

SQL_ITF010_Session1_NOAH

session1-2

Training Clarusway

Pear Deck - December 11, 2021 at 11:33AM

Part 1 - Summary

Use this space to summarize your thoughts on the lesson

Part 2 - Responses

Slide 1

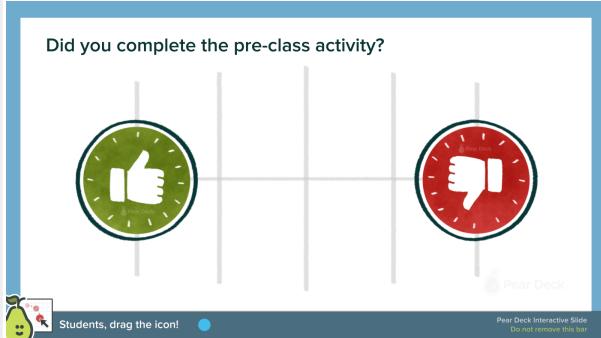


CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 2

Your Response



Use this space to take notes:

Slide 3

Table of Contents ➔

- ▶ What is a database?
- ▶ What is in a database?
- ▶ Structured Query Language (SQL)
- ▶ SQL Language Elements

CLARUSWAY®
WAY TO REINVENT YOURSELF

3

Use this space to take notes:

Slide 4



1

What is a database?

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 5

Your Response

Could you define what the database is?

Pear Deck Interactive Slide [Edit & Continue this slide](#)

Use this space to take notes:

Slide 6

► What is a database?



"A database is an organized collection of data stored in a computer system."

CLARUSWAY®
“a way to innovative knowledge”



Use this space to take notes:

Slide 7

► How are databases used in the real-world?



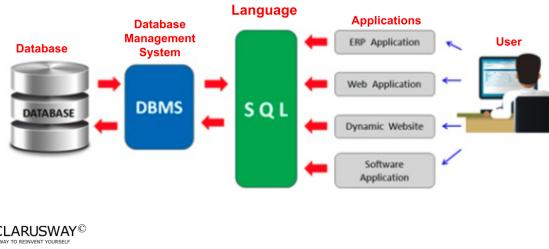
CLARUSWAY®
“a way to innovative knowledge”



Use this space to take notes:

Slide 8

► Database Management System



8

Use this space to take notes:

Slide 9

► Database Management System

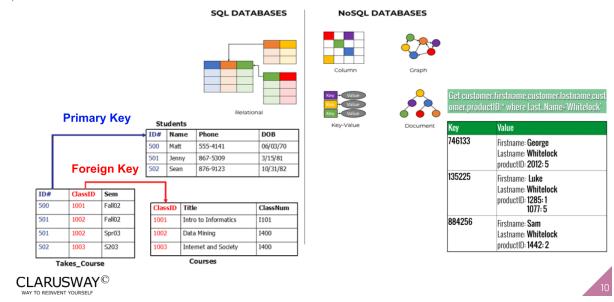


9

Use this space to take notes:

Slide 10

► Types of Database Management System



Use this space to take notes:

Slide 11

► Types of Database Management System



CLARUSWAY®
your no-brainer resource

11

Use this space to take notes:

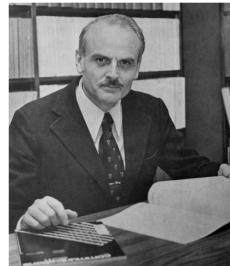
Slide 12

► Types of Database Management System

Edgar Frank "Ted" Codd
(19 August 1923 – 18 April 2003)

English computer scientist

While working for IBM, invented the relational model for database management.



12

Use this space to take notes:

Slide 13



"SQLite is a C-language library that implements a small, fast, self-contained, serverless, high-reliability, full-featured, SQL database engine. SQLite is the **most used database engine** in the world. SQLite is built into **all mobile phones** and **most computers** and comes bundled inside countless other applications that people use every day."

CLARUSWAY®
your IT environment simplified

13

Use this space to take notes:

Slide 14

Your Response

What is in a database?



Students, write your response!

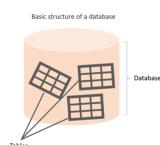


Pear Deck Interactive Slide
Do not remove this bar

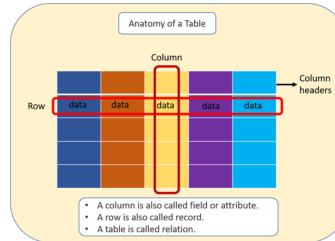
Use this space to take notes:

Slide 15

▶ What is in a database?



CLARUSWAY®
Learn to Review, Reflect

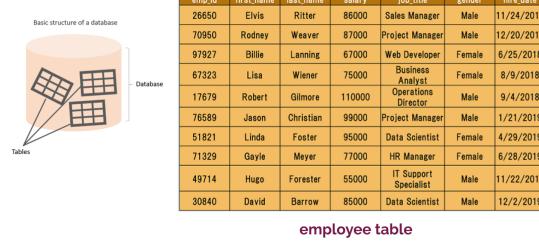


15

Use this space to take notes:

Slide 16

► What is in a database?



CLARUSWAY®
way to reinvent yourself

16

Use this space to take notes:

Slide 17



3 ► Structured Query Language (SQL)

CLARUSWAY®
way to reinvent yourself

Use this space to take notes:

Slide 18

► SQL



SQL stands for **Structured Query Language**

SQL lets you access and manipulate relational databases

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

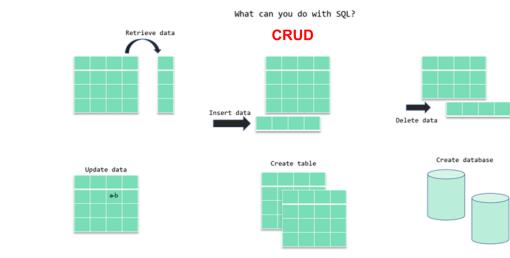
CLARUSWAY®
your no-regrets resource

18

Use this space to take notes:

Slide 19

► Structured Query Language (SQL)



CLARUSWAY®
your no-regrets resource

19

Use this space to take notes:

Slide 20

Your Response

Drag your dot to how you are feeling:

Keep going, I understand I'm a little confused Stop, I need help!

