SQL CRASH COURSE

- GQ March 2021

Plan in brief

1. Go through all this document
2. Do the queries
3. Deliver them to your mentor as per instructions

SQL Intro

SQL stands for structured query language. It is a language suited to handle data with structure, i.e. data organized in entities (~tables) with clear relations among them.

SQL, or SEQUEL, is a language used to create, access and manage relational databases, created by IBM in 1974; it is based on Codd’s relational model (1970).

SQL consists of the following groups of languages.

- Data Definition Language (DDL)
    - To create and define a database
    - Commands: CREATE, ALTER, DROP
- Data Manipulation Language (DML)
    - To read data from tables and modify them
    - Commands: SELECT, INSERT, DELETE, UPDATE
- Data Control Language (DCL)
    - To control access to the database
    - Commands: GRANT, REVOKE

We will focus on a part of the SQL language called DQL (Data Query Language) which is the set of commands to get data from an existing database, without modifying the db itself, a bit like some powerful read-only commands. It is a subset of DML.

Data in RDBMS (Relational DataBase Management Systems) are stored in database objects called tables. A table (entity), similarly to a dataframe, is a collection of related data entries (entities instances) and it consists of columns and rows, each row being a record. Different tables are connected with clear relations, that’s why it’s called Relational DBMS.

We will use MySQL, since it is widespread, scalable and includes a free version. Many companies like Facebook, Google and Adobe rely on MySQL for their websites. The most common use for mySQL is in fact for web database purposes. It has a client-server structure. Here to know more.

With close to negligible differences the syntax of a query made in MySQL will be the same in other RDBMS eg postgreSQL, MariaDB, MS SQL Server, Sequel Pro, SQLite.

Setting up the tools

The first step will be to set up the necessary tools to perform some queries.

Follow the links to install both MySQL and MySQL Workbench, they are a bit like R and RStudio or python and pycharm/jupyter: Windows (ignore steps for large dbs), Mac, Ubuntu.

NB Choose a simple password you won’t forget to prevent boring retrieval procedures.

Get the SQL database

1. Download .sql files attached to email and that you can find here
2. Open MySQL workbench
3. Open local connection
4. From workbench sql menu (top left corner button) open the .sql file and press the lightning button ⚡ to run the code that will appear; while it runs it will create the tables (it may take some minutes)
5. Same for the other 2 sql files
6. Et voilà, you have your db ready to be queried.

Basic concepts of Database Management Systems

Before starting the task, please go through the basic concepts of what a database is and get familiar with basic terminology as explained in this document and thisDatabase & E-R Model other one.Database & E-R Model

What is a query ?

A query is a request for data or information from a database table or a combination of tables. It helps users retrieve the data they need. Please refer to these slides for an overview of SQL queries.

Here you'll find a tutorial with the most common commands explanations to introduce you to the usage of basic SQL queries. Go through all of it or read a bit and then go straight to the query exercises; you can go back to the tutorial when you need it.

Query the db: explore each table one by one

Once you have access to the database on MySQL Workbench, create the following requested queries. They are organized by the three tables: *line_items*, *orders* and *products*.

Some tips follow.

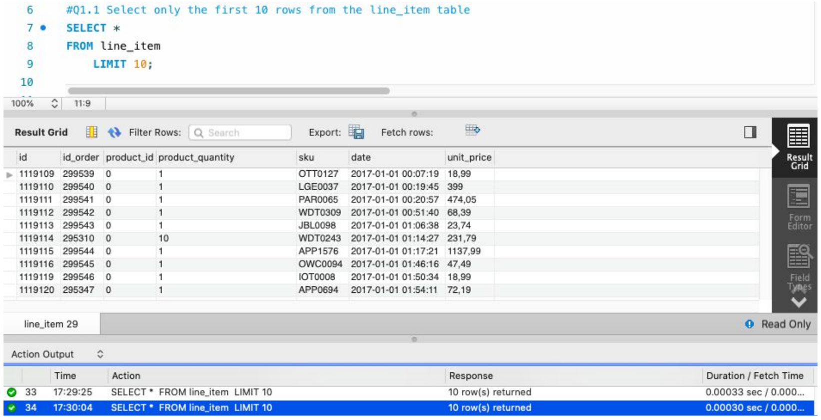
- To comment:
  - # inline comment
  - -- inline comment
  - /*multiple line
    comment*/
- Don't forget the semicolon at the end of each query.
- Commands are usually written in uppercase letters, while tables' names lowercase and usually in the singular form (in our case it's plural like in *orders*).
- Shortcuts:
  - Execute (all or selection) -> Ctrl + Shift + Enter.
  - Execute Current Statement -> Ctrl + Enter.

1. Table: *line_items*

#Q1. Select the entire line_item table.

