name	email	phone_number	hire_date	job_id	salary
1	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
2	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
3	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
4	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
5	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
6	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
7	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
8	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
9	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
10	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
11	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
12	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
13	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
14	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000

```
SELECT SUM (salary)
FROM employees;
```

sum  
numeric  
38050

CLARUSWAY®  
DON'T LET YOUR KNOWLEDGE GO TO WASTE

25

CLARUSWAY®  
DON'T LET YOUR KNOWLEDGE GO TO WASTE

## 2 AVG Function

## AVG Function

AVG function calculates the average of a numeric column.

Syntax

```
SELECT MAX (column_name)
FROM table_name;
```

CLARUSWAY®  
DON'T LET YOUR KNOWLEDGE GO TO WASTE

27

CLARUSWAY®  
DON'T LET YOUR KNOWLEDGE GO TO WASTE

## AVG Function

What is the average salary of the employees?

emp_id	last_name	first_name	email	phone_number	hire_date	job_id	salary
1	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
2	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
3	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
4	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
5	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
6	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
7	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
8	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
9	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
10	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
11	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
12	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
13	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000
14	Scott	Ty	TSCOTT	515.123.4567	1987-06-08	ANALYST	3000

```
SELECT AVG (salary)
FROM employees;
```

CLARUSWAY®  
DON'T LET YOUR KNOWLEDGE GO TO WASTE

27

CLARUSWAY®  
DON'T LET YOUR KNOWLEDGE GO TO WASTE

28

## 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	COUNT(Gender)	→	4
		Male	COUNT(Gender) WHERE Gender = 'Male'	→	2
		Female	COUNT(Gender) WHERE Gender = 'Female'	→	2

CLARUSWAY®  
DON'T LET YOUR KNOWLEDGE GO TO WASTE

CLARUSWAY®  
DON'T LET YOUR KNOWLEDGE GO TO WASTE

30

## GROUP BY Clause

Syntax

```
1 SELECT column_1, aggregate_function(column_2)
2 FROM table_name
3 GROUP BY column_1;
4
```

CLARUSWAY®  
ABOUT US | SERVICES | TECHNOLOGY



## 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 table_name
GROUP BY column_1;
```

CLARUSWAY®  
ABOUT US | SERVICES | TECHNOLOGY

32

## 2 GROUP BY with COUNT Function



## GROUP BY with COUNT Function



What is the number of employees per gender?

id	name	gender	age	salary	job	dept	empid	managerid	hiredate	salary	commission_pct	empid	managerid	hiredate	salary	commission_pct
1	John	Male	35	12000	Manager	10	1000		2000-09-01	12000	0.10	2	1000	2000-09-01	9000	0.10
2	Jane	Female	36	9000	Manager	10	2000		2000-09-01	9000	0.10	3	2000	2000-09-01	6000	0.10
3	Michael	Male	37	8000	Manager	10	3000		2000-09-01	8000	0.10	4	3000	2000-09-01	7000	0.10
4	Pat	Female	38	6000	Manager	10	4000		2000-09-01	6000	0.10	5	4000	2000-09-01	5000	0.10
5	Scott	Male	39	7000	Manager	10	5000		2000-09-01	7000	0.10	6	5000	2000-09-01	4000	0.10
6	Timothy	Male	40	6000	Manager	10	6000		2000-09-01	6000	0.10	7	6000	2000-09-01	5000	0.10
7	David	Male	41	5000	Manager	10	7000		2000-09-01	5000	0.10	8	7000	2000-09-01	4000	0.10
8	Michael	Male	42	4000	Manager	10	8000		2000-09-01	4000	0.10	9	8000	2000-09-01	3000	0.10
9	Pat	Female	43	3000	Manager	10	9000		2000-09-01	3000	0.10	10	9000	2000-09-01	2000	0.10
10	Scott	Male	44	2000	Manager	10	10000		2000-09-01	2000	0.10	11	10000	2000-09-01	1000	0.10
11	Timothy	Male	45	1000	Manager	10	11000		2000-09-01	1000	0.10	12	11000	2000-09-01	500	0.10
12	David	Male	46	500	Manager	10	12000		2000-09-01	500	0.10	13	12000	2000-09-01	250	0.10
13	Michael	Male	47	250	Manager	10	13000		2000-09-01	250	0.10	14	13000	2000-09-01	125	0.10

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

gender	count
1 F	6
2 M	8

CLARUSWAY®  
ABOUT US | SERVICES | TECHNOLOGY

CLARUSWAY®  
ABOUT US | SERVICES | TECHNOLOGY

34

## 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

CLARUSWAY®  
ABOUT US | SERVICES | TECHNOLOGY



## GROUP BY with COUNT Function



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

id	name	gender	age	salary	job	dept	empid	managerid	hiredate	salary	commission_pct	empid	managerid	hiredate	salary	commission_pct
1	John	Male	35	12000	Manager	10	1000		2000-09-01	12000	0.10	2	1000	2000-09-01	9000	0.10
2	Jane	Female	36	9000	Manager	10	2000		2000-09-01	9000	0.10	3	2000	2000-09-01	6000	0.10
3	Michael	Male	37	8000	Manager	10	3000		2000-09-01	8000	0.10	4	3000	2000-09-01	7000	0.10
4	Pat	Female	38	6000	Manager	10	4000		2000-09-01	6000	0.10	5	4000	2000-09-01	5000	0.10
5	Scott	Male	39	7000	Manager	10	5000		2000-09-01	7000	0.10	6	5000	2000-09-01	4000	0.10
6	Timothy	Male	40	6000	Manager	10	6000		2000-09-01	6000	0.10	7	6000	2000-09-01	5000	0.10
7	David	Male	41	5000	Manager	10	7000		2000-09-01	5000	0.10	8	7000	2000-09-01	4000	0.10
8	Michael	Male	42	4000	Manager	10	8000		2000-09-01	4000	0.10	9	8000	2000-09-01	3000	0.10
9	Pat	Female	43	3000	Manager	10	9000		2000-09-01	3000	0.10	10	9000	2000-09-01	2000	0.10
10	Scott	Male	44	2000	Manager	10	10000		2000-09-01	2000	0.10	11	10000	2000-09-01	1000	0.10
11	Timothy	Male	45	1000	Manager	10	11000		2000-09-01	1000	0.10	12	11000	2000-09-01	500	0.10
12	David	Male	46	500	Manager	10	12000		2000-09-01	500	0.10	13	12000	2000-09-01	250	0.10
13	Michael	Male	47	250	Manager	10	13000		2000-09-01	250	0.10	14	13000	2000-09-01	125	0.10

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

gender	count
1 F	4
2 M	1

CLARUSWAY®  
ABOUT US | SERVICES | TECHNOLOGY

CLARUSWAY®  
ABOUT US | SERVICES | TECHNOLOGY

38

## 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

CLARUSWAY®

37

CLARUSWAY®

## 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

CLARUSWAY®

38

CLARUSWAY®

## 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

CLARUSWAY®

40

## GROUP BY with SUM&AVG Functions

Let's calculate the total and average 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

## 4 GROUP BY with SUM&AVG Functions

CLARUSWAY®

CLARUSWAY®

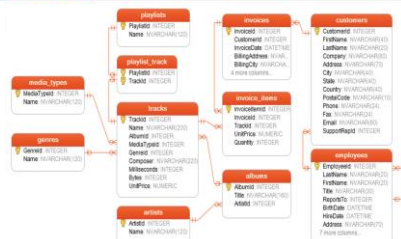
42

## Query Time



CLARUSWAY®  
NOT AN EMPLOYEE OF SAP

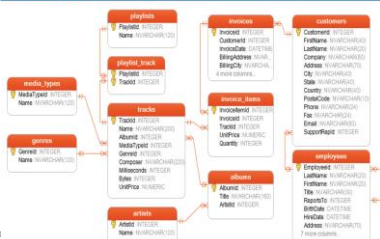
### How many invoices are in the digital music store?



CLARUSWAY®  
NOT AN EMPLOYEE OF SAP

4

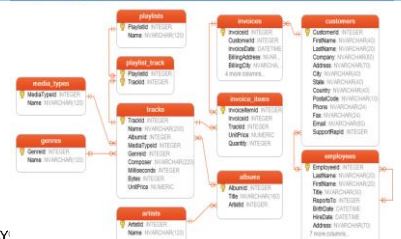
### How many composers are there in the digital music store?



CLARUSWAY®  
NOT AN EMPLOYEE OF SAP

4

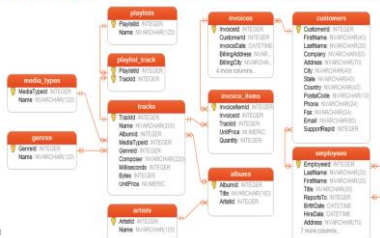
### Find the track name having the minimum duration.



CLARUSWAY®  
NOT AN EMPLOYEE OF SAP

46

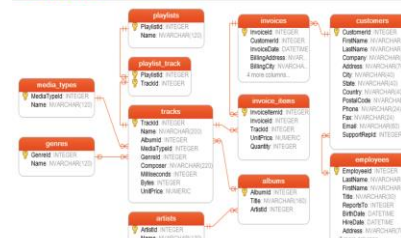
### Find the track name having the maximum duration.



CLARUSWAY®  
NOT AN EMPLOYEE OF SAP

47

### How much money did our store earn?

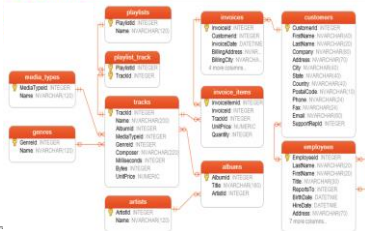


CLARUSWAY®  
NOT AN EMPLOYEE OF SAP

48



Find the tracks having duration bigger than the average duration.



CLARUSWAY®

48

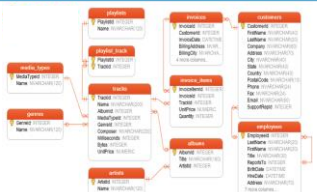
Find the total number of each composer's track. Your result will include name of the composer and number.



CLARUSWAY®

50

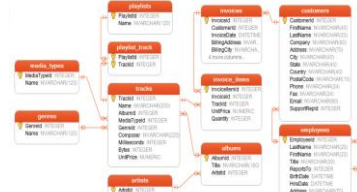
How many customers do we have from each country? Your result will include name of the country and number.



CLARUSWAY®

51

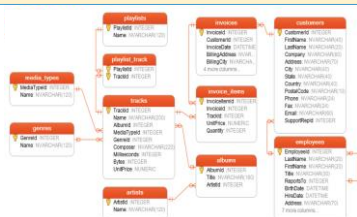
Find the minimum duration of track for each album. Your result will include album id and min duration.



CLARUSWAY®

52

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



CLARUSWAY®

53

JOINS

CLARUSWAY®

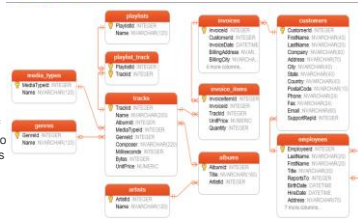
### Important Concepts

#### 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

- Introduction
- JOIN Types
- Inner JOIN
- Left JOIN

## 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.

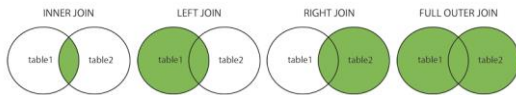
### ► Introduction



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.

CLARUSWAY®

81

CLARUSWAY®

### 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