Peer Deck Interactive Slide
Do not remove this bar

Students, drag the icon! ●

Use this space to take notes:

Slide 21



4

SQL Language Elements

CLARUSWAY®
BORN TO REINVENT YOURSELF

Use this space to take notes:

Slide 22



"The beginning of wisdom is the definition of terms."

Socrates (470 – 399 B.C.)

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 23

► SQL Language Rules



- 1) SQL is **not case-sensitive** language (Except our Strings).
- 2) SQL syntax looks like **English Grammar**.
`SELECT name, profit
FROM Companies
WHERE location == "USA"
ORDER BY number_of_employees;`
- 3) A **comma** ; is placed at the end of completed commands.
- 4) Generally, **BNF** notation is used
Keywords => UPPER CASE
identifiers => LOWER CASE
Example: `DROP TABLE students;`
- 5) **Non-numeric** expressions are enclosed in **single quotes**.
Examples: 'New York', 'John', '2021-02-01'

CLARUSWAY®
WAY TO REINVENT YOURSELF

23

Use this space to take notes:

Slide 24

► SQL Language Elements



SQL Language Elements

(SQL Syntax)

```
SELECT first_name FROM employees;
```

Color coding

Keyword Identifiers Terminating Semicolon

Statement

CLARUSWAY®
WAY TO REINVENT YOURSELF



Use this space to take notes:

Slide 25



► SELECT Statement

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 26

▶ Introduction

- You can retrieve rows from the columns of the table by using **SELECT** statement.
- **SELECT** statement is used with **FROM** keyword.
- The **SELECT** statement is used to select data from a database.

```
1  SELECT column_name(s) FROM table_name;  
2  ||
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

28

Use this space to take notes:

Slide 27

```
SELECT first_name FROM employees;  
employees table
```

emp_id	first_name	last_name	salary	job_title	gender	hire_date
29650	Elvis	Ritter	86000	Sales Manager	Male	11/24/2017
70950	Rodney	Weaver	87000	Project Manager	Male	12/20/2018
97927	Bille	Lanning	67000	Web Developer	Female	6/25/2018
67323	Lisa	Wiener	75000	Business Analyst	Female	8/9/2018
17679	Robert	Gilmore	110000	Operations Director	Male	9/4/2018
76589	Jason	Christian	99000	Project Manager	Male	1/21/2019
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2019
71329	Gayle	Mayer	77000	HR Manager	Female	6/28/2019
49714	Hugo	Forester	55000	IT Support Specialist	Male	11/22/2018
30840	David	Barrow	85000	Data Scientist	Male	12/2/2019

first_name
Elvis
Rodney
Bille
Lisa
Robert
Jason
Linda
Gayle
Hugo
David

CLARUSWAY®
WAY TO REINVENT YOURSELF

▶

Use this space to take notes:

Slide 28

► Basic Syntax



```
1 | SELECT first_name FROM employees;
```

```
1 | select column_name(s) from table_name;
2 | SELECT COLUMN_NAME(s) FROM TABLE_NAME;
3 |
```

```
1 |          column_name(s)
2 |          table_name;
3 |
```

CLARUSWAY®
way to reinvent yourself

28

Use this space to take notes:

Slide 29



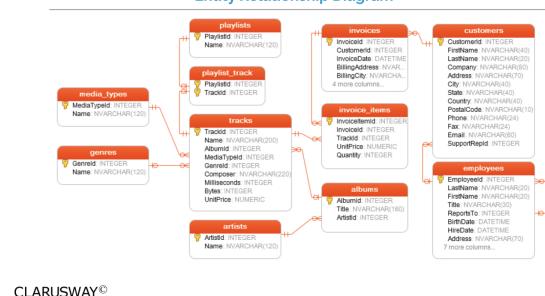
► Query Time > _

CLARUSWAY®
way to reinvent yourself

Use this space to take notes:

Slide 30

Entity Relationship Diagram



CLARUSWAY®
WAY TO REINVENT YOURSELF



30

Use this space to take notes:

Slide 31

Write a query that returns the track name using tracks table.



CLARUSWAY®
WAY TO REINVENT YOURSELF



31

Use this space to take notes:

Slide 32



3 Selecting Multiple Columns

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 33

▶ Selecting Multiple Columns



column1	column2	column3
column1_value1	column2_value1	column3_value1
column1_value2	column2_value2	column3_value2
column1_value3	column2_value3	column3_value3

query :

```
1 SELECT column1, column2 FROM table;
```

output :

```
1 column1      column2
2 -----
3 column1_value1 column2_value1
4 column1_value2 column2_value2
5 column1_value3 column2_value3
```



CLARUSWAY®
way to reinvent yourself

33

Use this space to take notes:

Slide 34



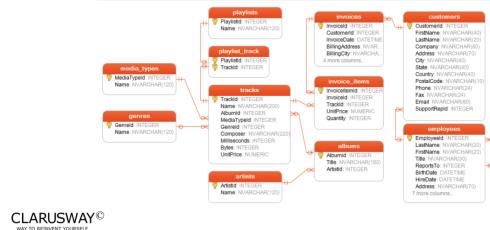
Query Time >_

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 35

Write a query that returns track name and its composer using tracks table.



CLARUSWAY®
WAY TO REINVENT YOURSELF



Use this space to take notes:

Slide 36

▶ Selecting All Columns



column1	column2	column3
column1_value1	column2_value1	column3_value1
column1_value2	column2_value2	column3_value2
column1_value3	column2_value3	column3_value3

query :

```
1 SELECT column1, column2, column3 FROM table1;
```

output :

```
1 column1      column2      column3
2 -----      -----
3 column1_value1 column2_value1 column3_value1
4 column1_value2 column2_value2 column3_value2
5 column1_value3 column2_value3 column3_value3
6
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

36

Use this space to take notes:

Slide 37

Write a query that returns all columns of albums table.



CLARUSWAY®
WAY TO REINVENT YOURSELF

37

Use this space to take notes:

Slide 38

▶ Selecting All Columns (Special Character)

To retrieve all of the information from your table, an asterisk (*) character can be used after the SELECT command.

query :

```
1 SELECT * FROM table1;
```

output :

```
1 column1      column2      column3  
2 .....  
3 column1_value1 column2_value1 column3_value1  
4 column1_value2 column2_value2 column3_value2  
5 column1_value3 column2_value3 column3_value3  
6
```

CLARUSWAY®
WAY TO REINVENT YOURSELF



38

Use this space to take notes:

Slide 39

Write a query that returns columns of tracks table.