#Q1.1 Select only the first 10 rows from the line_item table

Expected output:



```
#Q1.1 Select only the first 10 rows from the line_item table
SELECT *
FROM line_item
    LIMIT 10;
```

id	id_order	product_id	product_quantity	sku	date	unit_price
1119109	299539	0	1	OTT0127	2017-01-01 00:07:19	18,99
1119110	299540	0	1	LGE0037	2017-01-01 00:19:45	399
1119111	299541	0	1	PAR0065	2017-01-01 00:20:57	474,05
1119112	299542	0	1	WDT0309	2017-01-01 00:51:40	68,39
1119113	299543	0	1	JBL0098	2017-01-01 01:06:38	23,74
1119114	295310	0	10	WDT0243	2017-01-01 01:14:27	231,79
1119115	299544	0	1	APP1576	2017-01-01 01:17:21	1137,99
1119116	299545	0	1	OWC0094	2017-01-01 01:46:16	47,49
1119119	299546	0	1	IOT0008	2017-01-01 01:50:34	18,99
1119120	295347	0	1	APP0694	2017-01-01 01:54:11	72,19

	Time	Action	Response	Duration / Fetch Time
33	17:29:25	SELECT * FROM line_item LIMIT 10	10 row(s) returned	0.00033 sec / 0.000...
34	17:30:04	SELECT * FROM line_item LIMIT 10	10 row(s) returned	0.00030 sec / 0.000...

You can check how many rows were returned here ↑ , the default limit is 1000 rows.

#Q1.2. Select only the columns "stock keeping unit" (sku), unit_price and date from the line_item table (only the first 10 rows)

#Q2. Count the total number of rows of the line_item table

# Expected output: 293983

#Q2.1. Count the total number of unique "sku" from the line_item table.

#Expected output: 7951

#Q3. Generate a table with the average price of each sku.

#Q3.1. Now name the column of the previous query with the average price "avg_price" through and alias, and sort the list by that column (bigger to smaller price)

#Q4. Which products were bought in the largest quantities? Select the “stock keeping unit” (sku) and product_quantity of the 100 products with the biggest "product quantity"

1. Table: *orders*

#Q5. How many orders were placed in total?

#Q6. How many orders by state?

#Q7. Select all the orders placed in January of 2017

#Q8. How many orders were placed in January of 2017?

#Q9. How many orders were cancelled on January 4th 2017?

#Expected output: 23

#Q10. How many orders have been placed each month of the year?

#Q11. What is the total amount paid in all the orders?

#Expected output: 129,011,388

#Q12 What is the average amount paid per order? Give a result to the previous question with only 2 decimals

#Expected output: 568.56

#Q13 What is the date of the newest order?

# Expected output: 2017-01-01 00:07:19

#Q13.1 What about the oldest?

# Expected output: 2018-03-14 13:58:36

#Q13.2 What is the day with the highest amount paid (and how much was paid that day)?

# Expected output: 2017-11-24; 3,103,713

#Q13.3 What is the day with the highest amount of completed orders (and how many completed orders were placed that day)?

# Expected output: 2017-11-24; 1,569

2. Table: products

#Q14 -- How many products are there?

# Q15 -- How many brands?

# Q16 -- How many categories?

#Q16.1 -- How many products per brand ?

# Q16.2 -- How many products per category?

# Q17.1 -- What is the average price per brand?

# Q17.2 -- What is the average price per category?

# Q18.1 -- What is the name and description of the most expensive product per brand ?

# Q18.2 -- What is the name and description of the most expensive product per category?

About 18.2 Apparently there are differences between mac and windows.

The following query works only on windows.

```
SELECT brand,  MAX(price) as max_price, name_en, short_desc_en, ProductId
        FROM products
                group by brand;
```

It is not clear, though, how it chooses name_en and short_desc_en

On macs the previous query does not work, but following does.

```
SELECT brand, MAX(price) as max_price
        FROM products
                group by brand;
```

I think that the problem is the following. You can have more than one product (thus name and description) per category with the same price which is also the maximum price for that category, so it is ambiguous and it can't choose between these. Maybe if you are using windows it takes the first one or a random one.

Deliverable instructions

Deliver one pdf with your queries.

Write one SQL statement per question. When you find it proper in terms of readability, order your output and use aliases. Format large numbers and fractions appropriately.

For each question, present an answer in the following format:

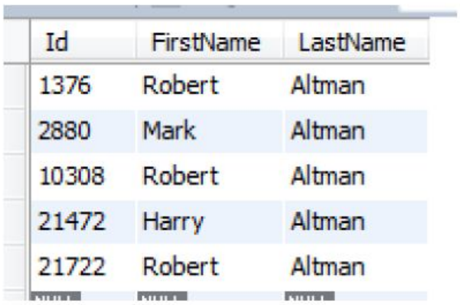
- Show the question number and question in black text;
- Show your answer (the SQL statement) in blue text (no screenshot);
- Show a screenshot from the Workbench showing output of 10 or fewer lines;
- Show how many rows were returned, in red text;
- SQL queries must be formatted in an easy-to-read fashion, i.e. placing most clauses on new lines, indenting subqueries, using caps lock for commands and lowercase for tables etc.

E.G.

7. List all users with the last name ‘Altman’

```
select * from User

where LastName = 'Altman';
```



Id	FirstName	LastName
1376	Robert	Altman
2880	Mark	Altman
10308	Robert	Altman
21472	Harry	Altman
21722	Robert	Altman

5 Rows Returned