2023

MySQL Database Queries

Jelena Cekmeniova

Cohort: GLA 16

Submission Date: 18.08.2023

Contents

Table of Figures

1. My SQL Project – Overview

“MySQL is an open-source relational database management system” that employs “SQL (Structured Query Language) to access this database and extract stored data” (Oracle, 2023).

This assignment intends to illustrates database query writing process, filtering and creating Relational Database Schema utilising MySQL Workbench. Database ‘store’ contains six tables (‘customers’, ‘order_items’, ‘order_statuses’, ‘orders’, ‘products’ and ‘shippers’), however, mainly ‘customers’ and ‘orders’ tables are used to complete demonstration.

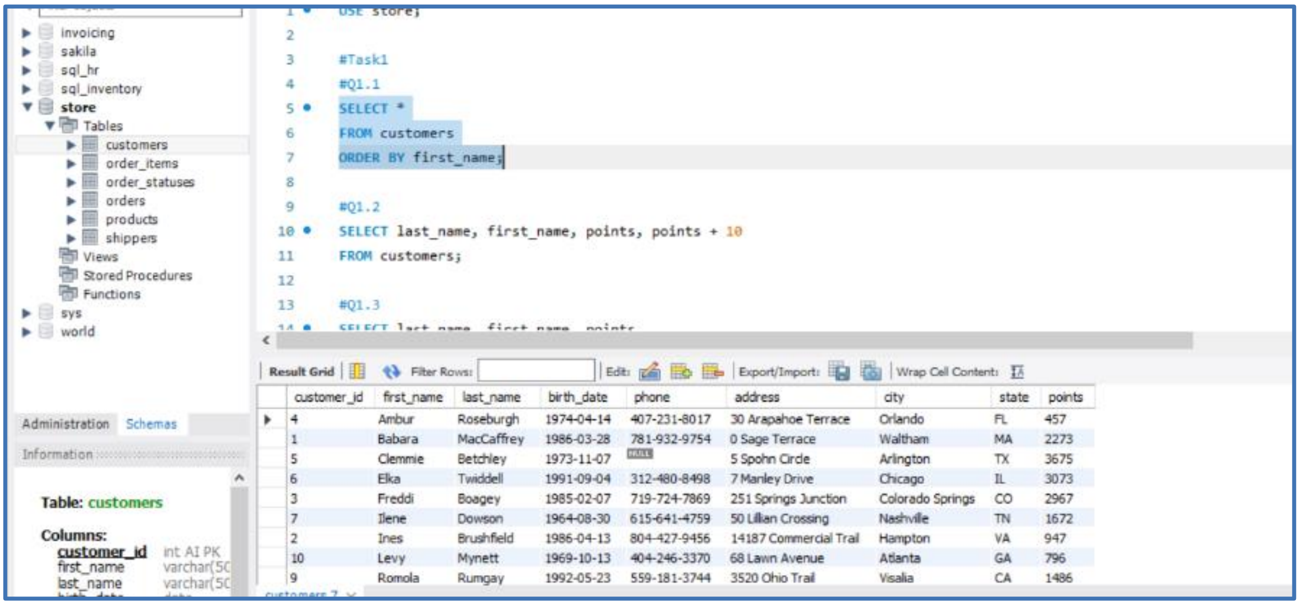
To load database file to MySQL Workbench, it was opened directly from software screen by clicking “Open a SQL Script File…” icon and selecting “create-db-store” file, then executing it. Schema representation of database and corresponding tables and fields can be seen in pane on left-hand side.

Finally, before starting querying, ‘USE store’ command was given in new query window to indicate which database is going to be utilised.

2. Task 1 – Working with ‘customers’ Table

To select all data from ‘customers’ table and order it by customers’ ‘first_name’, the following input was given, using “SELECT” statement and “*” to select “all” (Fig. 1).

Fig.1 – Selecting All Data from Table



```
USE store;

#Task1
#Q1.1
SELECT *
FROM customers
ORDER BY first_name;

#Q1.2
SELECT last_name, first_name, points, points + 10
FROM customers;

#Q1.3
SELECT last_name, first_name, points
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486

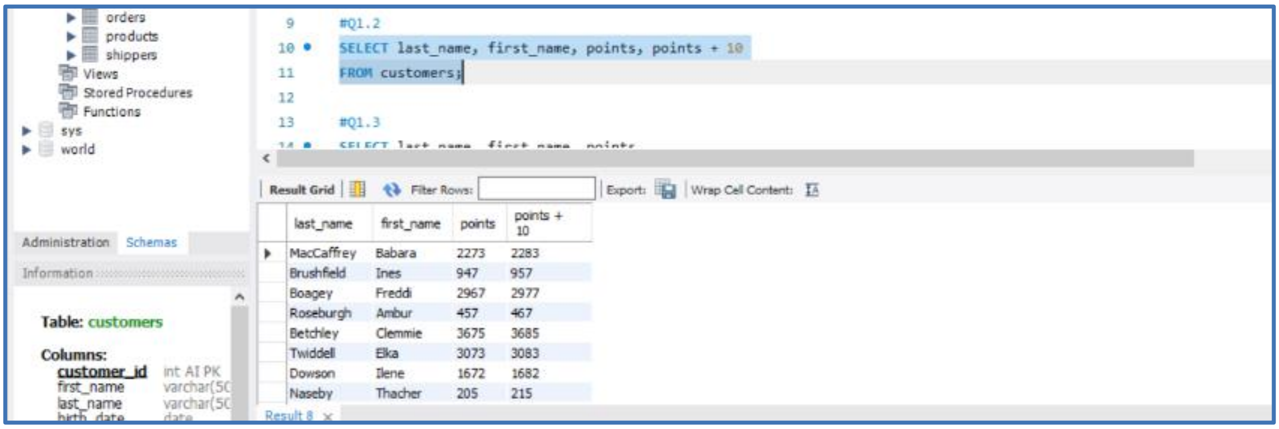
Table: customers

Columns:

customer_id	int AI PK
first_name	varchar(5C
last_name	varchar(5C

Further, to select specific columns from ‘customers’ table and add additional column with calculation based on numeric data in another column, “*” was replaced with specific columns’ names. Additionally, “+” operator was used to add “10” to numeric data in column ‘points’ to create additional field ‘point + 10’ (Fig.2) ((MySQL, 2023).

Fig. 2 – Selecting Specific Columns



```
#Q1.2
SELECT last_name, first_name, points, points + 10
FROM customers;

#Q1.3
SELECT last_name, first_name, points
```

Table: customers

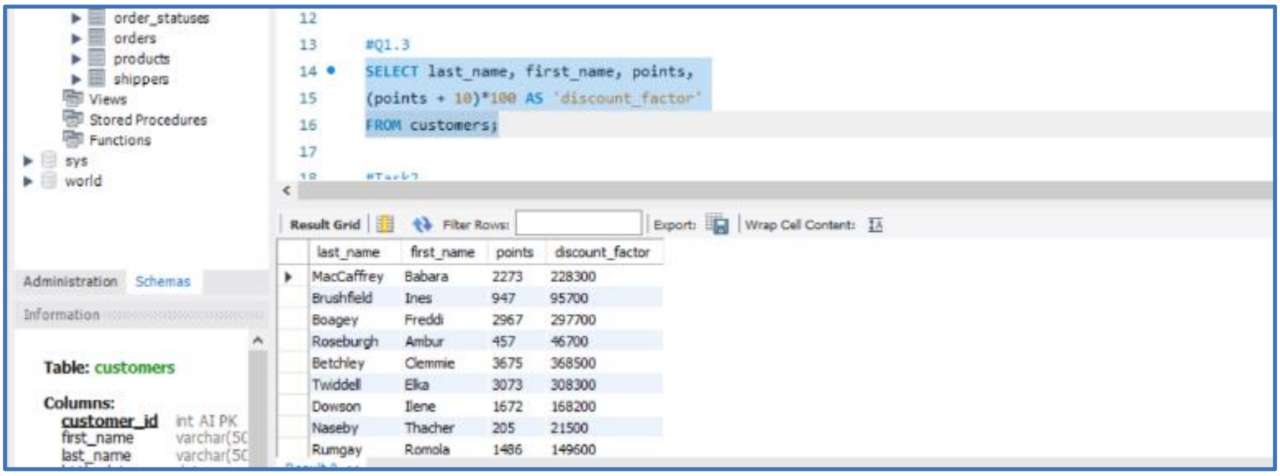
Columns:

customer_id	int AI PK
first_name	varchar(5C
last_name	varchar(5C
birth_date	date

last_name	first_name	points	points + 10
MacCaffrey	Babara	2273	2283
Brushfield	Ines	947	957
Boagey	Freddi	2967	2977
Roseburgh	Ambur	457	467
Betchley	Clemmie	3675	3685
Twiddell	Elka	3073	3083
Dowson	Ilene	1672	1682
Naseby	Thacher	205	215

To give temporary, but more meaningful name to the newly created column or so-called alias (Knowles, 2022) - “discount_factor” was used with statement “AS” (Fig. 3).

Fig.3 – Aliases



```
#Q1.3
SELECT last_name, first_name, points,
(points + 10)*100 AS 'discount_factor'
FROM customers;

#Task2
```

Table: customers

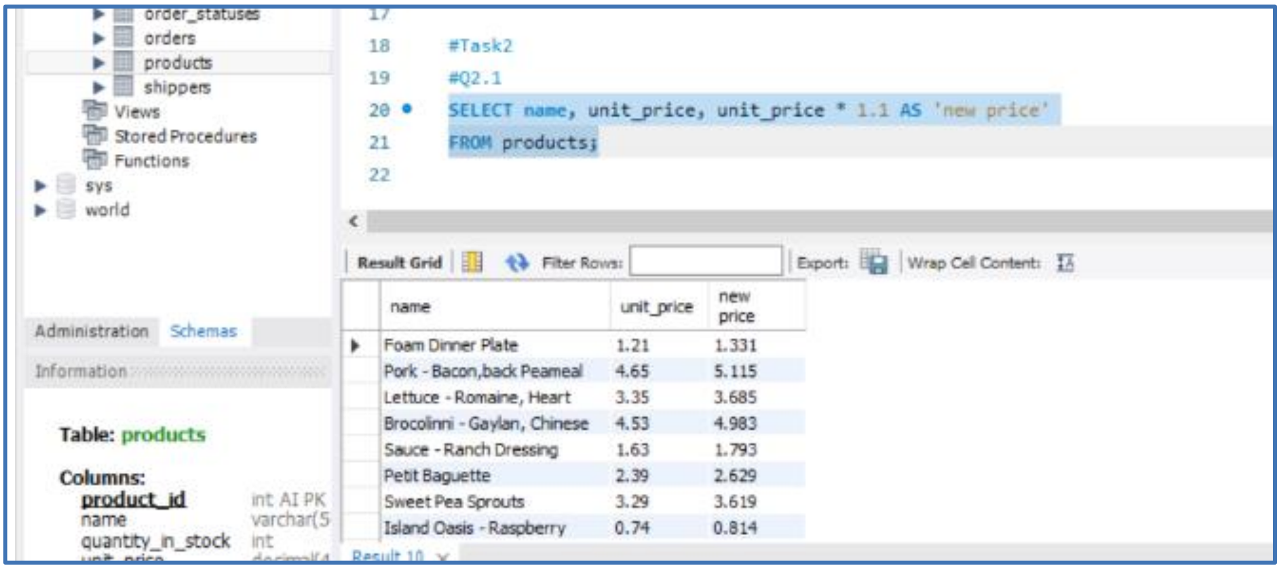
Columns:

customer_id	int AI PK
first_name	varchar(5C
last_name	varchar(5C

last_name	first_name	points	discount_factor
MacCaffrey	Babara	2273	228300
Brushfield	Ines	947	95700
Boagey	Freddi	2967	297700
Roseburgh	Ambur	457	46700
Betchley	Clemmie	3675	368500
Twiddell	Elka	3073	308300
Dowson	Ilene	1672	168200
Naseby	Thacher	205	21500
Rumgay	Romola	1486	149600

3. Task 2 – Working with ‘products’ Table

Fig. 4 – Querying ‘product’ Table and Adding Alias



```
#Task2
#Q2.1
SELECT name, unit_price, unit_price * 1.1 AS 'new price'
FROM products;
```

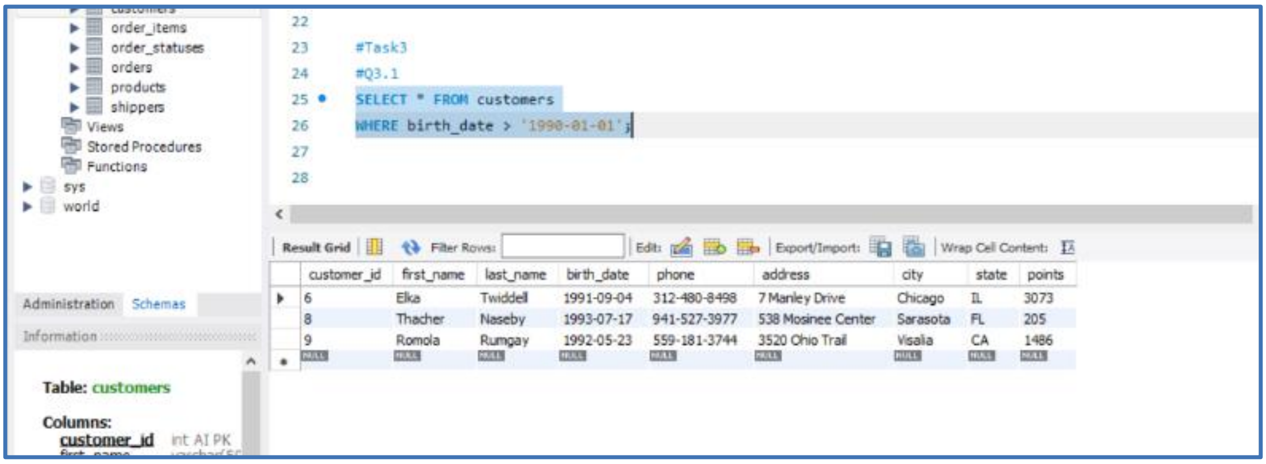
name	unit_price	new price
Foam Dinner Plate	1.21	1.331
Pork - Bacon,back Peameal	4.65	5.115
Lettuce - Romaine, Heart	3.35	3.685
Brocolinni - Gaylan, Chinese	4.53	4.983
Sauce - Ranch Dressing	1.63	1.793
Petit Baguette	2.39	2.629
Sweet Pea Sprouts	3.29	3.619
Island Oasis - Raspberry	0.74	0.814

Fig. 4 above shows that to complete Task 2 the similar set of statements was used as in Task 1. “SELECT“ statement is followed by specific columns’ names (‘name’, ‘unit_price’ and ‘unit_price * 1.1’, with the last temporary renamed as ‘new price’ with statement “AS”. All information was taken from ‘products’ table.

4. Task 3 – Using Comparison Operator

Task 3 required to utilise comparison operator “greater than” in conjunction with WHERE statement (MySQL, 2023), in order to find all available data for customers with date of birth that is greater than “1990-01-01” (Fig. 5).

Fig. 5 – Using “WHERE” Statement and Comparison Operator



```
#Task3
#Q3.1
SELECT * FROM customers
WHERE birth_date > '1990-01-01';
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. Task 4 - Writing More Complex Query

Fig. 6 – Joining Two Tables

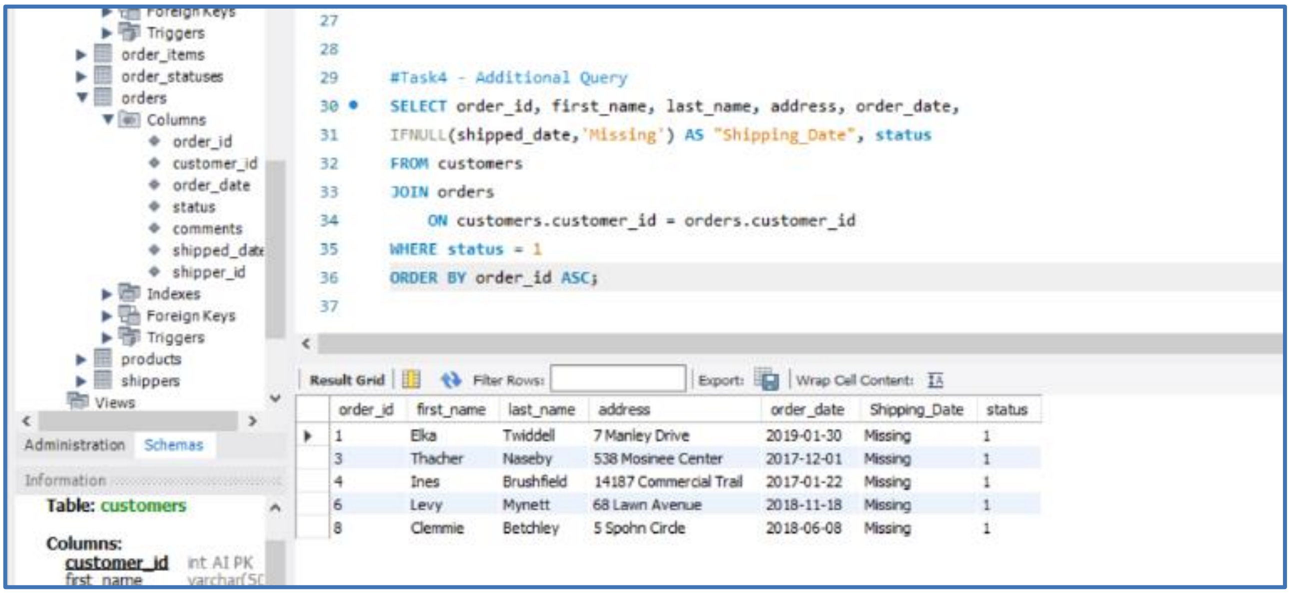


Table: customers

Columns:
customer_id int AI PK
first_name varchar(5(

```
#Task4 - Additional Query
SELECT order_id, first_name, last_name, address, order_date,
IFNULL(shipped_date,'Missing') AS "Shipping_Date", status
FROM customers
JOIN orders
    ON customers.customer_id = orders.customer_id
WHERE status = 1
ORDER BY order_id ASC;
```

order_id	first_name	last_name	address	order_date	Shipping_Date	status
1	Elka	Twiddell	7 Manley Drive	2019-01-30	Missing	1
3	Thacher	Naseby	538 Mosinee Center	2017-12-01	Missing	1
4	Ines	Brushfield	14187 Commercial Trail	2017-01-22	Missing	1
6	Levy	Mynett	68 Lawn Avenue	2018-11-18	Missing	1
8	Clemmie	Betchley	5 Spohn Circle	2018-06-08	Missing	1

Above query, (Fig. 6), demonstrates how function “JOIN” works connecting columns from two different tables – ‘customers’ and ‘orders.

Good to note that order, in which tables are displayed on the screen, correspond to order in which columns are selected after “SELECT” statement, mixing columns from both tables in desired by user order (‘order_id’, ‘first_name’, ‘last_name’, ‘order_date’, ‘Shipping_Date’, ‘status’) (Knowles, 2022).

Furthermore, since original ‘shipped_date’ column contains many NULL values, they are replaced with alias “missing” and new column given temporary name ‘Shipping_Date’ (MySQL, 2023).

“JOIN“ function allows to join two tables together, based on fact that both have common column – ‘customer_id’, which represent Primary Key for ‘customers’ table and Foreign Key for ‘orders’ table (W3Schools, 2023).

Finally, WHERE statement is utilised to select just records where ‘status’ is equal 1, and whole table is ordered by ‘order_id’ in ascending order (Knowles, 2022).

Other queries can be found in ‘store_sql_assignment’ script, emailed together with this assignment document.

6. Task 5 – Creating Relational Database Schema Diagram

EER (Enhanced Entity – Relationship) diagram for “store” database (Fig.7) represents relationship and cardinality between the tables within database (MySQL, 2023). As indicated in the beginning of assignment, there are six tables in total – ‘customers’, ‘orders’, ‘order_statuses’, ‘shippers’, ‘order_items’ and ‘products.

Relationships between tables are mainly established on Primary (PK) and Foreign Keys (FK) as a common column within both tables (MySQL, 2023). For example, Primary Key in ‘customers’ table is ‘customer_id’ is also Foreign Key in ‘orders’ table. Other relationships can be identified as following:

- ‘orders’ table: ‘order_id’ - PK, ‘customer_id’ and ‘status’ – FKs;
- ‘order_statuses’ table: ‘order_status_id’ – PK;
- ‘shippers’ table: ‘shipper_id’ – PK;
- ‘products’ table: ‘product_id’ – PK;
- ‘order_items’ table – ‘order_id’ and ‘product_id’ – Composite Keys (CKs).

Cardinality explains how tables are related in numerical aspect, with main ones one-to-one, one-to-many and many-to many. Specific rules are usually set by the company (Lucidchart, 2023). For instance, according to below EER diagram, orders can have one and only one status (1 or 2) at the same time, while order statuses are allocated to multiple orders. Same situation with ‘shippers’ and ‘orders’, where multiple orders might be allocated to one shipper, but specific order can be shipped by one and only one shipper. However, company may change the rules and orders can be split and delivered my different shippers.

This EER diagram also illustrates use of Composite Keys that can utilise multiple columns for each record identification. These are also usually used to describe many-to-many relationship through additional column between them (Knowles, 2022). For example, ‘order_items’ connects ‘orders’ and ‘products’ into many-to-many relationship, taking both Primary key – ‘order_id’ from ‘orders and Primary Key – ‘product_id’ from ‘products’ and using them as Composite Key.

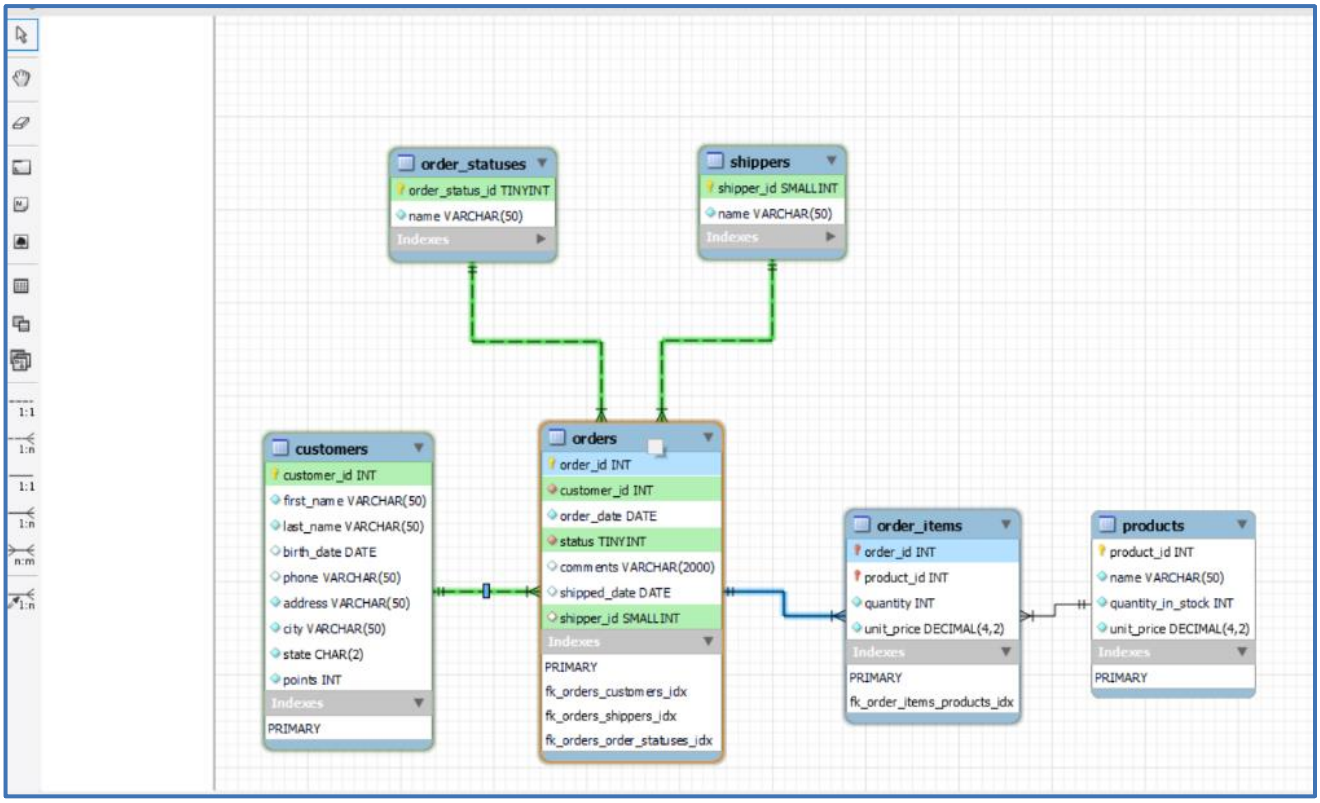
Additionally, solid and dotted lines between tables describe magnitude of such relationships. Therefore, solid lines represent “identified relationship”, where child table cannot be recognised without parents table and often used to highlight many to many relationships, mentioned above. On another hand, dotted line signifies that child table might be identified from parent table indipendantly (Lucidchart, 2023).

Finally, EER diagram also shows constraints imposed by author of original ‘store’ database script.

CONSTRAINT `fk_orders_order_statuses` FOREIGN KEY (`status`) REFERENCES `order_statuses` (`order_status_id`) ON UPDATE CASCADE

Line above describes constrain attached to Foreign Key - ‘status’ in ‘orders’ table, referencing Primary Key – ‘order_status_id’ in ‘order_statuses’ table. ‘ON UPDATE CASCADE’ statement requests to update records in child table ‘orders’, when corresponding records are updated in parent table ‘order_statuses’ (JavaTpoint, 2021).

Fig. 7 – EER (Enhanced Entity – Relationship) Diagram



7. Summary

In sum, using MySQL database management system is an essential tool that allows data analyst to extract information about various entities from company's database. Author of this assignment believes that MySQL system is relatively easy to learn and to use, as it has rather limited number of functions/ operators, however, tend to have inbuilt scalability and robust performance, especially in conjunction with other programming languages.

8. References

JavaTpoint (2021) *MySQL ON DELETE CASCADE*. JavaTpoint. Available from: https://www.javatpoint.com/mysql-on-delete-cascade#:~:text=MySQL%20ON%20UPDATE%20CASCADE,rows%20in%20the%20parent%20table. [Accessed 17 August 2023].

Knowles, C. (2022) *SQL for Beginners: An Informative and In-depth Guide to SQL Programming*. Independently published.

Lucidchart (2023) *What is an Entity Relationship Diagram (ERD)?* Lucid Software Inc. Available from: https://www.lucidchart.com/pages/er-diagrams [Accessed 17 August 2023].

MySQL (2023) *Arithmetic Operators*. Oracle. Available from: https://dev.mysql.com/doc/refman/8.0/en/arithmetic-functions.html#operator_plus [Accessed 17 August 2023].

MySQL (2023) *Comparison Functions and Operators*. Oracle. Available from: https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html#operator_greater-than [Accessed 17 August 2023].

MySQL (2023) *EER Diagrams*. Oracle. Available from: https://dev.mysql.com/doc/workbench/en/wb-eer-diagrams-section.html#:~:text=Enhanced%20Entity%2DRelationship%20(EER),shown%20in%20the%20associated%20diagram. [Accessed 17 August 2023].

MySQL (2023) *Problems with Column Aliases*. Oracle. Available from: https://dev.mysql.com/doc/refman/8.0/en/problems-with-alias.html [Accessed 17 August 2023].

Oracle (2023) *What is MySQL?* Oracle. Available from: https://www.oracle.com/mysql/what-is-mysql/ [Accessed 17 August 2023].

W3Schools (2023) *SQL Joins*. W3Schools. Available from: https://www.w3schools.com/sql/sql_join.asp [Accessed 17 August 2023].