IT2351 - Project

Business Statement

Keeping track of data helps businesses make smart and informed decisions. This database created will keep track of customers, products, toppings , vendors and sales data. This database includes a stored prcodures that will quickly retrieve frequently inquired data. It also includes build in protections like view, where only necessary data is quickly and conveniently available for those who need it. Overall this product wil provide the business with any metrics related to the aforementioned fields to help businesses maximise revenue.

Fields in Database

This database has 5 tables and fields are organized by their respective tables.

1) Customer table

customer_id	first_name	last_name	phone_number

2) Products table

product_id	product_name	product_price	current_product_inventory	vendor_id

3) Toppings table

topping_id	topping_name	topping_price	topping_inventory	vendor_id

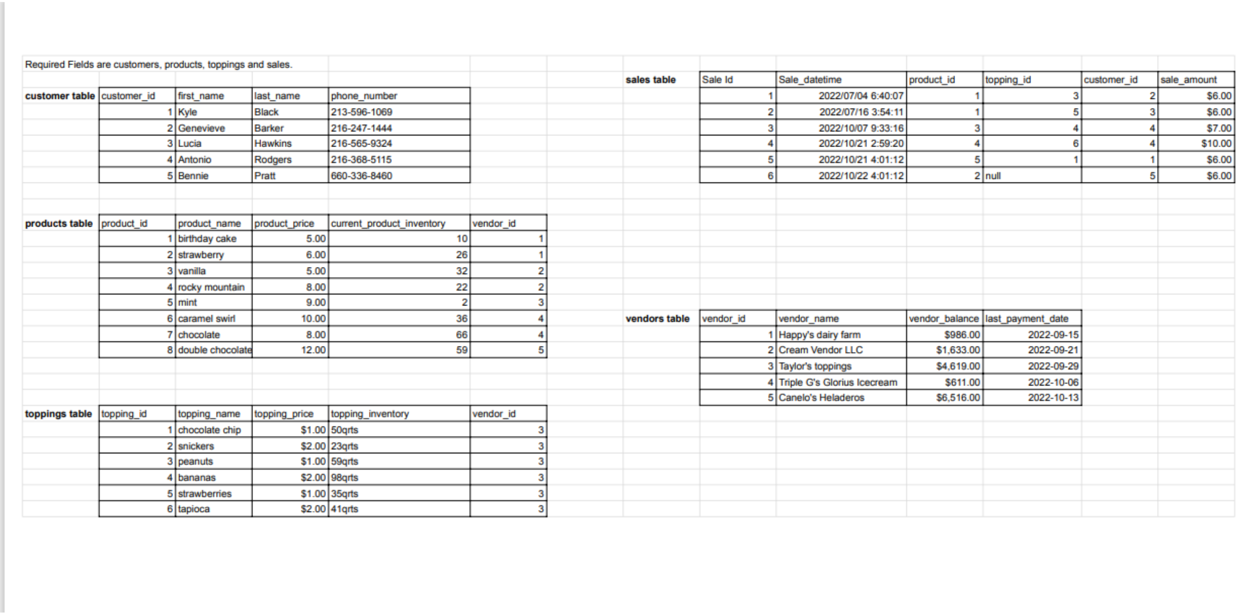
4) Sale table

sale_id	Sale_datetime	product_id	topping_id	customer_id	sale_amount

5) Vendors table

vendor_id	vendor_name	vendor_balance	last_payment_date

Sample data of the tables:-



Required Fields are customers, products, toppings and sales.

customer table

customer_id	first_name	last_name	phone_number
1	Kyle	Black	213-596-1069
2	Genevieve	Barker	216-247-1444
3	Lucia	Hawkins	216-565-9324
4	Antonio	Rodgers	216-368-5115
5	Bennie	Pratt	660-336-8460

products table

product_id	product_name	product_price	current_product_inventory	vendor_id
1	birthday cake	5.00	10	1
2	strawberry	6.00	26	1
3	vanilla	5.00	32	2
4	rocky mountain	8.00	22	2
5	mint	9.00	2	3
6	caramel swirl	10.00	36	4
7	chocolate	8.00	66	4
8	double chocolate	12.00	59	5

toppings table

topping_id	topping_name	topping_price	topping_inventory	vendor_id
1	chocolate chip	$1.00	50qrts	3
2	snickers	$2.00	23qrts	3
3	peanuts	$1.00	59qrts	3
4	bananas	$2.00	98qrts	3
5	strawberries	$1.00	35qrts	3
6	tapioca	$2.00	41qrts	3