CLARUSWAY®
WAY TO REINVENT YOURSELF



39

Use this space to take notes:

Slide 40



DISTINCT Clause

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 41

▶ Introduction



Columns in the tables may often contain some duplicate values, but you may only need the distinct values as a result. In such cases, we use the **SELECT** statement with the **DISTINCT** clause.

CLARUSWAY®
WAY TO REINVENT YOURSELF



Use this space to take notes:

Slide 42

▶ Introduction

The **SELECT DISTINCT** is used to return only distinct (different/unique) values to eliminate duplicate rows in a result set. Here is the syntax of the **DISTINCT** clause:

```
1 SELECT DISTINCT column_name(s) FROM table_name;  
2
```

CLARUSWAY®
Your No.1 Resource Provider

42

Use this space to take notes:

Slide 43

▶ No Duplicated Rows

query :

```
1 SELECT DISTINCT student FROM student_table;  
2
```

output :

```
1 student  
2 -----  
3 Student1  
4 Student2  
5 Student3  
6 Student4  
7 Student5  
8 Student6  
9 Student7  
10
```

CLARUSWAY®
Your No.1 Resource Provider

43

Use this space to take notes:

Slide 44

▶ Duplicated Rows



query :

```
1 | SELECT DISTINCT lesson FROM student_table;
```

student_table		
student	lesson	grade
1 Student1	Mathematics	95
2 Student2	Literature	65
3 Student3	Mathematics	45
4 Student4	Chemistry	85
5 Student5	Physics	70
6 Student6	Physics	75
7 Student7	Mathematics	75

output:

```
1 lesson
2 -----
3 Mathematics
4 Literature
5 Chemistry
6 Physics
7
```

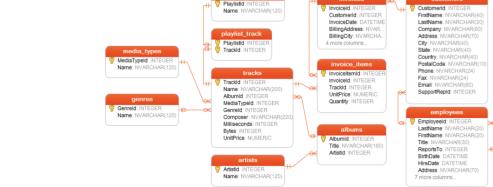
CLARUSWAY®
way to reinvent yourself

44

Use this space to take notes:

Slide 45

Find the name of composers of each track using tracks table.



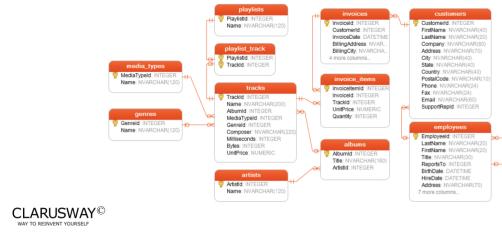
CLARUSWAY®
way to reinvent yourself

45

Use this space to take notes:

Slide 46

Write a query that return distinct
AlbumId, MediaTypeId pair.



Use this space to take notes:

Slide 47

► WHERE & LIMIT Clauses

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 48

▶ Introduction



The **WHERE** clause is used to filter records.
It allows you to define a specific search condition for
the result set returned by a query.

Syntax

```
1 SELECT column_name(s) FROM table_name WHERE condition(s);  
2
```



CLARUSWAY®
your to success

Use this space to take notes:

Slide 49

▶ WHERE Clause - Operators



Operators in the WHERE Clause

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. This operator may be written as != in some versions of SQL
BETWEEN	Test if a value is between a certain range of values
LIKE	Determine if a character string matches a predefined pattern
IN	Test whether or a value matches any value in a list



CLARUSWAY®
your to success

Use this space to take notes:

Slide 50

► WHERE Clause - Operators



student_table

student	lesson	grade
Student1	Mathematics	95
Student2	Literature	65
Student3	Mathematics	45
Student4	Chemistry	85
Student5	Physics	70
Student6	Physics	70
Student7	Mathematics	75

query :

```
1 SELECT * FROM student_table WHERE grade > 70
2
```

output:

student	lesson	grade
1
2
3
4	Student4	Chemistry 85
5	Student5	Physics 70
6	Student7	Mathematics 75
7		

CLARUSWAY®
your IT education provider

50

Use this space to take notes:

Slide 51

► Example-1



student_table

student	lesson	grade
1 Student1	Mathematics	95
2 Student2	Literature	65
3 Student3	Mathematics	45
4 Student4	Chemistry	85
5 Student5	Physics	70
6 Student6	Physics	75
7 Student7	Mathematics	75

query :

```
1 SELECT * FROM student_table WHERE lesson = "Mathematics";
2
```

output :

student	lesson	grade
1
2
3
4	Student1	Mathematics 95
5	Student3	Mathematics 45
6	Student7	Mathematics 75
7		

CLARUSWAY®
your IT education provider

51

Use this space to take notes:

Slide 52

► Example-2



query :

```
1 SELECT * FROM student_table WHERE grade < 70
```

student_table		
student	lesson	grade
1 Student1	Mathematics	95
2 Student2	Literature	65
3 Student3	Mathematics	45
4 Student4	Chemistry	85
5 Student5	Physics	70
6 Student6	Physics	75
7 Student7	Mathematics	75

output:

```
1 student      lesson      grade
2 .....      .....      .....
3 Student2    Literature  65
4 Student3    Mathematic 45
5 |
```

CLARUSWAY®
way to reinvent yourself

52

Use this space to take notes:

Slide 53

Find the track names of Jimi Hendrix.



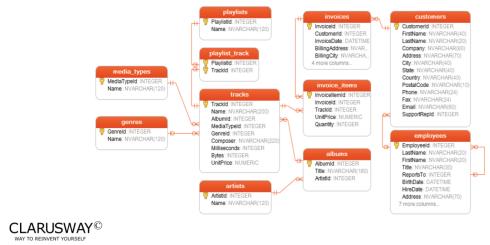
CLARUSWAY®
way to reinvent yourself

53

Use this space to take notes:

Slide 54

Find all the info of the invoices of which total amount is greater than \$10.



5

Use this space to take notes:

Slide 55



5 ➤ LIMIT Clause

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 56

► LIMIT Clause



- The **LIMIT** clause is used to filter records.
- It constrains the number of rows returned by a query.