```
1 SELECT columns
2 FROM table_A
3 INNER JOIN table_B ON join_conditions
```

join\_conditions  
table\_A.common\_field = table\_B.common\_field



CLARUSWAY®

83

CLARUSWAY®

84

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

tests

CLARUSWAY®

85

CLARUSWAY®

86

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

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

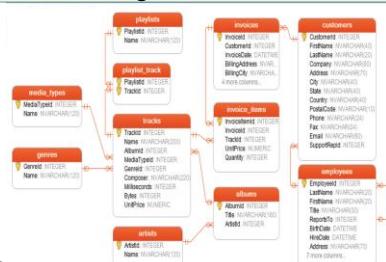
name	exam	score	passing_score
John	SQL	75	70
Mary	AWS	80	80
Clark	Python	60	70

## ▶ INNER JOIN

### Syntax of Join of Multiple Tables

```
1 SELECT columns
2 FROM table_A
3 INNER JOIN table_B
4     ON join_conditions1 AND join_conditions2
5 INNER JOIN table_C
6     ON join_conditions3 OR join_conditions4
7 ...
```

**Find the genre of each track.**



## tracks

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

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	Raiders of the Wild	3	2	1
5	Process 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	Ignite The Nation	1	1	1
9	Snowballad	1	1	1
10	Evil Walks	1	1	1

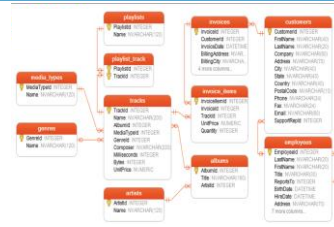
  

GenreId	Name
1	Rock
2	Jazz
3	Metal
4	Alternative & Punk
5	Rock And Roll
6	Blues
7	Latin
8	Gospel
9	Pop
10	Soundtrack

AlbumId	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	Raiders of the Wild	Rock
5	Process of the Dawn	Rock
6	Put The Finger On You	Rock
7	Let's Get It Up	Rock
8	Ignite The Nation	Rock
9	Snowballad	Rock
10	Evil Walks	Rock

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



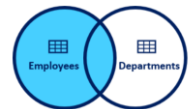
CLARUSWAY®

76

## LEFT JOIN

### 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

```
1 SELECT columns
2 FROM table_A
3 LEFT JOIN table_B ON join_conditions
```

join\_conditions

table\_A.common\_field = table\_B.common\_field

CLARUSWAY®

CLARUSWAY®

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

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60

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

exam	passing_score
SQL	70
AWS	80
Python	70
Network	60

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

CLARUSWAY®

CLARUSWAY®

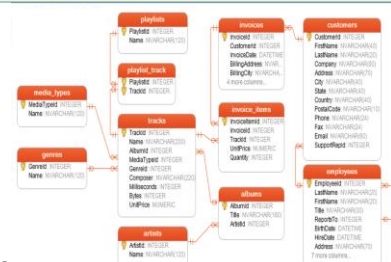
```
SELECT tests.exam, tests.passing_score,
       students.name, students.score
FROM tests
LEFT JOIN students ON tests.exam = students.exam;
```

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

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

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?**