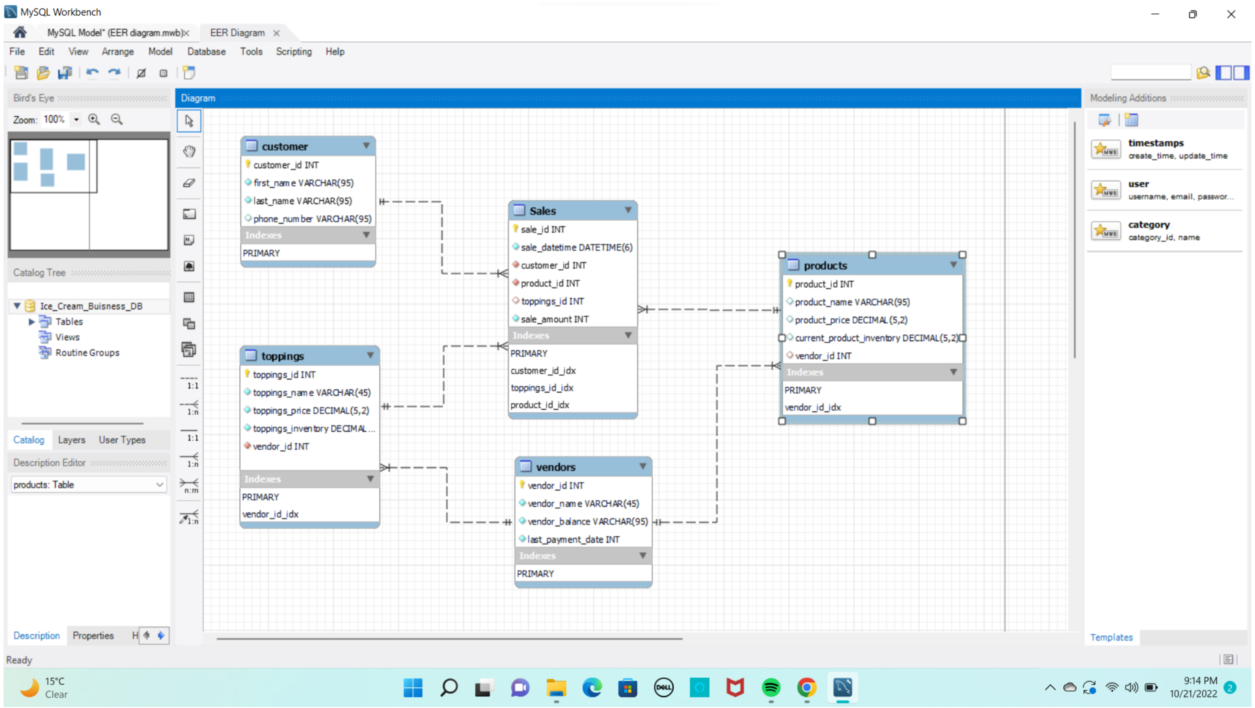
sales table

Sale Id	Sale_datetime	product_id	topping_id	customer_id	sale_amount
1	2022/07/04 6:40:07	1	3	2	$6.00
2	2022/07/16 3:54:11	1	5	3	$6.00
3	2022/10/07 9:33:16	3	4	4	$7.00
4	2022/10/21 2:59:20	4	6	4	$10.00
5	2022/10/21 4:01:12	5	1	1	$6.00
6	2022/10/22 4:01:12	2	null	5	$6.00

vendors table

vendor_id	vendor_name	vendor_balance	last_payment_date
1	Happy's dairy farm	$986.00	2022-09-15
2	Cream Vendor LLC	$1,633.00	2022-09-21
3	Taylor's toppings	$4,619.00	2022-09-29
4	Triple G's Glorius Icecream	$611.00	2022-10-06
5	Canelo's Heladeros	$6,516.00	2022-10-13

EER Diagram



customer
- customer_id INT
- first_name VARCHAR(95)
- last_name VARCHAR(95)
- phone_number VARCHAR(95)