Here is the syntax of the **LIMIT** clause.

```
1 SELECT column_name(s) FROM table_name LIMIT number_rows;
```

58

Use this space to take notes:

Slide 57

► LIMIT Clause



query:

```
1 SELECT * FROM student_table LIMIT 3;
```

output:

student	lesson	grade
1 Student1	Mathematics	95
2 Student2	Literature	65
3 Student3	Mathematics	45
4 Student4	Chemistry	85
5 Student5	Physics	70
6 Student6	Physics	75
7 Student7	Mathematics	75

57

Use this space to take notes:

Slide 58

► LIMIT Clause



We can also combine **LIMIT** with **WHERE**. In that case, **LIMIT** clause is placed after the **WHERE** clause.

CLARUSWAY®
your IT education resource

58

Use this space to take notes:

Slide 59

► LIMIT Clause



query:

```
1 SELECT * FROM student_table WHERE grade > 70 LIMIT 2;
```

student_table		
student	lesson	grade
1 Student1	Mathematics	95
2 Student2	Literature	65
3 Student3	Mathematics	45
4 Student4	Chemistry	85
5 Student5	Physics	70
6 Student6	Physics	75
7 Student7	Mathematics	75

output:

```
1 student      lesson      grade
2 -----
3 Student1    Mathematics  95
4 Student4    Chemistry   85
5
```

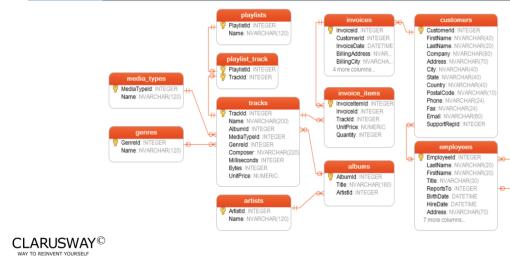
CLARUSWAY®
your IT education resource

59

Use this space to take notes:

Slide 60

Find all the info of the invoices of which total amount is greater than \$10. Just return the first 4 rows.



CLARUSWAY®
WAY TO REINVENT YOURSELF

69

Use this space to take notes:

Slide 61

ORDER BY Clause

A
Z
↓
Z
A
↓

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 62

► Order By Clause

- In case you want to retrieve data in alphabetical or numeric order, we use **ORDER BY** keyword.
- By default **ORDER BY** keyword sorts the records in ascending order.
- Use the keyword **DESC** to sort the records in descending order. You can also use **ASC** explicitly to sort the data in ascending order.

Syntax

```
1 SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC;
```

CLARUSWAY®
data to empower business

62

Use this space to take notes:

Slide 63

► Order By Clause

emp_id	first_name	last_name	salary	job_title	gender	hire_date
28660	Eve	Weller	60000	Sales Manager	Male	11/24/2011
20950	Rodney	Weaver	87000	Project Manager	Male	12/20/2018
97977	Billy	Lamont	67000	Web Developer	Female	12/25/2018
67221	Lisa	Werner	75000	Business Development	Female	8/8/2018
117879	Robert	Gilmour	110000	Operations Manager	Male	9/4/2018
76589	Jesse	Christian	59000	Project Manager	Male	1/21/2019
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2019
11329	Gayle	Weyer	77000	HR Manager	Female	8/28/2019
49714	Hugo	Forster	150000	IT Support Lead	Male	1/22/2019
20840	David	Barlow	85000	Data Scientist	Male	12/2/2019

query :

```
1 SELECT * FROM employees ORDER BY first_name ASC;
```

output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
28660	Eve	Weller	60000	Sales Manager	Male	11/24/2011
20950	Rodney	Weaver	87000	Project Manager	Male	12/20/2018
97977	Billy	Lamont	67000	Web Developer	Female	12/25/2018
67221	Lisa	Werner	75000	Business Development	Female	8/8/2018
117879	Robert	Gilmour	110000	Operations Manager	Male	9/4/2018
76589	Jesse	Christian	59000	Project Manager	Male	1/21/2019
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2019
11329	Gayle	Weyer	77000	HR Manager	Female	8/28/2019
49714	Hugo	Forster	150000	IT Support Lead	Male	1/22/2019
20840	David	Barlow	85000	Data Scientist	Male	12/2/2019

CLARUSWAY®
data to empower business

63

Use this space to take notes:

Slide 64

► Sorting in Descending Order



query :

```
1. SELECT * FROM employees ORDER BY first_name DESC;
```

emp_id	first_name	last_name	salary	job_title	gender	hire_date
70000	Eva	Wenner	67000	Project Manager	Male	11/05/2011
70001	Robert	Wagner	67000	Project Manager	Male	11/20/2018
87927	Bella	Lamont	67000	Web Developer	Female	4/25/2018
67232	Lisa	Weller	70000	Business Analyst	Female	8/6/2018
17679	Robert	Gilmores	110000	Operations Director	Male	8/4/2018
76589	Jesse	Christien	99000	Project Manager	Male	1/21/2019
51871	Linda	Foster	99000	Data Scientist	Female	4/25/2019
71329	Gayle	Meyer	77000	HR Manager	Female	6/28/2019
49714	Hugo	Forster	55000	IT Support Spec	Male	11/22/2019
30840	David	Barrow	85000	Data Scientist	Male	12/2/2019

output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
70000	Eva	Wenner	67000	Project Manager	Male	11/05/2011
70001	Robert	Wagner	67000	Project Manager	Male	11/20/2018
87927	Bella	Lamont	67000	Web Developer	Female	4/25/2018
67232	Lisa	Weller	70000	Business Analyst	Female	8/6/2018
17679	Robert	Gilmores	110000	Operations Director	Male	8/4/2018
76589	Jesse	Christien	99000	Project Manager	Male	1/21/2019
51871	Linda	Foster	99000	Data Scientist	Female	4/25/2019
71329	Gayle	Meyer	77000	HR Manager	Female	6/28/2019
49714	Hugo	Forster	55000	IT Support Spec	Male	11/22/2019
30840	David	Barrow	85000	Data Scientist	Male	12/2/2019
30840	David	Barrow	85000	Data Scientist	Male	2019-12-02
30840	David	Barrow	85000	Data Scientist	Male	2019-06-25
30840	David	Barrow	85000	Data Scientist	Male	2018-06-25

CLARUSWAY®
data to knowledge™

64

Use this space to take notes:

Slide 65

► Sorting in Descending Order



query :

```
1. SELECT first_name, last_name, salary FROM employees ORDER BY salary DESC;
```

emp_id	first_name	last_name	salary
70000	Eva	Wenner	67000
70001	Robert	Wagner	67000
87927	Bella	Lamont	67000
67232	Lisa	Weller	70000
17679	Robert	Gilmores	110000
76589	Jesse	Christien	99000
51871	Linda	Foster	99000
71329	Gayle	Meyer	77000
49714	Hugo	Forster	55000
30840	David	Barrow	85000

output:

