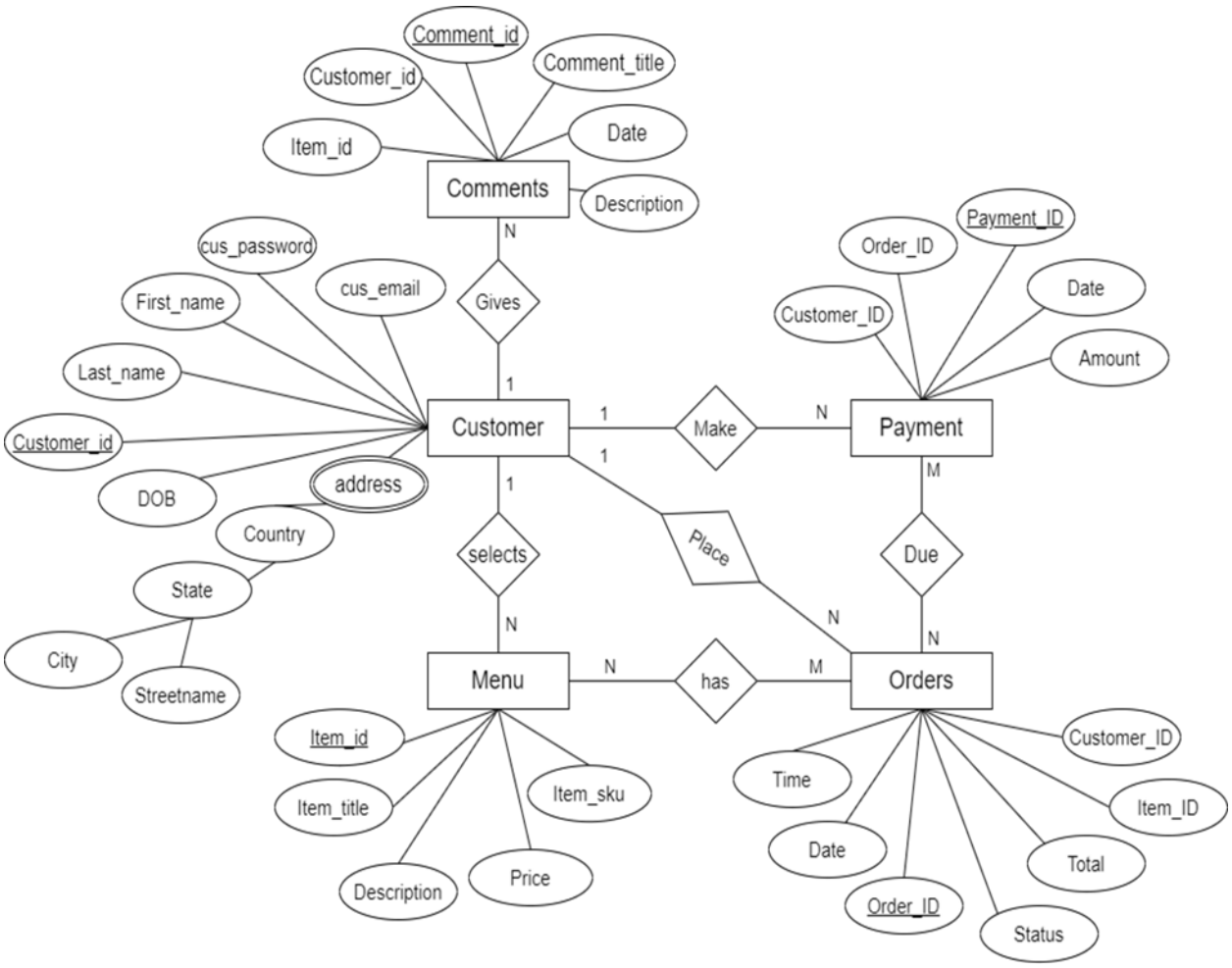
CS540 – Advanced Database Administrator

Group Project: Online Food Order

The online food order system database developed on the MySQL. To manage the container database, we have used visual studio code and for the database instance we have used MySQL workbench, MySQL Shell and MySQL command line client to run the DBA level commands.