Indexes
- PRIMARY

Sales
- sale_id INT
- sale_datetime DATETIME(6)
- customer_id INT
- product_id INT
- toppings_id INT
- sale_amount INT

Indexes
- PRIMARY
- customer_id_idx
- toppings_id_idx
- product_id_idx

products
- product_id INT
- product_name VARCHAR(95)
- product_price DECIMAL (5,2)
- current_product_inventory DECIMAL(5,2)
- vendor_id INT

Indexes
- PRIMARY
- vendor_id_idx

toppings
- toppings_id INT
- toppings_name VARCHAR(45)
- toppings_price DECIMAL(5,2)
- toppings_inventory DECIMAL...
- vendor_id INT

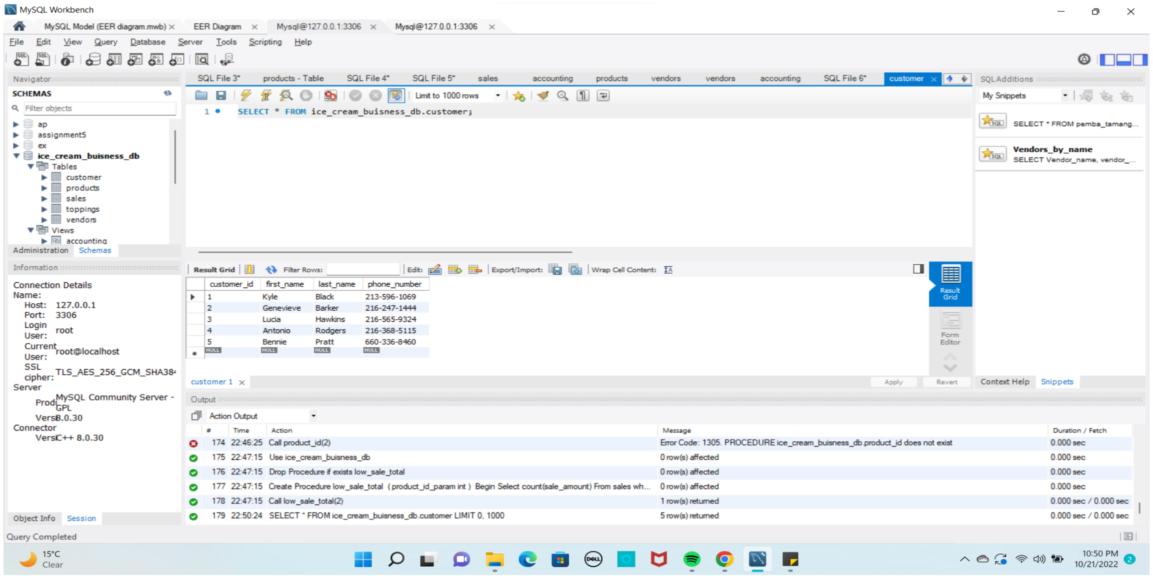
Indexes
- PRIMARY
- vendor_id_idx

vendors
- vendor_id INT
- vendor_name VARCHAR(45)
- vendor_balance VARCHAR(95)
- last_payment_date INT

Indexes
- PRIMARY

Sample Data

The following are sample data from the tables customers, products , sales, toppings and vendors respectively in MySQL.



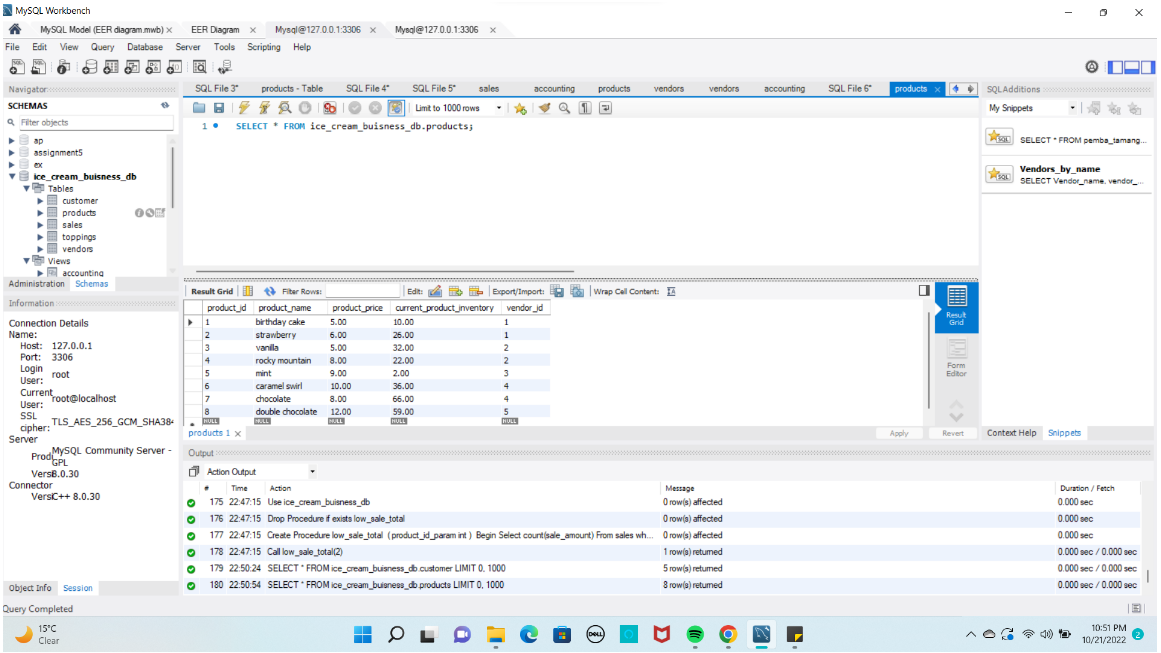
```
SELECT * FROM ice_cream_buisness_db.customer;
```

customer_id	first_name	last_name	phone_number
1	Kyle	Black	213-596-1069
2	Genevieve	Barker	216-247-1444
3	Lucia	Hawkins	216-565-9324
4	Antonio	Rodgers	216-368-5115
5	Bennie	Pratt	660-336-8460
NULL	NULL	NULL	NULL

```
#    Time      Action                                                                                          Message                                                                    Duration / Fetch
174  22:46:25  Call product_id(2)                                                                              Error Code: 1305. PROCEDURE ice_cream_buisness_db.product_id does not exist  0.000 sec
175  22:47:15  Use ice_cream_buisness_db                                                                       0 row(s) affected                                                          0.000 sec
176  22:47:15  Drop Procedure if exists low_sale_total                                                         0 row(s) affected                                                          0.000 sec
177  22:47:15  Create Procedure low_sale_total  ( product_id_param int )  Begin Select count(sale_amount) From sales wh...  0 row(s) affected                                    0.000 sec
178  22:47:15  Call low_sale_total(2)                                                                          1 row(s) returned                                                          0.000 sec / 0.000 sec
179  22:50:24  SELECT * FROM ice_cream_buisness_db.customer LIMIT 0, 1000                                      5 row(s) returned                                                          0.000 sec / 0.000 sec
```

Connection Details
Name:
Host: 127.0.0.1
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA384
Server Product: MySQL Community Server - GPL
Version: 8.0.30
Connector Version: C++ 8.0.30

Query Completed



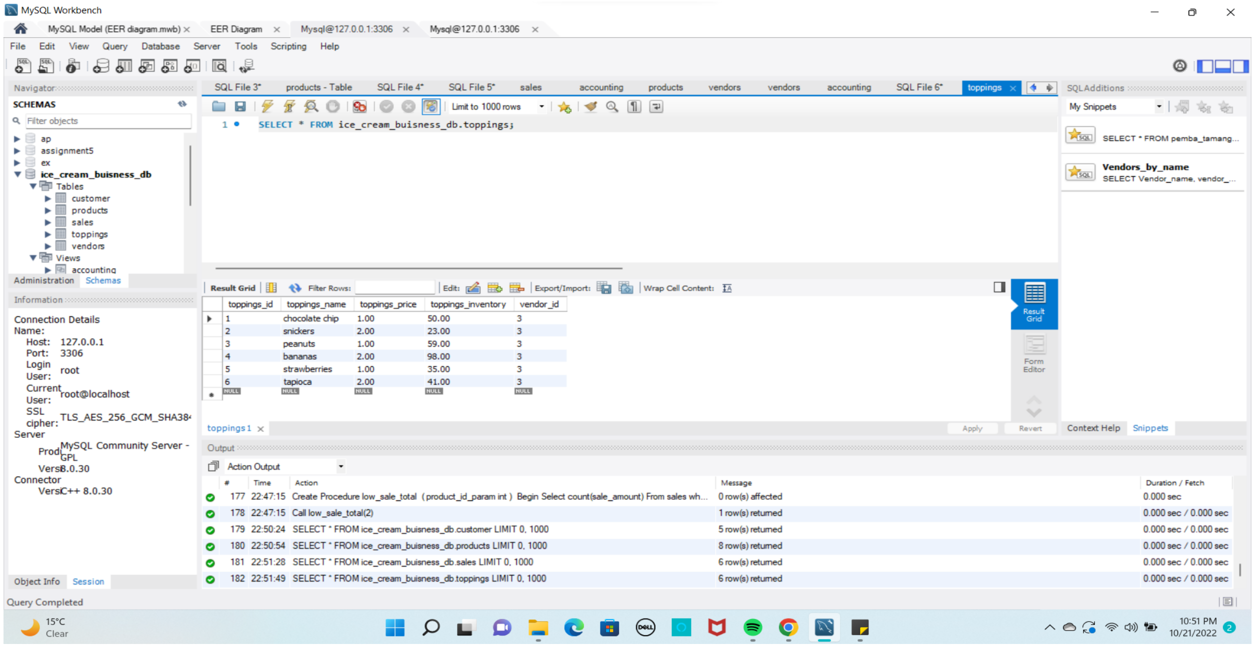
```
SELECT * FROM ice_cream_buisness_db.products;
```

product_id	product_name	product_price	current_product_inventory	vendor_id
1	birthday cake	5.00	10.00	1
2	strawberry	6.00	26.00	1
3	vanilla	5.00	32.00	2
4	rocky mountain	8.00	22.00	2
5	mint	9.00	2.00	3
6	caramel swirl	10.00	36.00	4
7	chocolate	8.00	66.00	4
8	double chocolate	12.00	59.00	5
NULL	NULL	NULL	NULL	NULL

```
#    Time      Action                                                                                          Message            Duration / Fetch
175  22:47:15  Use ice_cream_buisness_db                                                                       0 row(s) affected  0.000 sec
176  22:47:15  Drop Procedure if exists low_sale_total                                                         0 row(s) affected  0.000 sec
177  22:47:15  Create Procedure low_sale_total  ( product_id_param int )  Begin Select count(sale_amount) From sales wh...  0 row(s) affected  0.000 sec
178  22:47:15  Call low_sale_total(2)                                                                          1 row(s) returned  0.000 sec / 0.000 sec
179  22:50:24  SELECT * FROM ice_cream_buisness_db.customer LIMIT 0, 1000                                      5 row(s) returned  0.000 sec / 0.000 sec
180  22:50:54  SELECT * FROM ice_cream_buisness_db.products LIMIT 0, 1000                                      8 row(s) returned  0.000 sec / 0.000 sec
```

Connection Details
Name:
Host: 127.0.0.1
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA384
Server Product: MySQL Community Server - GPL
Version: 8.0.30
Connector Version: C++ 8.0.30

Query Completed



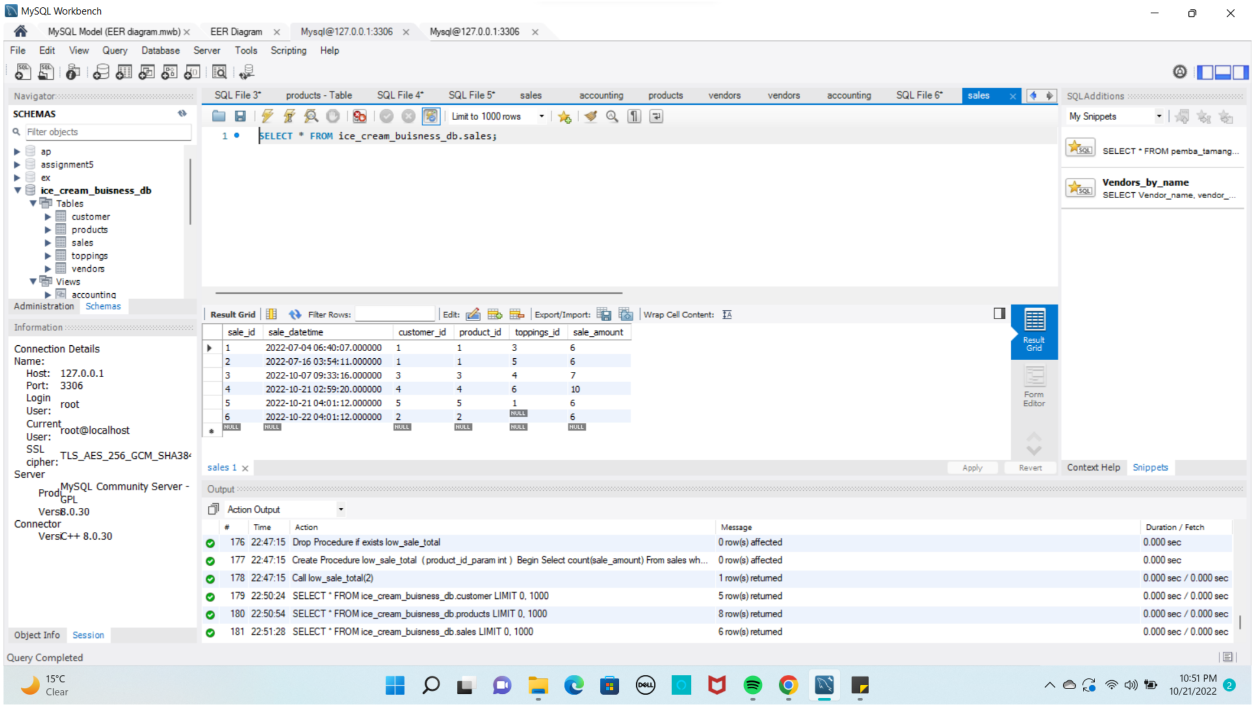
```sql
SELECT * FROM ice_cream_buisness_db.toppings;
```

toppings_id	toppings_name	toppings_price	toppings_inventory	vendor_id
1	chocolate chip	1.00	50.00	3
2	snickers	2.00	23.00	3
3	peanuts	1.00	59.00	3
4	bananas	2.00	98.00	3
5	strawberries	1.00	35.00	3
6	tapioca	2.00	41.00	3
NULL	NULL	NULL	NULL	NULL

```
#    Time      Action                                                                                           Message            Duration / Fetch
177  22:47:15  Create Procedure low_sale_total  ( product_id_param int )  Begin Select count(sale_amount) From sales wh...  0 row(s) affected  0.000 sec
178  22:47:15  Call low_sale_total(2)                                                                           1 row(s) returned  0.000 sec / 0.000 sec
179  22:50:24  SELECT * FROM ice_cream_buisness_db.customer LIMIT 0, 1000                                       5 row(s) returned  0.000 sec / 0.000 sec
180  22:50:54  SELECT * FROM ice_cream_buisness_db.products LIMIT 0, 1000                                       8 row(s) returned  0.000 sec / 0.000 sec
181  22:51:28  SELECT * FROM ice_cream_buisness_db.sales LIMIT 0, 1000                                          6 row(s) returned  0.000 sec / 0.000 sec
182  22:51:49  SELECT * FROM ice_cream_buisness_db.toppings LIMIT 0, 1000                                       6 row(s) returned  0.000 sec / 0.000 sec
```

Connection Details
Name:
Host: 127.0.0.1
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA38
Server
Product: MySQL Community Server - GPL
Version: 8.0.30
Connector
Version: C++ 8.0.30

Query Completed



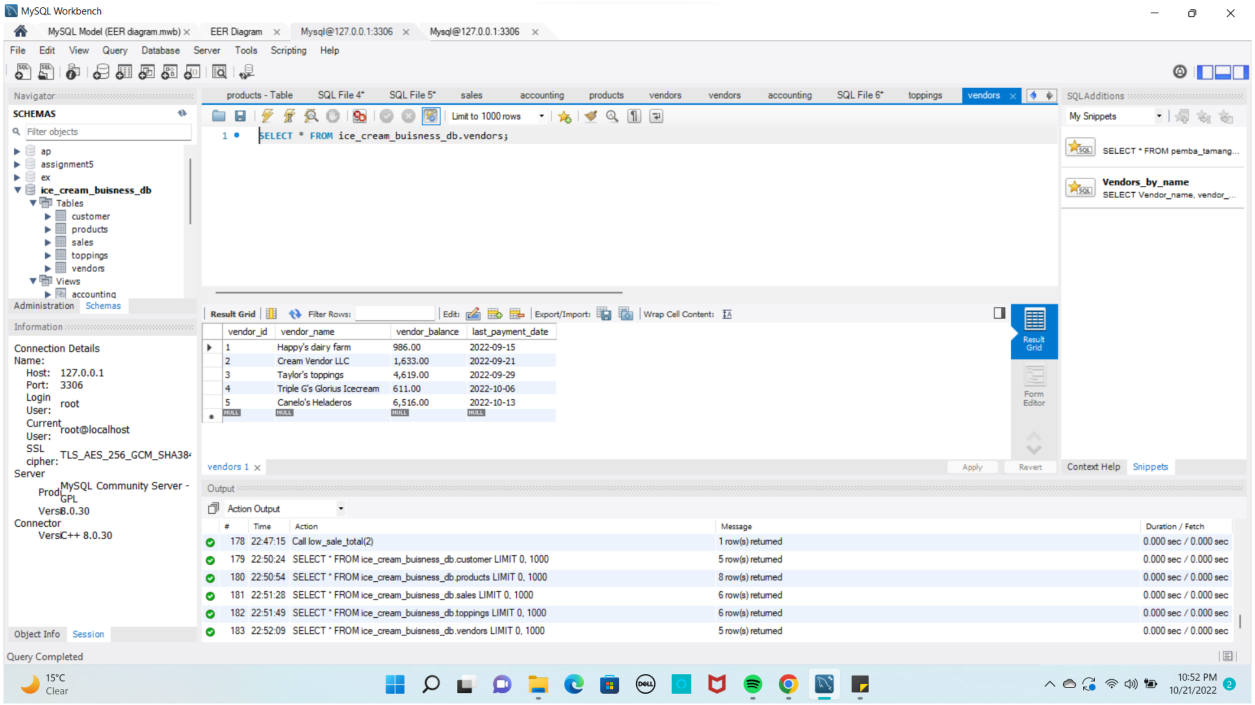
```sql
SELECT * FROM ice_cream_buisness_db.sales;
```

sale_id	sale_datetime	customer_id	product_id	toppings_id	sale_amount
1	2022-07-04 06:40:07.000000	1	1	3	6
2	2022-07-16 03:54:11.000000	1	1	5	6
3	2022-10-07 09:33:16.000000	3	3	4	7
4	2022-10-21 02:59:20.000000	4	4	6	10
5	2022-10-21 04:01:12.000000	5	5	1	6
6	2022-10-22 04:01:12.000000	2	2	NULL	6
NULL	NULL	NULL	NULL	NULL	NULL

```
#    Time      Action                                                                                           Message            Duration / Fetch
176  22:47:15  Drop Procedure if exists low_sale_total                                                          0 row(s) affected  0.000 sec
177  22:47:15  Create Procedure low_sale_total  ( product_id_param int )  Begin Select count(sale_amount) From sales wh...  0 row(s) affected  0.000 sec
178  22:47:15  Call low_sale_total(2)                                                                           1 row(s) returned  0.000 sec / 0.000 sec
179  22:50:24  SELECT * FROM ice_cream_buisness_db.customer LIMIT 0, 1000                                       5 row(s) returned  0.000 sec / 0.000 sec
180  22:50:54  SELECT * FROM ice_cream_buisness_db.products LIMIT 0, 1000                                       8 row(s) returned  0.000 sec / 0.000 sec
181  22:51:28  SELECT * FROM ice_cream_buisness_db.sales LIMIT 0, 1000                                          6 row(s) returned  0.000 sec / 0.000 sec
```

Connection Details
Name:
Host: 127.0.0.1
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA38
Server
Product: MySQL Community Server - GPL
Version: 8.0.30
Connector
Version: C++ 8.0.30

Query Completed



Information

Connection Details
Name:
Host: 127.0.0.1
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_256_GCM_SHA384
Server
Prod MySQL Community Server - GPL
Vers8.0.30
Connector
VersC++ 8.0.30

```sql
SELECT * FROM ice_cream_buisness_db.vendors;
```

vendor_id	vendor_name	vendor_balance	last_payment_date
1	Happy's dairy farm	986.00	2022-09-15
2	Cream Vendor LLC	1,633.00	2022-09-21
3	Taylor's toppings	4,619.00	2022-09-29
4	Triple G's Glorius Icecream	611.00	2022-10-06
5	Canelo's Heladeros	6,516.00	2022-10-13
NULL	NULL	NULL	NULL

```
#    Time      Action                                                      Message            Duration / Fetch
178  22:47:15  Call low_sale_total(2)                                      1 row(s) returned  0.000 sec / 0.000 sec
179  22:50:24  SELECT * FROM ice_cream_buisness_db.customer LIMIT 0, 1000  5 row(s) returned  0.000 sec / 0.000 sec
180  22:50:54  SELECT * FROM ice_cream_buisness_db.products LIMIT 0, 1000  8 row(s) returned  0.000 sec / 0.000 sec
181  22:51:28  SELECT * FROM ice_cream_buisness_db.sales LIMIT 0, 1000     6 row(s) returned  0.000 sec / 0.000 sec
182  22:51:49  SELECT * FROM ice_cream_buisness_db.toppings LIMIT 0, 1000  6 row(s) returned  0.000 sec / 0.000 sec
183  22:52:09  SELECT * FROM ice_cream_buisness_db.vendors LIMIT 0, 1000   5 row(s) returned  0.000 sec / 0.000 sec
```

Description of View

A database has a view feature that provides just the necessary data needed to pay the suppliers. It shows the fields vendor_id , vendor_name , vendor_balance , current_product_inventory of each individual product that comes from their respective vendor . This will conviniently provide jake from accounting identify each vendor, the balance the shop has with each vendor, when the vendor was last payed and the current supply of products each vendor supplies. This information is crucial to maintain the flow of buisness . This information can help Jake pay vendors make smart decisions on which vendor to prioritize if needed.

Description of Stored Procedure

Triggers and stored procedures are extremely useful tool for a business. Triggers can save a lot of repetitive work and provide valuable information. For example a trigger that automatically deducts the amount of unit sold, updates the inventory and shows the remaimaing units left in inventory with every sale can be provide very useful information in keeping track of inventory and making sure all products are available. A stored procedure in a similar way, can save a lot of time and effort by saving a script that yields on-demand information. For example, this database has a stored procedure named sale_total. The user can call this procedure to know the total sale count of any product. This information can be useful in determining which product brings in most revenue.

Database Factors to be Addressed

There are three main factors that needs to be addressed by a Database Administrator. They are as follows :-

Access - The level of access and user priviledge must be controlled vey well. This database contains extremely sensitive buisness information. To ensure appropriate access , there must be built in user access privilege system and a clear understanding among staff who can view and who can edit information in the database.

Exception Handling - Errors and exceptions are bound to happen. A safe way to handle exceptions must be built into the database to avoid a complete crash of the database and subsequent interruption in the business. Concorruency issues might be a common issue , as the changes in inventory and sale in multiple registers would be constant. Systems like locking and queuing should be considered , so that we can simultaneously uphold database integrity and have the buisness encounter as less delays as possible.

Creating Backup - The business relies on the database for essential information, crucial databases always need to have a back-up incase the current one is somehow corrupted, hacked or destroyed.

Conclusion

As the buisness grows The database could include more fields such as type of payment used, or have a stratified class of "special customers" who frequent the store a lot and are entitled to discounts on thier 10th visit or something similar. However increasing database also means the increasingly high resources it uses. Maintaining a database can also prove to be a huge drain on resources. As the amount and complexity of data increases , so will the task of maintaining the database. Storing the database will become increasingly impractical, using a data warehouse should be considered as the data grows. Cloud services such as Google 's big query and AWS database services might prove to be more efficient. Another concern to be noted is that of cybersequirity a strong antivirus program and a employee seminar on threats such as SQL injections might be helpful and go a long wau in ensuring sequirity.