1	first_name	last_name	salary
2	Robert	Gilmores	110000
3	Robert	Christien	99000
4	Linda	Foster	99000
5	Bella	Lamont	99000
6	Lisa	Weller	77000
7	Elisa	Ritter	60000
8	David	Barrow	60000
9	Gayle	Meyer	60000
10	Jesse	Wenner	55000
11	Elisie	Lamont	55000
12	Hugo	Forster	55000

CLARUSWAY®
data to knowledge™

65

Use this space to take notes:

Slide 66

► Sorting By Multiple Columns ➤

```
1 SELECT column_name(s) FROM table_name ORDER BY column1 ASC|DESC, column2  
      ASC|DESC, columnN ASC|DESC;  
2
```

CLARUSWAY®
data to drive your business

66

Use this space to take notes:

Slide 67

► Sorting By Multiple Columns ➤

query:

```
1 SELECT * FROM employees ORDER BY gender DESC, first_name ASC;
```

emp_id	first_name	last_name	email	job_title	gender	hire_date
28650	Eve	Holt	eholt@reqres.in	Sales Manager	Male	2012-08-17
70950	Rodney	Werner	rwerner@reqres.in	Project Manager	Male	2012-09-29
87927	Billie	Lamming	blamming@reqres.in	Web Developer	Female	2013-01-23
87323	Lisa	Wiener	lwiener@reqres.in	Business Development	Female	2013-01-23
17679	Robert	Gilmore	rgilmore@reqres.in	Operations Manager	Male	2014-01-23
76889	Jean	Christien	jchristien@reqres.in	Project Manager	Male	2015-01-23
51821	Linda	Foster	lfoster@reqres.in	Data Scientist	Female	2015-01-23
73129	Geyle	Meyer	gmeyer@reqres.in	HR Manager	Female	2016-01-23
49714	Hugo	Forester	hforester@reqres.in	IT Support	Male	2016-01-23
30840	David	Barrow	dbarrow@reqres.in	Data Scientist	Male	2017-01-23

output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
30840	David	Barrow	85000	Data Scientist	Male	2017-12-02
30840	David	Barrow	85000	Data Scientist	Male	2017-12-02
49714	Hugo	Forester	55000	IT Support Spw	Male	2016-01-22
49714	Hugo	Forester	55000	IT Support Spw	Male	2016-01-22
17679	Robert	Gilmore	110000	Operations Dir	Male	2013-01-23
17679	Robert	Gilmore	110000	Operations Dir	Male	2013-01-23
79727	Billie	Lamming	67000	Web Developer	Female	2016-01-23
79727	Billie	Lamming	67000	Web Developer	Female	2016-01-23
51821	Linda	Foster	95000	Data Scientist	Female	2015-01-23
51821	Linda	Foster	95000	Data Scientist	Female	2015-01-23
67723	Lisa	Wiener	75000	Business Analy	Female	2018-04-09

CLARUSWAY®
data to drive your business

67

Use this space to take notes:

Slide 68

► ORDER BY Clause with WHERE Clause

```
1 SELECT column_name(s) FROM table_name WHERE condition ORDER BY column_name(s)s
2 ASC|DESC;
```

↓ Beautifying

```
1 SELECT column_name(s)
2 FROM table_name
3 WHERE condition
4 ORDER BY column_name(s)s ASC|DESC;
5
```

CLARUSWAY®
data to knowledge connector

68

Use this space to take notes:

Slide 69

► ORDER BY Clause with WHERE Clause

query :

```
1 SELECT *
2 FROM employees
3 WHERE salary > 80000
4 ORDER BY first_name DESC;
```

Output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
28650	Elvis	Ritter	86000	Sales Manager	Male	11/24/2011
70950	Rodney	Weaver	871000	Project Manager	Male	12/20/2014
37927	Willie	Lanning	88000	Web Developer	Female	9/20/2013
37325	Allen	Wong	100000	Analyst	Male	9/5/2014
27619	Robert	Gleeson	110000	Software Dev.	Male	9/4/2013
76359	Jean	Christie	90000	Project Manager	Male	1/21/2013
51821	Linda	Foster	95000	Data Scientist	Female	4/28/2013
71322	David	Wever	771000	HR Manager	Female	4/28/2013
48714	Hazel	Forster	55000	IT Support	Male	11/22/2016
30843	David	Barrow	85000	Data Scientist	Male	12/2/2013

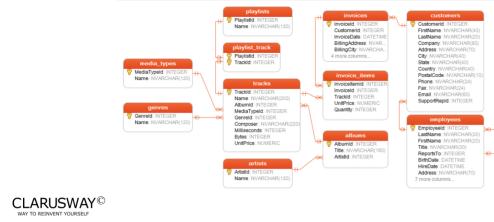
CLARUSWAY®
data to knowledge connector

69

Use this space to take notes:

Slide 70

Find all the info of the invoices of which total amount is greater than \$10. Then sort them by the total amount in descending order.



CLARUSWAY®
WAY TO REINVENT YOURSELF

9

Use this space to take notes:

Slide 71

»

AND, OR & NOT Operators

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 72



1 ▶ Introduction

CLARUSWAY®
WAY TO REINVENT YOURSELF

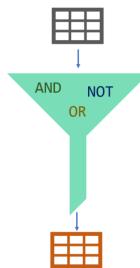
Use this space to take notes:

Slide 73

▶ Introduction



In SQL, **AND**, **OR** & **NOT** keywords are called logical operators. Their purposes are filtering the data based on conditions.



CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 74

► AND Operator



The **AND** operator is used with the **WHERE** clause and combines multiple expressions. It returns only those records where both conditions (in **WHERE** clause) evaluate to **True**.

