SQL: Data Definition Language and Data Manipulation Language

Anita Mila Oktafani

DQLab Live Class Data Analyst with SQL & Python in Google Platform

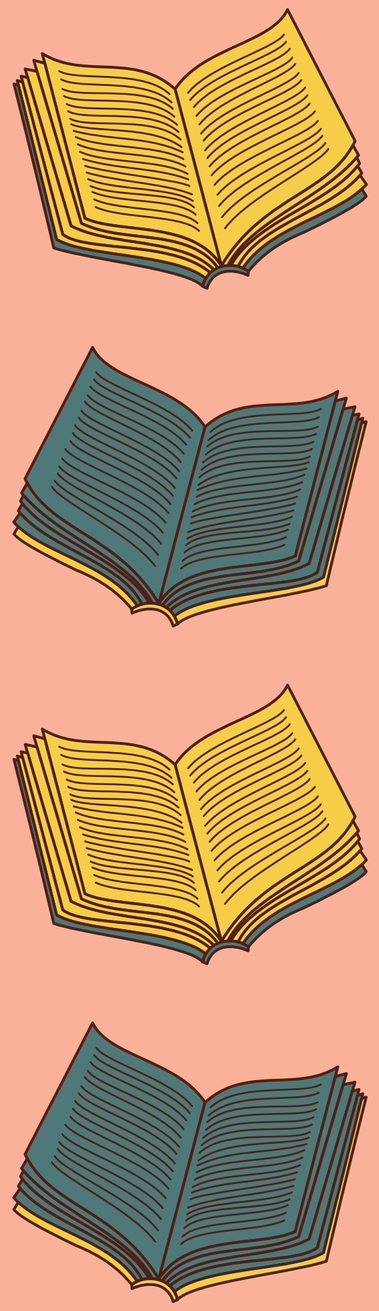
Table of Content

Data Definition Language (DDL)

What is DDL?

Data Definition Language (DDL) is a SQL command to create, modify, or even delete the database structures.

The command of DDL are as follows:
•CREATE -> used to create a database
•DROP -> used to delete a table or database



Create and Drop

Create

The database contains table to store entities. Tables consist of fields (columns) and records (rows). The CREATE command is used to create a table. This is the example of creating a table in Bigquery:

```
CREATE TABLE dqlab-409205.datacourse.course(
  course_id INT,
  course_name STRING(50) NOT NULL
);
```

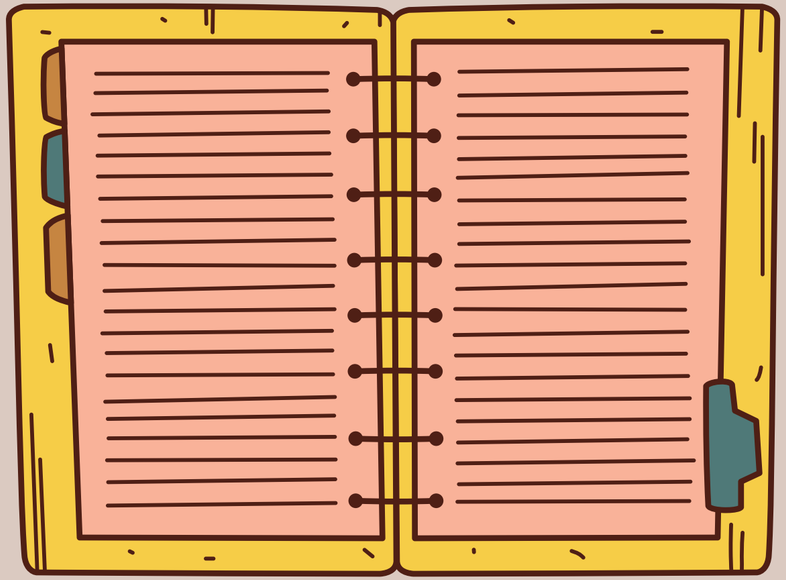
Field name	Type
course_id	INTEGER
course_name	STRING(50)