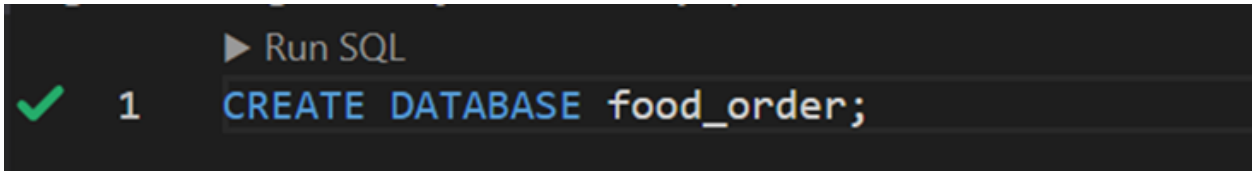
Overview of ER Diagram



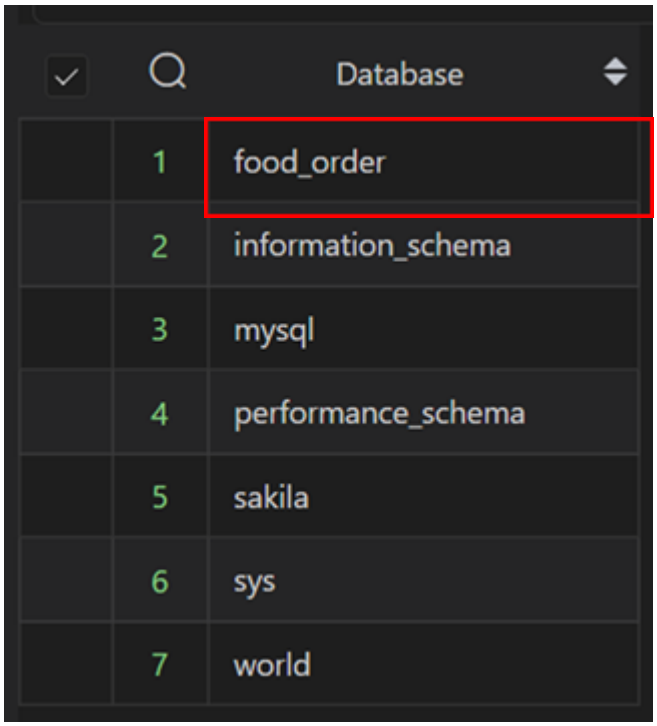
Entities: Customer, Comments, Payment, Menu, Orders.

To manage the database, Database Administrator commands are following:

1. Create the database:



Output:



2. Create the User “Jenish”.

```
mysql> create user 'jenish'@'localhost' identified by 'Jenish1';
Query OK, 0 rows affected (0.15 sec)
```

3. Check the default privileges of the Jenish user.

```
mysql> show grants for 'jenish'@'localhost';
+-----------------------------------------------+
| Grants for jenish@localhost                   |
+-----------------------------------------------+
| GRANT USAGE ON *.* TO `jenish`@`localhost`    |
+-----------------------------------------------+
1 row in set (0.00 sec)
```

4. Connect to the database a jenish user.

```
MySQL  SQL > \connect jenish@localhost
Creating a session to 'jenish@localhost'
Please provide the password for 'jenish@localhost': *******
Save password for 'jenish@localhost'? [Y]es/[N]o/Ne[v]er (default No): y
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 47 (X protocol)
Server version: 8.0.28 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL  localhost:33060+ ssl  SQL >
```

5. Check the default database access of user.

```
MySQL  localhost:33060+ ssl  SQL > show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
+--------------------+
1 row in set (0.0022 sec)
MySQL  localhost:33060+ ssl  SQL >
```

The result only shows the default schema database because the admin still does provide any privileges to the Jenish user to access any other database.