Syntax

```
1 WHERE left_condition AND right_condition  
2 |
```

CLARUSWAY®
data to empower business

74

Use this space to take notes:

Slide 75

► AND Operator



query :

```
1 SELECT *  
2 FROM employees  
3 WHERE job_title = 'Data Scientist' AND gender = 'Male';  
4 |
```

emp_id	first_name	last_name	salary	job_title	gender	hire_date
26600	Elvis	Birner	66000	Sales Manager	Male	1/12/2013
70900	Rodney	Werner	87000	Project Manager	Male	12/20/2018
87927	Stile	Lanning	87000	Web Developer	Female	6/25/2018
47223	Lisa	Werner	76000	Business Development	Female	8/7/2018
17679	Robert	Gilesius	110000	Operations Manager	Male	9/4/2018
76599	Jean	Christian	90000	Project Manager	Male	1/7/2019
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2019
71329	Gayle	Meyer	77000	HR Manager	Female	6/28/2019
49714	Hugo	Forrester	85000	IT Support	Male	11/22/2019
30840	David	Barrow	85000	Data Scientist	Male	12/2/2019

output:

```
emp_id ..... first_name ..... last_name ..... salary ..... job_title ..... gender ..... hire_date .....  
10840 ..... David ..... Barrow ..... 85000 ..... Data Scientist ..... Male ..... 2019-12-02
```

75

Use this space to take notes:

Slide 76

► OR Operator

The **OR** operator is used with the **WHERE** clause and combines multiple expressions. It displays the record where either one of conditions (in WHERE clause) evaluates to **True**.

Syntax

```
1 WHERE first_condition OR second_condition  
2 |
```

CLARUSWAY®
data to drive your business

78

Use this space to take notes:

Slide 77

► OR Operator

emp_id	first_name	last_name	salary	job_title	gender	hire_date
28660	Eve	Weller	60000	Sales Manager	Male	11/24/2011
20950	Rodney	Weaver	87000	Project Manager	Male	7/20/2018
97977	Billy	Lamont	67000	Web Developer	Female	4/25/2018
67221	Lisa	Wiser	75000	Business Development	Female	8/8/2018
117879	Robert	Glover	110000	Operations Manager	Male	8/4/2018
76689	Jesse	Christian	59000	Project Manager	Male	1/21/2019
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2019
11329	Gayle	Weyer	77000	HR Manager	Female	8/28/2019
49714	Hugo	Forrester	150000	IT Support Specialist	Male	1/22/2019
20840	David	Barrow	85000	Data Scientist	Male	12/2/2019

```
query :  
1 SELECT *  
2 FROM employees  
3 WHERE job_title = 'Data Scientist' OR gender = 'Male';  
4
```

output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
28660	Eve	Weller	60000	Sales Manager	Male	11/24/2011
20950	Rodney	Weaver	87000	Project Manager	Male	7/20/2018
97977	Billy	Lamont	67000	Web Developer	Female	4/25/2018
67221	Lisa	Wiser	75000	Business Development	Female	8/8/2018
117879	Robert	Glover	110000	Operations Manager	Male	8/4/2018
76689	Jesse	Christian	59000	Project Manager	Male	1/21/2019
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2019
11329	Gayle	Weyer	77000	HR Manager	Female	8/28/2019
49714	Hugo	Forrester	150000	IT Support Specialist	Male	1/22/2019
20840	David	Barrow	85000	Data Scientist	Male	12/2/2019

CLARUSWAY®
data to drive your business

77

Use this space to take notes:

Slide 78

▶ NOT Operator



The **NOT** operator is used to negate a condition in the **WHERE** clause. **NOT** is placed right after **WHERE** keyword. You can use it with AND & OR operators.

Syntax

```
1 WHERE NOT first_condition  
2
```

CLARUSWAY®
data to empower business

78

Use this space to take notes:

Slide 79

▶ NOT Operator



query :

```
1 SELECT *  
2 FROM Employees  
3 WHERE NOT gender = 'Female';
```

emp_id	first_name	last_name	salary	job_title	gender	hire_date
26451	Elvis	Kitter	86000	Sales Manager	Male	11/08/2013
70560	Ridley	Weaver	87000	Project Manager	Male	12/20/2018
87927	Belle	Lanning	87000	Web Developer	Female	6/26/2018
87223	Lisa	Werner	76500	Business Development	Female	8/3/2018
17679	Robert	Gilles	110000	Operations Manager	Male	9/4/2018
76569	Jean	Christien	96000	Project Manager	Male	1/21/2019
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2019
71209	Gayle	Meyer	77000	HR Manager	Female	6/28/2019
49714	Hugo	Forster	85000	IT Support	Male	11/28/2019
30840	David	Barrow	85000	Data Scientist	Male	12/2/2019

output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
26451	Elvis	Kitter	86000	Sales Manager	Male	2017-11-24
70560	Ridley	Weaver	87000	Project Manager	Male	2018-06-12
87927	Belle	Lanning	87000	Web Developer	Female	2018-06-23
87223	Lisa	Werner	76500	Business Development	Female	2018-08-03
17679	Robert	Gilles	110000	Operations Manager	Male	2018-09-04
76569	Jean	Christien	96000	Project Manager	Male	2019-01-21

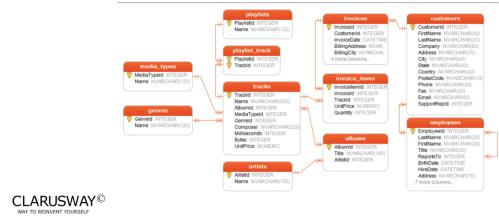
CLARUSWAY®
data to empower business

79

Use this space to take notes:

Slide 80

Find all the info of the invoices of which billing country is not USA. Then sort them by the total amount in ascending order.



CLARUSWAY®
WAY TO REINVENT YOURSELF



8
0

Use this space to take notes:

Slide 81



CLARUSWAY®
WAY TO REINVENT YOURSELF

BETWEEN OPERATOR

Use this space to take notes:

Slide 82

▶ Introduction

The **BETWEEN** operator is used for comparison in **WHERE** clauses. It's a comparison operator. You can use it to test if a value is in a range of values. If the value is in the specified range, the query returns all records fallen within that range.

```
1 WHERE test_expression BETWEEN low_expression AND high_expression  
2 |
```



```
1 WHERE test_expression >= low_expression AND test_expression <= high_expression  
2 |
```

CLARUSWAY®
data to drive your business

82

Use this space to take notes:

Slide 83

▶ Introduction



query :

```
1 SELECT *  
2 FROM employees  
3 WHERE salary BETWEEN 80000 AND 90000;  
4 |
```

emp_id	first_name	last_name	salary	job_title	gender	hire_date
26650	Elvis	Ritter	86000	Sales Manager	Male	11/24/2017
70950	Rodney	Weaver	87000	Project Manager	Male	12/10/2018
31720	Elaine	Ernest	75000	Analyst	Female	4/5/2018
41723	Lita	Wiser	76000	Software Analyst	Female	8/14/2018
17679	Robert	Glover	100000	Operations Director	Male	9/14/2018
74499	Jean	Christie	99000	Product Manager	Male	1/17/2018
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2018
71329	Gavile	Meyer	77000	HR Manager	Female	8/28/2018
49714	Hugo	Forster	55000	IT Support Specialist	Male	11/22/2018
20840	David	Burke	85000	Data Scientist	Male	12/2/2018

output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
26650	Elvis	Ritter	86000	Sales Manager	Male	2017-11-24
70950	Rodney	Weaver	87000	Project Manager	Male	2018-12-28

CLARUSWAY®
data to drive your business

83

Use this space to take notes:

Slide 84

▶ NOT BETWEEN Operator



We can use **NOT BETWEEN** to negate the result of the **BETWEEN** operator. The following is the syntax:

Syntax

```
1 WHERE test_expression NOT BETWEEN low_expression AND high_expression  
2
```

CLARUSWAY®
data to drive your business

84

Use this space to take notes:

Slide 85

▶ BETWEEN with Date Example



query :

```
1 SELECT *  
2 FROM employees  
3 WHERE hire_date BETWEEN '2013-06-01' AND '2015-03-31'  
4 ORDER BY hire_date;
```

output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
28650	Elvis	Ritter	86000	Sales Manager	Male	11/24/2017
70950	Redey	Waeer	87000	Project Manager	Male	12/20/2018
87927	Stile	Lamting	87000	Web Developer	Female	6/20/2018
87323	Lisa	Waine	76000	Analyst	Female	8/8/2018
17619	Robert	Gilmore	110000	Director	Male	9/4/2018
76549	Jean	Christian	99000	Project Manager	Male	1/21/2019
51811	Linda	Foster	95000	Data Scientist	Female	4/29/2019
71229	Geyle	Weyer	77000	Analyst	Female	6/29/2019
49714	Hugo	Forster	55000	IT Support Specialist	Male	1/12/2020
30840	David	Burke	85000	Data Scientist	Male	12/2/2019

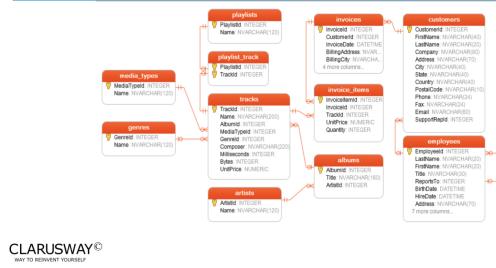
CLARUSWAY®
data to drive your business

85

Use this space to take notes:

Slide 86

Find the newest invoice date among the invoice dates between 2009 and 2011.



CLARUSWAY®
WAY TO REINVENT YOURSELF

87

Use this space to take notes:

Slide 87



Using **BETWEEN** is tricky for datetime! While **BETWEEN** is generally inclusive of endpoints, it assumes the time is at 00:00:00 (i.e. midnight) for **datetime**. So, the end point is exclusive. But, if you have just **date**, then **BETWEEN** behaves as expected.

CLARUSWAY®
WAY TO REINVENT YOURSELF

87

Use this space to take notes:

Slide 88



► IN OPERATOR

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 89



1 ► Introduction

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 90

▶ Introduction



The **IN** operator is used to determine whether a value matches any value in a list. We use **IN** operator with **WHERE** clause.

Syntax

```
1 WHERE column_name IN (value_list)  
2 |
```

CLARUSWAY®
data to drive your business

90

Use this space to take notes:

Slide 91

▶ Introduction



query:

```
1 SELECT *  
2 FROM employees  
3 WHERE job_title IN ('Data Scientist', 'Business Analyst');  
4 |
```

Output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
29650	Elvis	Ritter	80000	Sales Manager	Male	1/24/2011
30950	John	Smith	70000	Software Engineer	Male	1/24/2011
47917	Darryl	Loring	93000	Web Developer	Female	1/25/2018
47323	Lisa	Winer	76000	Business Analyst	Female	8/6/2018
17678	Robert	Gibson	110000	Operations Manager	Male	8/4/2018
26569	Jean	Orsini	90000	Project Manager	Male	1/21/2018
51821	Linda	Foster	90000	Data Scientist	Female	4/28/2018
71329	David	Meyer	77000	HR Manager	Female	8/28/2018
49714	Hugh	Forster	95000	IT Support Specialist	Male	11/22/2018
20940	David	Barrow	95000	Data Architect	Male	12/2/2018

CLARUSWAY®
data to drive your business

91

Use this space to take notes:

Slide 92



If you have a query in which you use many **OR** operators, consider using the **IN** operator instead. This will make your query more readable.

CLARUSWAY®
WAY TO REINVENT YOURSELF

93
92

Use this space to take notes:

Slide 93

▶ NOT IN Operator

94

We are going to add the keyword **NOT** to our **IN** operator.

query :

```
1 SELECT *  
2 FROM employee  
3 WHERE job_title  
4 NOT IN ('Operations Director', 'HR Manager', 'Sales Manager');  
5 |||
```

output:

emp_id	first_name	last_name	salary	job_title	gender	hire_date
29890	Eduo	Ritter	86000	Sales Manager	Male	11/24/2011
79950	Rodney	Weaver	87000	Project Manager	Male	12/20/2011
87927	Will	Lamont	88000	Software Engineer	Female	10/20/2011
87870	Bill	Wong	76000	Business Analyst	Male	9/9/2010
11939	Robert	Elmire	110000	Department Manager	Male	9/4/2010
76589	Jean	Orsi	99000	Project Manager	Male	1/21/2010
51821	Linda	Fester	95000	Data Scientist	Female	4/25/2010
71329	Gavyle	Weyer	77000	HR Manager	Female	4/26/2010
49714	Hugo	Forrester	95000	IT Support Specialist	Male	11/22/2010
30940	David	Barrow	85000	Data Scientist	Male	12/2/2010

CLARUSWAY®
way to reinvent yourself

95

Use this space to take notes:

Slide 94

Find the first and last name of the customers who gave an order from Belgium, Norway, Canada and USA.