Drop

To delete an existing table, we can write the table name after DROP TABLE clause. If the table that was deleted does not exist, database system will issue an error message.

```
DROP TABLE dqlab-409205.datacourse.course;
```

Not found: Table dqlab-409205:datacourse.course was not found in location US



Query SELECT, SORT, AGREGATION, and HAVING

QUERY SELECT

To select data from a specific fields, specify the list of fields after the SELECT clause in the SELECT statement. For example, to select data from PassengerId, Name, Sex, and Age that contains all rows from titanic table.

specify fields

```
-- QUERY SELECT
SELECT PassengerId, Name, Sex, Age
FROM `dqlab-boootcamp.data_analytic.titanic_date`;
```

Row	PassengerId	Name	Sex	Age
1	180	Leonard, Mr. Lionel	male	36.0
2	264	Harrison, Mr. William	male	40.0
3	278	Parkes, Mr. Francis ...	male	null
4	303	Johnson, Mr. Willia...	male	19.0
5	414	Cunningham, Mr. Al...	male	null
6	467	Campbell, Mr. Willi...	male	null
7	482	Frost, Mr. Anthony ...	male	null
8	598	Johnson, Mr. Alfred	male	49.0

```
SELECT * FROM `dqlab-boootcamp.data_analytic.titanic_date`;
```

or we can use (*) to select all fields from the table

QUERY SORT

The ORDER BY clause is optional in the SELECT statement. The ORDER BY clause allows you to order the rows returned by SELECT clause based on one or more ordering expressions in ascending or descending order.

```
-- QUERY SORT
SELECT * FROM `dqlab-boootcamp.data_analytic.titanic_date`
ORDER BY Age ASC;
```

place the ORDER BY clause after the FROM clause

define an ordering expression after the ORDER BY clause.

```
SELECT * FROM `dqlab-boootcamp.data_analytic.titanic_date`
ORDER BY Age DESC;
```

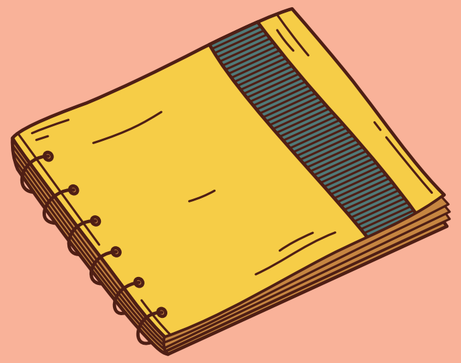
use the **ASC** option to sort the result set with an ascending sort expression and **DESC** to sort the result setwith a descending sorting expression.

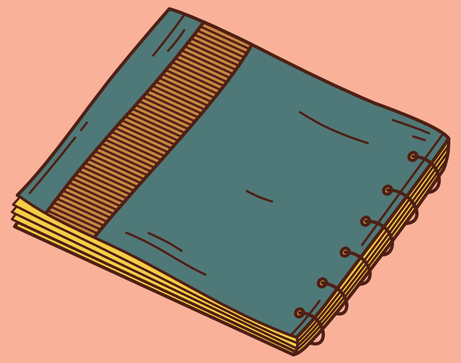
AGGREGATION

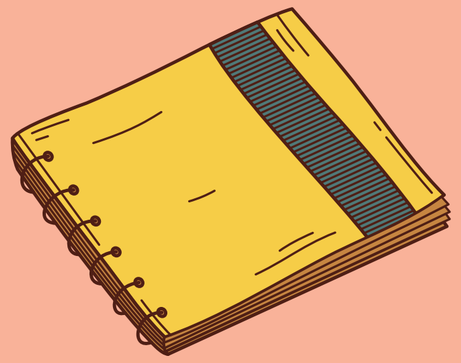
SQL aggregate functions perform calculations on a set of values and return one single value. Because aggregate functions operate on a set of values, they are often used with a GROUP BY clause in a SELECT statement. The GROUP BY clause splits the result set into groups of values, and the aggregate function returns one value for each group.

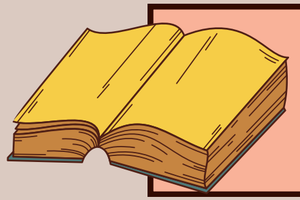
The following is an example of using an aggregate function with the GROUP BY clause:

- AVG()
- COUNT()
- MAX()
- MIN()
- SUM()









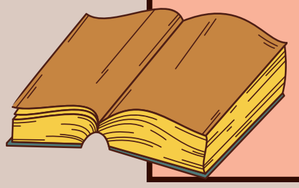
COUNT()

COUNT() function returns the number of items in a set.

```
-- COUNT() FUNCTION
SELECT Sex, COUNT(*) as COUNT #nama kolom count
FROM `dqlab-boootcamp.data_analytic.titanic_date`
GROUP BY Sex;
```

used a GROUP BY clause in a SELECT statement

Row	Sex	COUNT
1	male	577
2	female	314



AVG()

AVG() function returns the average of a set of values.

```
-- AVG() FUNCTION
SELECT Sex, AVG(Age) as Avg_age
FROM `dqlab-boootcamp.data_analytic.titanic_date`
GROUP BY Sex;
```

Row	Sex	Avg_age
1	male	30.72664459161...
2	female	27.91570881226...

MAX()

MAX() function returns the maximum value in a set.

```
-- MAX() FUNCTION
SELECT Sex, MAX(Age) as Max_age
FROM `dqlab-boootcamp.data_analytic.titanic_date`
GROUP BY Sex;
```

Row	Sex	Max_age
1	male	80.0
2	female	63.0

MIN()

MIN() function returns the minimum value in a set.

```
-- MIN() FUNCTION
SELECT Sex, MIN(Age) as Min_Age
FROM `dqlab-boootcamp.data_analytic.titanic_date
GROUP BY Sex;
```

Row	Sex	Min_Age
1	male	0.42
2	female	0.75

SUM()

SUM() function returns the sum of all distinct values or values in a set.

```
-- SUM() FUNCTION
SELECT Sex, SUM(Fare) as Sum_fare
FROM `dqlab-boootcamp.data_analytic.titanic_date
GROUP BY Sex;
```

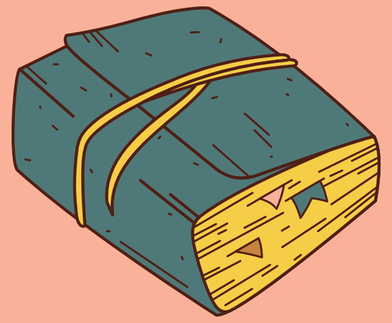
Row	Sex	Sum_fare
1	male	14727.28650000..
2	female	13966.66279999..

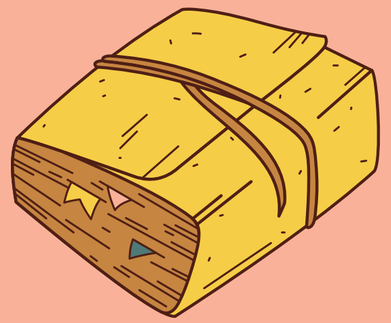
MORE THAN 1 AGGREGATE

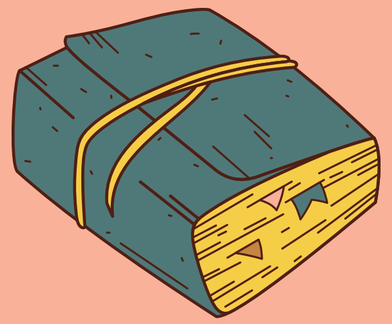
```
-- AGGREGATE >1
SELECT Sex
, COUNT(*) as Count
, AVG(Age) as Avg_age
, MIN(Age) as Min_age
, MAX(Age) as Max_age
FROM `dqlab-boootcamp.data_analytic.titanic_date`
GROUP BY Sex;
```

combine more than 1 aggregate

Row	Sex	Count	Avg_age	Min_age	Max_age
1	male	577	30.72664459161...	0.42	80.0
2	female	314	27.91570881226...	0.75	63.0







HAVING Statement

To specify group conditions, you use the HAVING clause. The HAVING clause is often used with the GROUP BY clause in a SELECT statement.

```
-- HAVING
SELECT Sex, Survived
, COUNT(*) as Count
, AVG(Age) as Avg_age
, MIN(Age) as Min_age
, MAX(Age) as Max_age
FROM `dqlab-boootcamp.data_analytic.titanic_date`
GROUP BY Sex, Survived
HAVING max_age >70;
```

used a HAVING after a GROUP BY clause in a SELECT statement

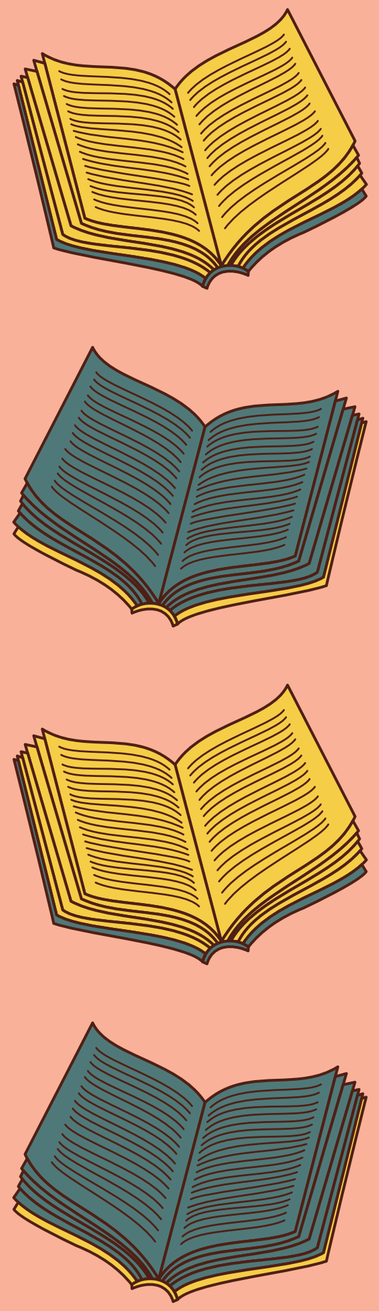
Row	Sex	Survived	Count	Avg_age	Min_age	Max_age
1	male	0	360	31.61805555555...	1.0	74.0
2	male	1	93	27.27602150537...	0.42	80.0

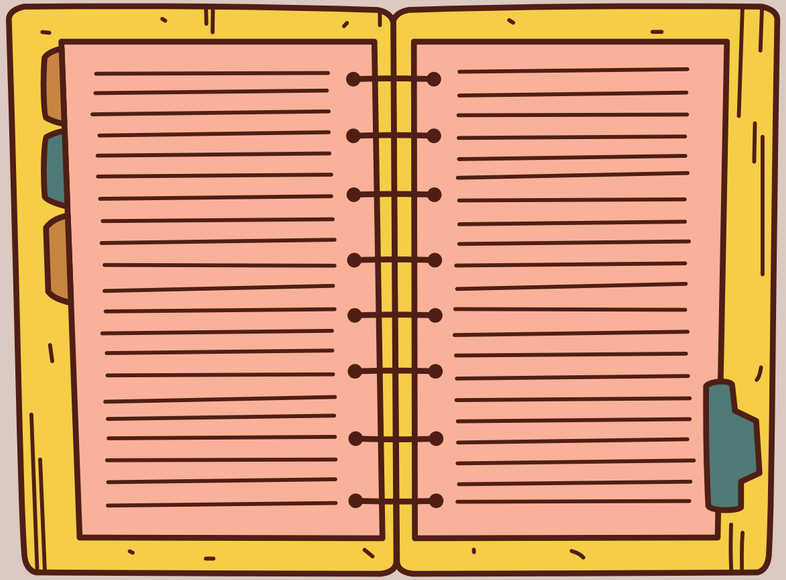
Data Manipulation Language (DML)

What is DML?

Data Manipulation Language (DML) is a SQL command for manipulation of data in tables.
DML commands include

- INSERT -> adds a record to the database
- UPDATE -> changes a record in the database
- DELETE -> deletes records in the database





Query INSERT, UPDATE and DELETE

QUERY INSERT

To insert a row into a table, you using the following syntax of the INSERT statement.

```
---------- Insert into manually

insert into `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final_trial`
values
(111111111,1,'1. Super Happy Customer') ,
(222222222,2,'2. Loyal Customer') ,
(333333333,2,'1. Super Happy Customer') ,
(444444444,2,'9. At Risk Customer') ,
(555555555,1,'9. At Risk Customer')
;

select * from `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final_trial` ;
```

Row	subs_no	brand	chi_segment_2
1	222222222	2	2. Loyal Customer
2	444444444	2	9. At Risk Customer
3	555555555	1	9. At Risk Customer
4	111111111	1	1. Super Happy Customer
5	333333333	2	1. Super Happy Customer

INSERT From SELECT Statement

We can use the INSERT statement to query data from one or more several tables and insert them into another table as follows:

```
--------- Insert Into from Table

insert into `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final_trial`
select subs_no, brand, chi_segment_2
from `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final`
where date_id = "2023-10-15" and brand = 2 and chi_segment_2 = '9. At Risk Customer'
limit 5
;

select * from `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final_trial` ;
```

Row	subs_no	brand	chi_segment_2
1	841044446	1	1. Super Happy Customer
2	840329346	1	1. Super Happy Customer
3	842005260	1	1. Super Happy Customer
4	841093326	1	1. Super Happy Customer
5	849551356	1	1. Super Happy Customer
6	2143911349	2	9. At Risk Customer
7	2143785330	2	9. At Risk Customer
8	2138971968	2	9. At Risk Customer
9	2143525152	2	9. At Risk Customer
10	2141634235	2	9. At Risk Customer

UPDATE Statement

To change existing data in a table, you use the UPDATE statement.

specify the table you want to update in the UPDATE clause

```
---------- Query Update

update `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final_trial`
set
  chi_segment_2 = '3. Happy New Customer'
where
  subs_no = 222222222
;

select * from `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final_trial` ;
```

specify the columns whose values you want to change in the SET clause

specify the rows you want to update in the WHERE clause

Row	subs_no	brand	chi_segment_2
1	222222222	2	3. Happy New Customer
2	555555555	1	9. At Risk Customer
3	111111111	1	1. Super Happy Customer
4	333333333	2	1. Super Happy Customer
5	444444444	2	9. At Risk Customer

DELETE Statement

To delete one or more rows from a table, you use a statement DELETE

```
--------- Delete from Table with Condition

delete from `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final_trial`
where brand = 1 ;

select * from `cvm-cloud-dwh-prod-4383.prd_cvm_analytic_tb.cx_chi_segmentation_final_trial` ;
```

provide the name of the table where you want to delete the rows.

define conditions in the WHERE clause to identify the necessary rows deleted. If you omit the WHERE clause, all rows in the table will be deleted.

Row	subs_no	brand	chi_segment_2
1	2143911349	2	9. At Risk Customer
2	2143785330	2	9. At Risk Customer
3	2138971968	2	9. At Risk Customer
4	2143525152	2	9. At Risk Customer
5	2141634235	2	9. At Risk Customer

we should always use the DELETE statement with caution

THANK YOU!!

LinkedIn:
https://www.linkedin.com/in/anitamilaoktafani/

DQLab Live Class Data Analyst with SQL & Python in Google Platform