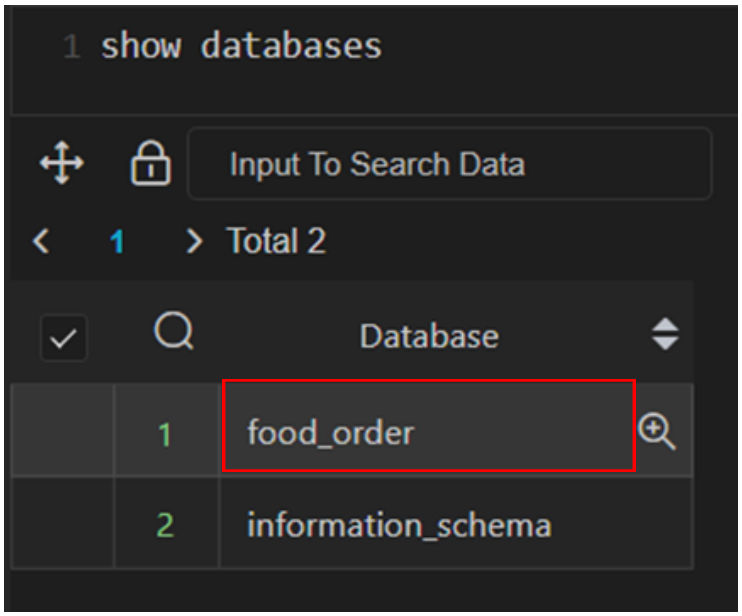
6. Grant the privileges of the food_order database to Jenish user.

```
mysql> grant all on food_order.* to 'jenish'@'localhost';
Query OK, 0 rows affected (0.15 sec)
```