CLARUSWAY®
WAY TO REINVENT YOURSELF

3
4

Use this space to take notes:

Slide 95



LIKE OPERATOR

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 96

▶ Introduction



Syntax

```
1 SELECT column_name(s)
2 FROM table_name
3 WHERE column_1 LIKE pattern;
4 |
```

CLARUSWAY®
Your Path to Professional Success!

96

Use this space to take notes:

Slide 97

▶ Introduction



After **LIKE** keyword, we construct a pattern. SQL provides two special characters for constructing patterns. These are also called wildcards.

- Percent (%): The % character matches any sequence of zero or more characters.
- Underscore (_): The _ character matches any single character

CLARUSWAY®
Your Path to Professional Success!

97

Use this space to take notes:

Slide 98

▶ Introduction

query :

```
1 SELECT *  
2 FROM student_info  
3 WHERE county LIKE 'Wo%';  
4
```

output:

```
1 SELECT *  
2 FROM student_info  
3 WHERE county LIKE 'No%'  
4
```

	student_id	first_name	last_name	gender	state	county	field	start_date
1	170555	Megan	Walker	F	West Virginia	Wood	Front-End Developer	2019-06-21

9

Use this space to take notes:

Slide 99

► Percent Character Example

2

query :

```
1 SELECT *  
2 FROM student_info  
3 WHERE field LIKE '%Developer';
```

student_info table						
student_id	stu_name	stu_class	stu_gpa	stu_email	stu_major	stu_awards
S10001	John	Computer	3.8	John@gmail.com	Computer	None
S10002	Olivia	Math	3.9	Olivia.Taylor@gmail.com	Math	None
S10003	David	Chemical	3.7	David.Taylor@gmail.com	Chemical	None
S10004	Grace	Physics	3.6	Grace.Taylor@gmail.com	Physics	None
S10005	Henry	English	3.5	Henry.Taylor@gmail.com	English	None
S10006	King	Math	3.4	King.Taylor@gmail.com	Math	None
S10007	Richard	Physics	3.3	Richard.Taylor@gmail.com	Physics	None
S10008	Grace	Chemical	3.2	Grace.Taylor@gmail.com	Chemical	None
S10009	Olivia	Computer	3.1	Olivia.Taylor@gmail.com	Computer	None
S10010	David	Math	3.0	David.Taylor@gmail.com	Math	None
S10011	Grace	Physics	2.9	Grace.Taylor@gmail.com	Physics	None
S10012	Henry	English	2.8	Henry.Taylor@gmail.com	English	None
S10013	King	Math	2.7	King.Taylor@gmail.com	Math	None
S10014	Richard	Physics	2.6	Richard.Taylor@gmail.com	Physics	None
S10015	Grace	Chemical	2.5	Grace.Taylor@gmail.com	Chemical	None
S10016	Olivia	Computer	2.4	Olivia.Taylor@gmail.com	Computer	None
S10017	David	Math	2.3	David.Taylor@gmail.com	Math	None
S10018	Grace	Physics	2.2	Grace.Taylor@gmail.com	Physics	None
S10019	Henry	English	2.1	Henry.Taylor@gmail.com	English	None
S10020	King	Math	2.0	King.Taylor@gmail.com	Math	None
S10021	Richard	Physics	1.9	Richard.Taylor@gmail.com	Physics	None
S10022	Grace	Chemical	1.8	Grace.Taylor@gmail.com	Chemical	None
S10023	Olivia	Computer	1.7	Olivia.Taylor@gmail.com	Computer	None
S10024	David	Math	1.6	David.Taylor@gmail.com	Math	None
S10025	Grace	Physics	1.5	Grace.Taylor@gmail.com	Physics	None
S10026	Henry	English	1.4	Henry.Taylor@gmail.com	English	None
S10027	King	Math	1.3	King.Taylor@gmail.com	Math	None
S10028	Richard	Physics	1.2	Richard.Taylor@gmail.com	Physics	None
S10029	Grace	Chemical	1.1	Grace.Taylor@gmail.com	Chemical	None
S10030	Olivia	Computer	1.0	Olivia.Taylor@gmail.com	Computer	None
S10031	David	Math	0.9	David.Taylor@gmail.com	Math	None
S10032	Grace	Physics	0.8	Grace.Taylor@gmail.com	Physics	None
S10033	Henry	English	0.7	Henry.Taylor@gmail.com	English	None
S10034	King	Math	0.6	King.Taylor@gmail.com	Math	None
S10035	Richard	Physics	0.5	Richard.Taylor@gmail.com	Physics	None
S10036	Grace	Chemical	0.4	Grace.Taylor@gmail.com	Chemical	None
S10037	Olivia	Computer	0.3	Olivia.Taylor@gmail.com	Computer	None
S10038	David	Math	0.2	David.Taylor@gmail.com	Math	None
S10039	Grace	Physics	0.1	Grace.Taylor@gmail.com	Physics	None
S10040	Henry	English	0.0	Henry.Taylor@gmail.com	English	None

output

3

Use this space to take notes:

Slide 100

▶ Underscore Character Example



query :

```
1 SELECT first_name
2 FROM employees
3 WHERE first_name LIKE '%_l_is';
```

output :

```
1 first_name
2 -----
3 Elvis
```



CLARUSWAY®
way to reinvent yourself

Use this space to take notes:

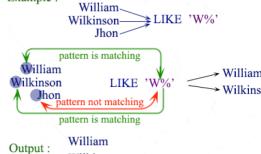
Slide 101



SQL LIKE Operator

Syntax : LIKE pattern

Example :



Example :

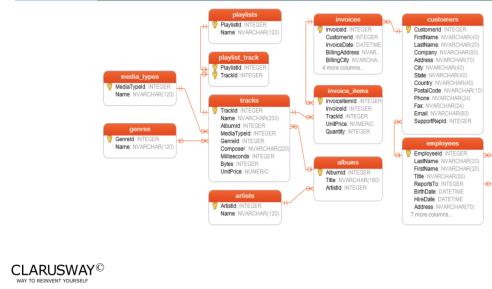


© w3resource.com

Use this space to take notes:

Slide 102

Find the track names of Bach.



CLARUSWAY®
WAY TO REINVENT YOURSELF

10
2

Use this space to take notes:

Slide 103

THANKS!
Any questions?



CLARUSWAY®
WAY TO REINVENT YOURSELF

10
3

Use this space to take notes: