Intermediate SQL

DigitalSkola
Uncover The World Of Digital Skills With Us

One thing we should know before we start ...

"Data is like garbage. You'd better know what you are going to do with it before you collect it."

- Mark Twain

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Table of Content

What will We Learn Today?

1. Distinct
2. String Functions
3. Aggregate Functions
4. Case When
5. Where Statement
6. Group By
7. Having
8. Order By
9. Limit

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Let’s get your hand dirty!

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Open DBeaver in your PC

- Click add new connection in the top left
- Pick PostgreSQL
- Fill the credentials as below:
  - Host: digitalskoladb.c04me33o8tni.ap-southeast-1.rds.amazonaws.com
  - Port: 5432
  - Database: sandbox
  - Username: ds11_(nomor kelompok)
  - Password: ds11_(nomor kelompok)

© Copyright by Digital Skola 2020

General Function

SQL has a mandatory format to follow. Wrong format and/or order would causing error when we run it.

```
SELECT column1, column2,...
FROM table_name
WHERE condition (s)
GROUP BY field_name (s)
HAVING condition (s)
ORDER BY field_name (s)
LIMIT number
```

Tips: Not every lines needed to run the query

© Copyright by Digital Skola 2020

Select

Used to pick field or column to be shown. You could use (“*”) to pick all columns. Case sensitive applied in the column name.

```
SELECT * FROM table_name;

SELECT Column1, Column2,... FROM table_name;
```

Tips:

1. It’s better to pick columns one by one compared to use *
2. We could change column name using aliases function (AS)

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Distinct

Remove duplicates in one column. Effective to shows one column without group by or to count unique values.

```
SELECT DISTINCT column1
FROM table_name
```

Tips:

1. Use it accordingly as it removes duplicate rows

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

String Function

Used to manipulate string(s)

Function	Return Type	Description	Example	Result
string || string	text	String concatenation	'Post' || 'greSQL'	PostgreSQL
string || non-string or non-string || string	text	String concatenation with one non-string input	'Value: ' || 42	Value: 42
char_length(string) or character_length(string)	int	Number of characters in string	char_length('jose')	4
lower(string)	text	Convert string to lowercase	lower('TOM')	tom
position(substring in string)	int	Location of specified substring	position('om' in 'Thomas')	3
substring(string [from int] [for int])	text	Extract substring	substring('Thomas' from 2 for 3)	hom
upper(string)	text	Convert string to uppercase	upper('tom')	TOM

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Aggregate Function

Used to summarize values

Function	Argument Type(s)	Return Type	Description
`avg(`***expression***`)`	smallint, int, bigint, real, double precision, numeric, or interval	numeric for any integer-type argument, double precision for a floating-point argument, otherwise the same as the argument data type	the average (arithmetic mean) of all non-null input values
`count(*)`	any	bigint	number of input rows
`count(`***expression***`)`	any	bigint	number of input rows for which the value of *expression* is not null
`max(`***expression***`)`	any numeric, string, date/time, network, or enum type, or arrays of these types	same as argument type	maximum value of *expression* across all non-null input values
`min(`***expression***`)`	any numeric, string, date/time, network, or enum type, or arrays of these types	same as argument type	minimum value of *expression* across all non-null input values
`sum(`***expression***`)`	smallint, int, bigint, real, double precision, numeric, interval, or money	bigint for smallint or int arguments, numeric for bigint arguments, otherwise the same as the argument data type	sum of *expression* across all non-null input values

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Case When Function

Goes through conditions and returns a value when the first condition is met. So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause.

If there is no ELSE part and no conditions are true, it returns NULL.

```
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END
```

Tips:

1. Fill conditions accordingly to make it effective
2. CASE WHEN function can be put inside SELECT, WHERE, and ORDER BY

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Where

Used to set limitation and/or condition to the query to filter the data based on needs.

Need to use AND / OR to set multiple limitations.

```
SELECT column1, column2,...
FROM table_name
WHERE condition (s)
```

Tips:

1. It's better to use more conditions compared to shows more rows
2. Always separate AND and OR with brackets
3. And and OR would be processed like conjunction and disjunction in logic math
4. For strings, use equal ('=') to filter with exact value and LIKE to search for the value

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Group By

Used to summarize the value and group it by specific criteria. We need to use AGGREGATE function to summarize numbers.

```
SELECT column1, column2,...
FROM table_name
GROUP BY column1, column2,...
```

Tips:

1. Can change column names into 1,2,3,.. sequential based on the columns in the SELECT
2. Don't need to include aggregated columns

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Having

Used to set limitation and/or condition to the query to filter the data based on needs using aggregated columns.

Need to use AND / OR to set multiple limitations.

```
SELECT column1, column2,...
FROM table_name
GROUP BY column1, column2,...
HAVING condition (s)
```

Tips:

1. It's better to use more conditions compared to shows more rows
2. Always separate AND and OR with brackets
3. And and OR would be processed like conjunction and disjunction in logic math
4. Never forget to put GROUP BY before the function

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Order By

Used to order the data based on certain column or more. Default would be set into ascending (ASC).

```
SELECT column1, column2,...
FROM table_name
ORDER BY column1, column2,...
```

Tips:

1. Can change column names into 1,2,3,.. sequential based on the columns in the SELECT
2. Use it only if only we need to see ordered result
3. Use LIMIT function to make the query faster

© Copyright by Digital Skola 2020

Limit

Insignificant for small data but would be a total mess if we didn't use it for big data especially if the occasion is only to explore.

```
SELECT column1, column2,...
FROM table_name
LIMIT number
```

Tips:

1. Use it accordingly to make the query faster
2. Significant to use before join and/or together with ORDER BY

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Query Processing Order

It’s not from top to bottom

1. Getting Data (From, Join)
2. Row Filter (Where)
3. Grouping (Group by)
4. Group Filter (Having)
5. Return Expressions (Select)
6. Order & Paging (Order by & Limit / Offset)

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Biggest Tips for Today!

Actually, ...

1. Neither courses nor bootcamp can cover 100% SQL functions that exist out there
2. But we could give you concept, context, and tips
3. Next, it's all on Google!
4. So, don't worry :)

© Copyright by Digital Skola 2020

DigitalSkola
Uncover The World Of Digital Skills With Us

Homework

Please do some research about ...

1. Window function and its example
2. Using filter inside SELECT
3. How to make an effective query

Let’s meet again on Monday to discuss your findings :)

© Copyright by Digital Skola 2020

DigitalSkola

Uncover The World Of Digital Skills With Us

Thank YOU

© Copyright by Digital Skola 2020