7. Check the privileges of a Jenish user.

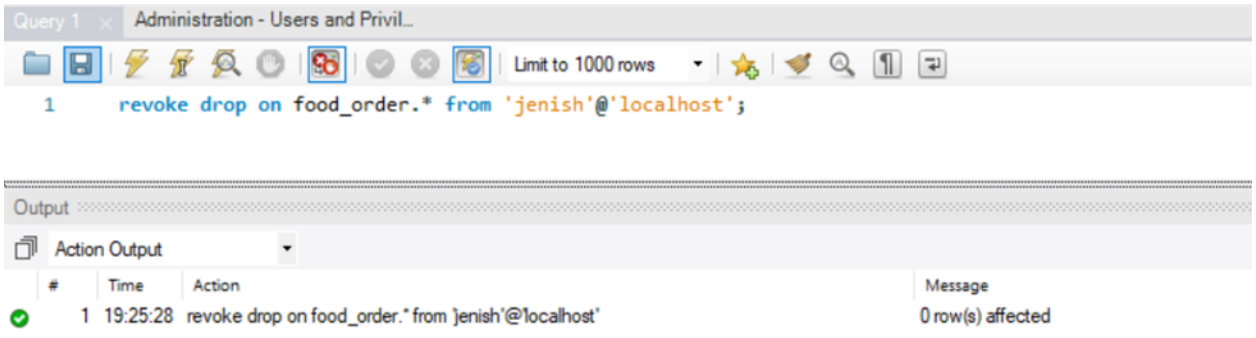
```
mysql> show grants for 'jenish'@'localhost';
+-----------------------------------------------------------------------+
| Grants for jenish@localhost                                           |
+-----------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `jenish`@`localhost`                            |
| GRANT ALL PRIVILEGES ON `food_order`.* TO `jenish`@`localhost`        |
+-----------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

8. Again check the database of as Jenish user.

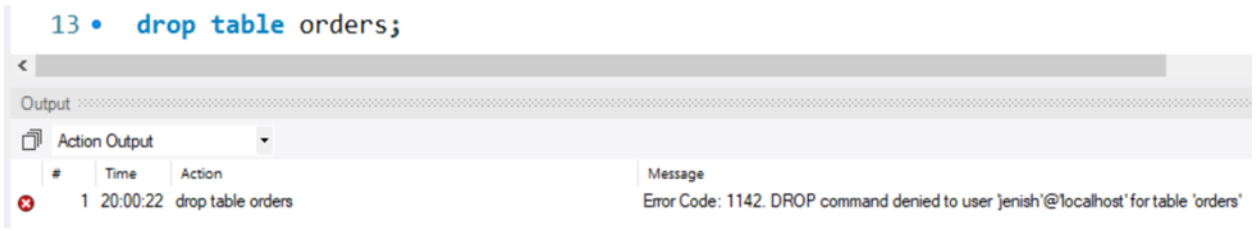


This time the user can see the food_order because the admin has given all the privileges of the Food_order database to the user.

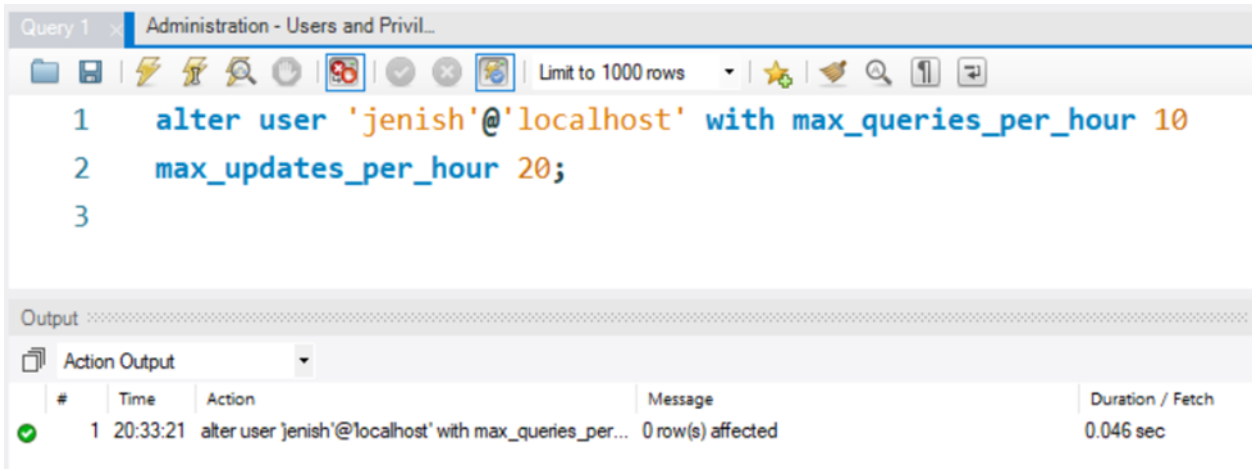
9. Revoke drop privilege of user Jenish on Food-order database



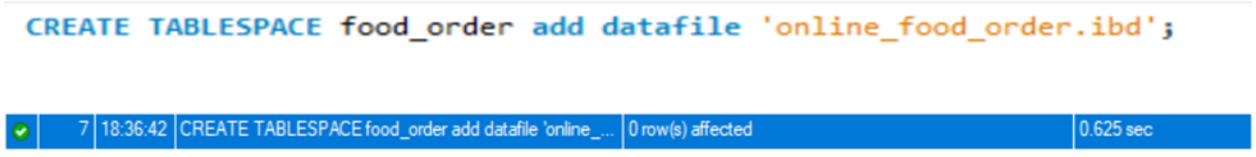
10. Try to drop table as Jenish user



11. Alter the privileges on Jenish User, where users can only run 10 queries per hour and can do only 20 updates in the database.



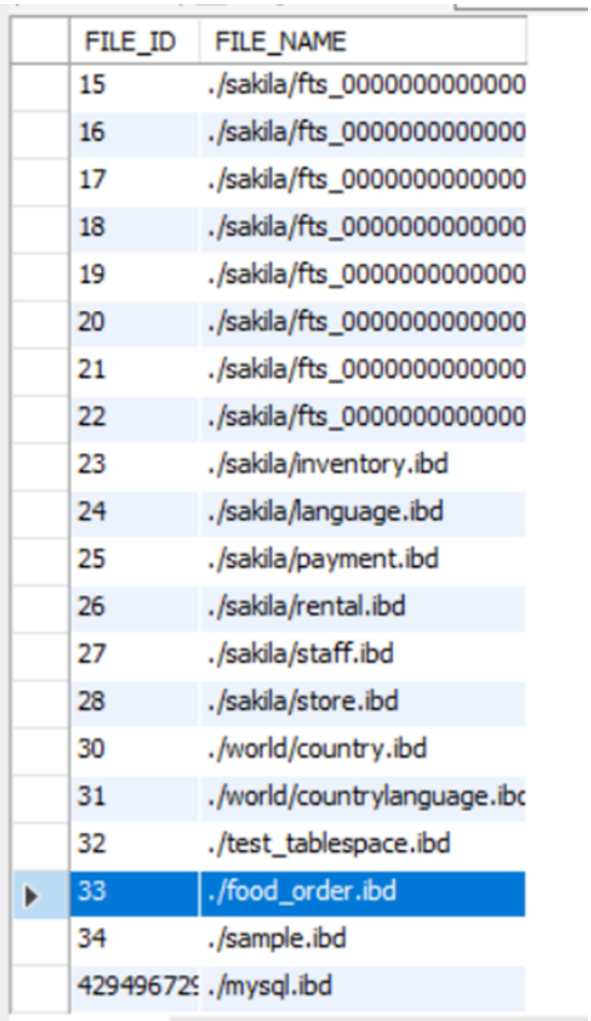
12. Create the tablespace of food_order database.



13. List the tablespaces of MySQL local instance.

```
SELECT FILE_ID, FILE_NAME FROM INFORMATION_SCHEMA.FILES
       WHERE FILE_NAME LIKE '%.ibd%' ORDER BY FILE_ID;
```

Output:



FILE_ID	FILE_NAME
15	./sakila/fts_0000000000000
16	./sakila/fts_0000000000000
17	./sakila/fts_0000000000000
18	./sakila/fts_0000000000000
19	./sakila/fts_0000000000000
20	./sakila/fts_0000000000000
21	./sakila/fts_0000000000000
22	./sakila/fts_0000000000000
23	./sakila/inventory.ibd
24	./sakila/language.ibd
25	./sakila/payment.ibd
26	./sakila/rental.ibd
27	./sakila/staff.ibd
28	./sakila/store.ibd
30	./world/country.ibd
31	./world/countrylanguage.ibd
32	./test_tablespace.ibd
33	./food_order.ibd
34	./sample.ibd
42949672	./mysql.ibd

14. List all users of MySQL local instance.

```
1 •  select user from mysql.user
```

Result Grid | Filter Rows: | Export: | Wra

user
jenish
mysql.infoschema
mysql.session
mysql.sys
root

15. List the memory usage of all schemas in MySQL instances.

```
SELECT s.schema_name,
CONCAT(IFNULL(ROUND((SUM(t.data_length)+SUM(t.index_length))/1024/1024,2),0.00),"Mb") total_size,
CONCAT(IFNULL(ROUND(((SUM(t.data_length)+SUM(t.index_length))-SUM(t.data_free))/1024/1024,2),0.00),"Mb") data_used,
CONCAT(IFNULL(ROUND(SUM(data_free)/1024/1024,2),0.00),"Mb") data_free,
IFNULL(ROUND((((SUM(t.data_length)+SUM(t.index_length))-SUM(t.data_free))/((SUM(t.data_length)+SUM(t.index_length)))*
FROM INFORMATION_SCHEMA.SCHEMATA s, INFORMATION_SCHEMA.TABLES t
WHERE s.schema_name = t.table_schema
```

Output:

	SCHEMA_NAME	total_size	data_used	data_free	pct_used
1	sakila	6.45Mb	6.45Mb	0.00Mb	100.00
2	mysql	2.63Mb	-137.38Mb	140.00Mb	-5233.33
3	world	0.80Mb	0.80Mb	0.00Mb	100.00
4	online_food_order	0.23Mb	0.23Mb	0.00Mb	100.00
5	sys	0.02Mb	0.02Mb	0.00Mb	100.00
6	information_schema	0.00Mb	0.00Mb	0.00Mb	0.00
7	performance_schema	0.00Mb	0.00Mb	0.00Mb	0.00

16. list the memory usage of all tables in MySQL instance.

```sql
SELECT s.schema_name, table_name,
CONCAT(IFNULL(ROUND((SUM(t.data_length)+SUM(t.index_length))/1024/1024,2),0.00),"Mb") total_size,
CONCAT(IFNULL(ROUND(((SUM(t.data_length)+SUM(t.index_length))-SUM(t.data_free))/1024/1024,2),0.00),"Mb") data_used,
CONCAT(IFNULL(ROUND(SUM(data_free)/1024/1024,2),0.00),"Mb") data_free,
IFNULL(ROUND((((SUM(t.data_length)+SUM(t.index_length))-SUM(t.data_free))/((SUM(t.data_length)+SUM(t.index_length)))*100),2),0) pct_used
FROM INFORMATION_SCHEMA.SCHEMATA s, INFORMATION_SCHEMA.TABLES t
```

Output:

	SCHEMA_NAME	TABLE_NAME	total_size	data_used	data_free	pct_used
30	online_food_order	feedback	0.03Mb	0.03Mb	0.00Mb	100.00
31	online_food_order	orders	0.03Mb	0.03Mb	0.00Mb	100.00
32	sakila	actor	0.03Mb	0.03Mb	0.00Mb	100.00
33	online_food_order	company	0.02Mb	0.02Mb	0.00Mb	100.00
34	online_food_order	customer	0.02Mb	0.02Mb	0.00Mb	100.00
35	online_food_order	employee	0.02Mb	0.02Mb	0.00Mb	100.00
36	online_food_order	login	0.02Mb	0.02Mb	0.00Mb	100.00
37	online_food_order	menu	0.02Mb	0.02Mb	0.00Mb	100.00
38	world	category	0.02Mb	0.02Mb	0.00Mb	100.00
39	world	comments	0.02Mb	0.02Mb	0.00Mb	100.00
40	sakila	category	0.02Mb	0.02Mb	0.00Mb	100.00

17. Find out the Redo log file from InnoDB engine.

```sql
SHOW ENGINE INNODB STATUS
```

This command shows all the information of the current running instance.

Output:

```
LOG
---
Log sequence number              30972250
Log buffer assigned up to        30972250
Log buffer completed up to       30972250
Log written up to                30972250
Log flushed up to                30972250
Added dirty pages up to          30972250
Pages flushed up to              30972250
Last checkpoint at               30972250
2569 log i/o's done, 0.00 log i/o's/second
```

```
----------------------
BUFFER POOL AND MEMORY
----------------------
Total large memory allocated 0
Dictionary memory allocated 512459
Buffer pool size   511
Free buffers       233
Database pages     270
Old database pages 0
Modified db pages  0
Pending reads      0
Pending writes: LRU 0, flush list 0, single page 0
Pages made young 0, not young 0
0.00 youngs/s, 0.00 non-youngs/s
Pages read 20548, created 374, written 5810
0.00 reads/s, 0.00 creates/s, 0.00 writes/s
No buffer pool page gets since the last printout
Pages read ahead 0.00/s, evicted without access 0.00/s, Random read ahead 0.00/s
LRU len: 270, unzip_LRU len: 0
I/O sum[2]:cur[0], unzip sum[0]:cur[0]
--------------
```

```
------------------------
LATEST FOREIGN KEY ERROR
------------------------
2022-04-08 16:25:17 0x3370 Transaction:
TRANSACTION 5429, ACTIVE 0 sec updating or deleting, thread declared inside InnoDB 4999
mysql tables in use 1, locked 1
4 lock struct(s), heap size 1128, 2 row lock(s), undo log entries 1
MySQL thread id 11, OS thread handle 13168, query id 669 localhost 127.0.0.1 root updating
DELETE FROM orders WHERE order_id=261
Foreign key constraint fails for table `online_food_order`.`payment`:
,
  CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`order_id`) REFERENCES `orders` (`order_id`)
Trying to delete in parent table, in index PRIMARY tuple:
DATA TUPLE: 9 fields;
 0: len 4; hex 80000105; asc     ;;
 1: len 6; hex 000000001535; asc      5;;
 2: len 7; hex 020000013a07b9; asc     :  ;;
 3: len 4; hex 80000005; asc     ;;
 4: SQL NULL;
 5: SQL NULL;
 6: len 4; hex 50616964; asc Paid;;
 7: len 3; hex 8fcb66; asc   f;;
 8: SQL NULL;

But in child table `online_food_order`.`payment`, in index order_id, there is a record:
PHYSICAL RECORD: n_fields 2; compact format; info bits 0
 0: len 4; hex 80000105; asc     ;;
 1: len 4; hex 800000a6; asc     ;;
```

The pluggable Database commands are following:

1. **Overview of the customer table.**

```
1 SELECT * FROM customer;
```

Output:

	customer_id int	customer_email varchar(255)	customer_password varchar(255)	customer_first_name varchar(255)	customer_last_name varchar(255)	customer_dob varchar(255)	customer_address varchar(255)	customer_city varchar(255)
1	1	aman@gmail.com	test	Aman	Kumar	2021-10-15	gjhg	Mumbai
2	2	sumit@gmail.com	test	Pawan	Kumar	2021-10-15	jhjk	Jaipur
3	3	rahul@gmail.com	test	Rahul	Kumar	2021-10-08	jhg	Mumbai
4	4	sumit@gmail.com	test	Sumit	Kumar	2021-10-16	jk	Delhi
5	5	amit@gmail.com	test	Amit	Kumar	2021-10-26	gg1	Kanpur

2. **Overview of the Comments table.**

Select * from comments;

Output:

	comments_ int	item_id int	customer_id int	comments_title text	comments_date varchar(255)	comments_descriptic text
1	1	1	1	My First Comment	2021-10-27	Multiply sea night grass
2	2	1	2	Multiply sea night grass	2021-10-27	Multiply sea night grass
3	3	1	3	Multiply sea night grass	2021-10-27	Multiply sea night grass
4	4	1	4	Multiply sea night grass	2021-10-27	Multiply sea night grass
5	5	1	5	Multiply sea night grass	2021-10-27	Multiply sea night grass
6	6	1	2	Multiply sea night grass	2021-10-27	Multiply sea night grass
7	7	1	2	Multiply sea night grass	2021-10-27	Multiply sea night grass
8	133	5	1	asasdf	2021-10-27	asdfadsf
9	139	10	5	I have no intention of le	2021-10-27	Unperturbed by the stin
10	199	7	5	asdf	2022-02-27	adfasdf
11	200	7	5	asdf	2022-02-27	asdf
12	279	10	5	Very good product	2022-03-12	This is very good produc

3. **Overview of the Payment table.**

```
1 SELECT * FROM payment LIMIT 100;
```

Output:

	* payment_id int	* order_id int	* customer_id int	* payment_date date	payment_amount float
1	165	5	1	2021-11-06	10000
2	166	5	2	2021-11-06	15000
3	167	5	3	2021-11-06	12000

We Use 3 trigger functions to automate the value updation in our project

1. **Update total values in the payment table:**
   one customer has many orders, we give the customer one order id and many item's id. We get those item id prices from the menu table and in the payment table we count those all values from order id. So we can find the total order amount.

```
begin
set new.payment_amount = (select order_total from orders where order_id=new.order_id);
end
```

2. **Update the item is Paid/Unpaid:** sometimes the user has so many items in his/her cart but they remove something from their cart. We keep the item so we can target the audience in the future. That's why we had the paid/unpaid column(payment status) in the order table. The update payment status trigger just updates the payment status inside the order table and the status of the order.

```
begin
update orders o, payment p set o.order_status="Paid" where p.order_id=o.order_id;
end
```

3. **Order total**: this trigger simply fetches the item price from the menu order.

```
begin

set new.order_total = (select price from menu where menu.item_id=new.item_id);
end
```

4. **List of triggers in the current database**

   Show TRIGGERS;

   Output: