Task 2
MIP-DA-05 Batch

UNCOVER INSIGHTS FROM REAL SALES DATA

MS. MANSI S JADHAV

Agenda

Introduction

Import Data

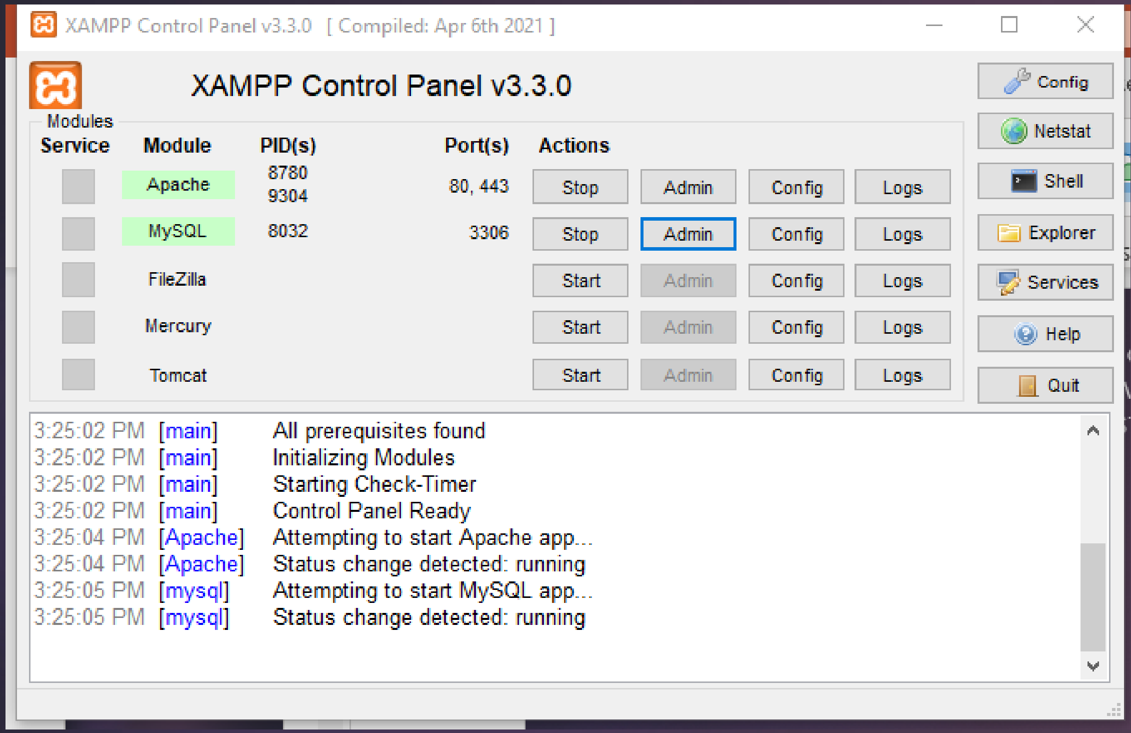
Analysis using SQl Query

Conclusion

Introduction

The database consists of several tables of a delivery service. The "Customers" table holds information regarding the customers, including their unique identifiers (cust_id), names, contact details, demographics such as gender, date of birth and their joining dates. "Products," details the inventory available for delivery, featuring attributes like product name, brand, category, procurement cost per unit, and maximum retail price (mrp). Pincode-related data, such as cities and states, is stored in the "Pincode" table, facilitating location-based services. The "Delivery Person" table tracks information about the delivery person, containing their unique IDs (delivery_person_id), names, joining dates, and assigned pincode areas."Orders" table consolidates order-specific details, including order IDs, dates, quantities, pricing information, product and customer associations, payment types, and delivery pincode destinations. These tables collectively provide a comprehensive framework for managing and tracking various aspects of the delivery service, from customer interactions to product inventory and logistics management.

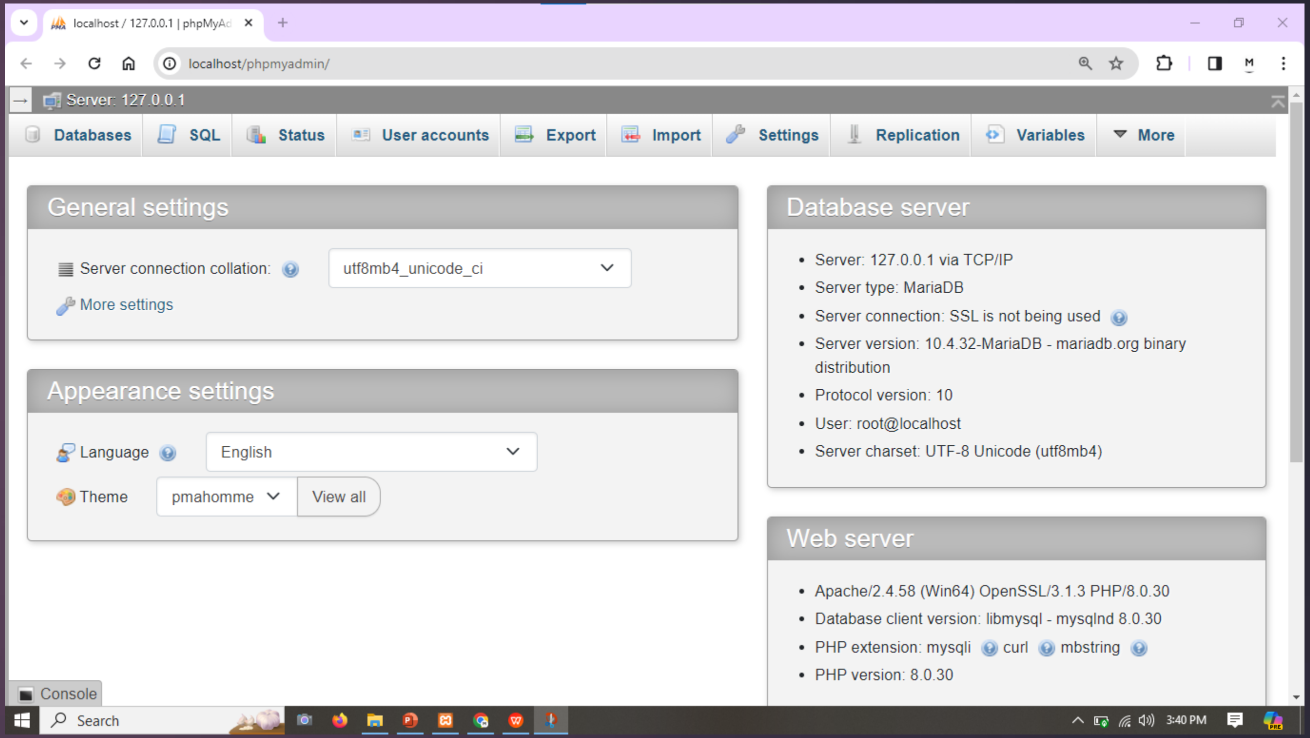
I have use Xampp Server to do analysis of sales_database. So first we will open Xampp Server and then we will start Apache and Mysql and Click on Mysql Admin and then localhost will get open.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions			
	Apache	8780 9304	80, 443	Stop	Admin	Config	Logs
	MySQL	8032	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
3:25:02 PM  [main]    All prerequisites found
3:25:02 PM  [main]    Initializing Modules
3:25:02 PM  [main]    Starting Check-Timer
3:25:02 PM  [main]    Control Panel Ready
3:25:04 PM  [Apache]  Attempting to start Apache app...
3:25:04 PM  [Apache]  Status change detected: running
3:25:05 PM  [mysql]   Attempting to start MySQL app...
3:25:05 PM  [mysql]   Status change detected: running
```



General settings

Server connection collation: utf8mb4_unicode_ci

More settings

Appearance settings

Language: English

Theme: pmahomme View all

Database server

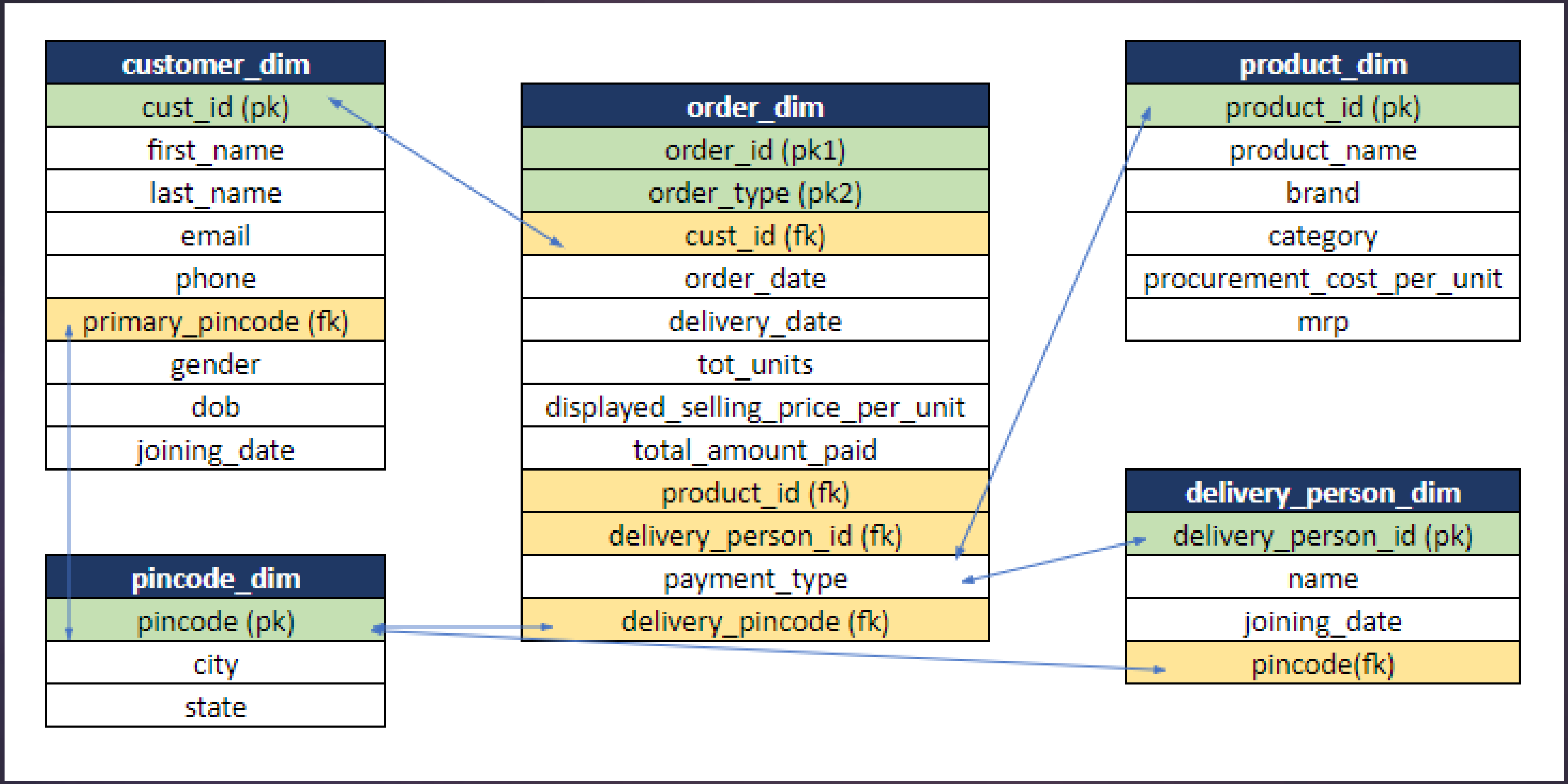
- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.32-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30
- Database client version: libmysql - mysqlnd 8.0.30
- PHP extension: mysqli curl mbstring
- PHP version: 8.0.30

3:40 PM

Structure of the Database



customer_dim
cust_id (pk)
first_name
last_name
email
phone
primary_pincode (fk)
gender
dob
joining_date

order_dim
order_id (pk1)
order_type (pk2)
cust_id (fk)
order_date
delivery_date
tot_units
displayed_selling_price_per_unit
total_amount_paid
product_id (fk)
delivery_person_id (fk)
payment_type
delivery_pincode (fk)

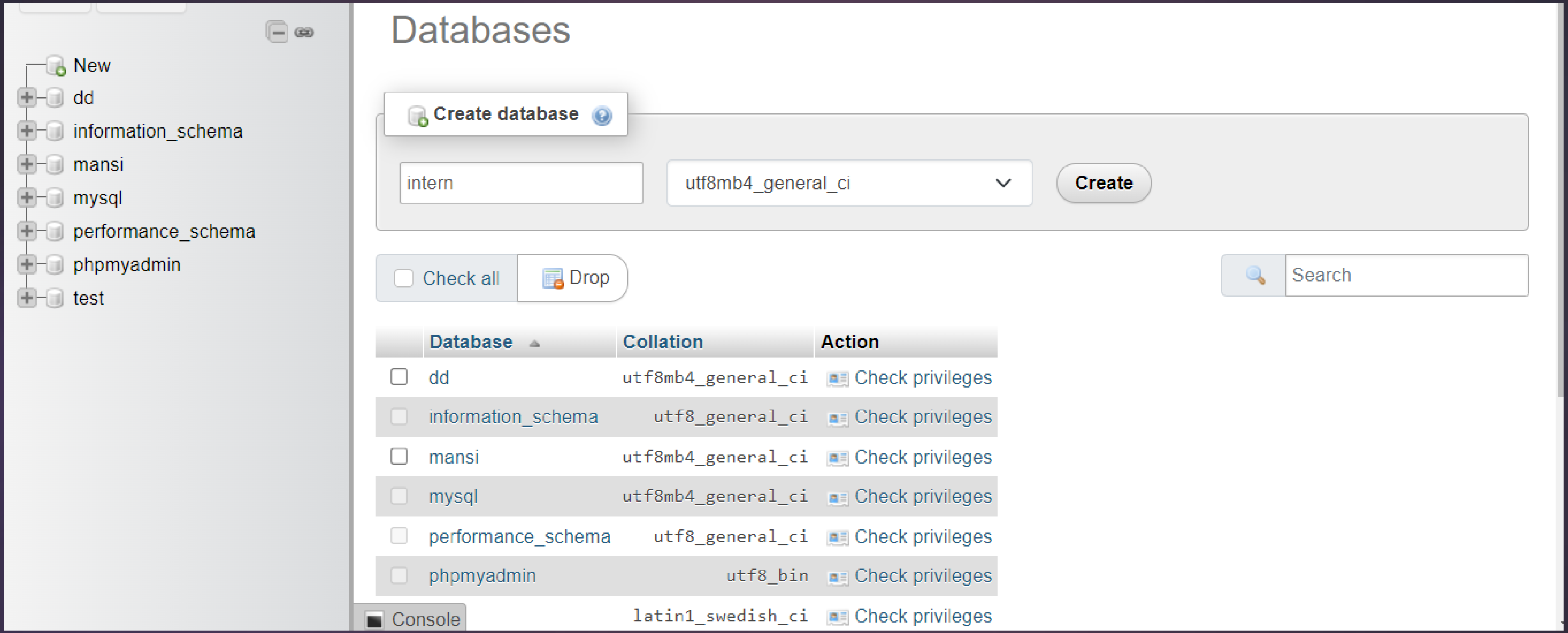
product_dim
product_id (pk)
product_name
brand
category
procurement_cost_per_unit
mrp

pincode_dim
pincode (pk)
city
state

delivery_person_dim
delivery_person_id (pk)
name
joining_date
pincode(fk)

Create Database

First we will create database and then import tables in it one by on.



New
dd
information_schema
mansi
mysql
performance_schema
phpmyadmin
test

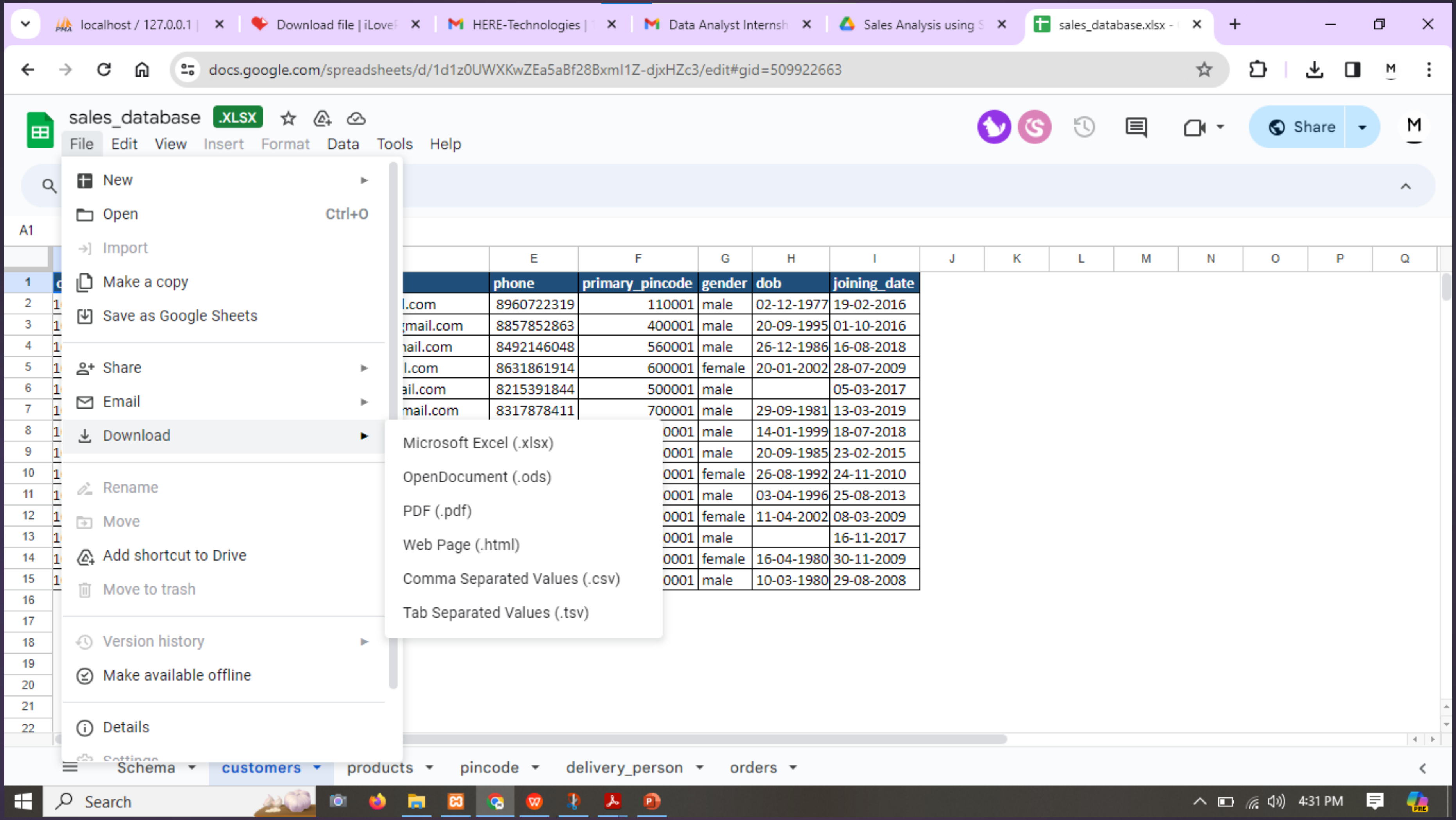
Databases

Create database

intern | utf8mb4_general_ci | Create

	Database	Collation	Action
☐	dd	utf8mb4_general_ci	Check privileges
☐	information_schema	utf8_general_ci	Check privileges
☐	mansi	utf8mb4_general_ci	Check privileges
☐	mysql	utf8mb4_general_ci	Check privileges
☐	performance_schema	utf8_general_ci	Check privileges
☐	phpmyadmin	utf8_bin	Check privileges
		latin1_swedish_ci	Check privileges

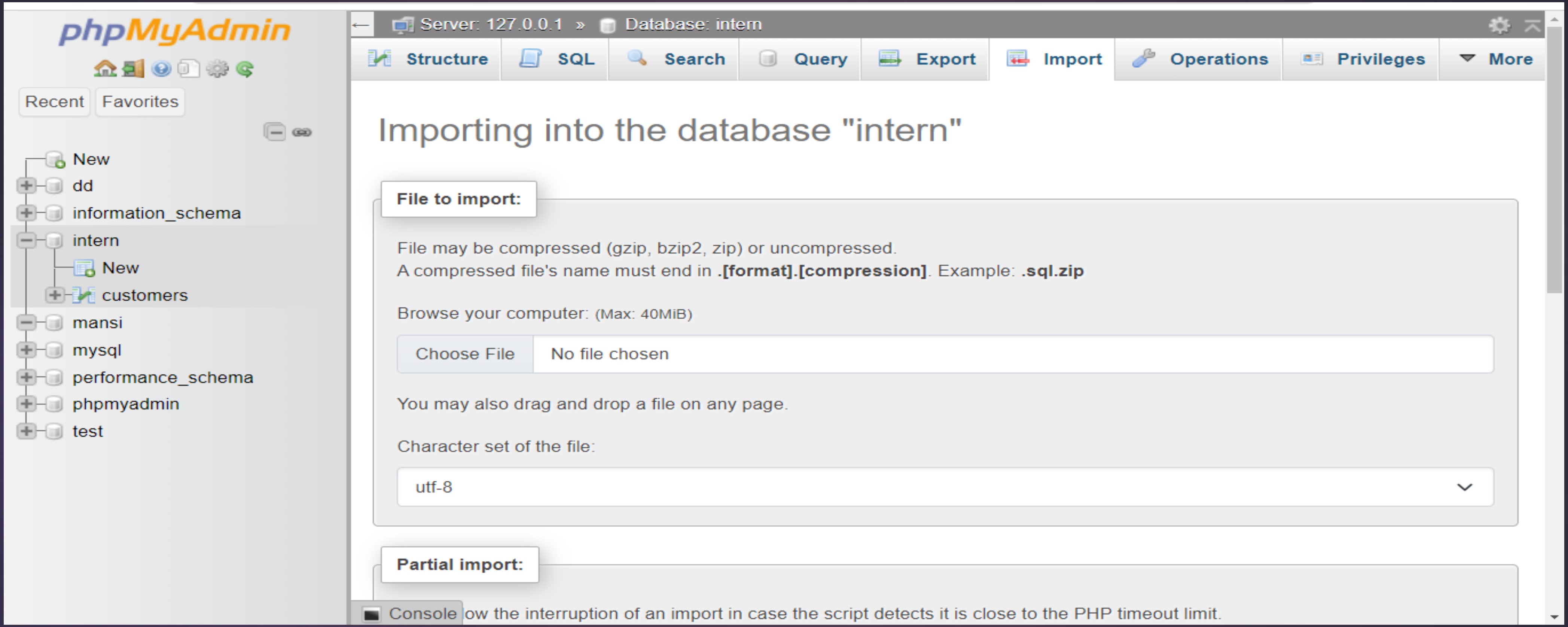
Convert all the sheets of .xlsv into .csv one by one. In the same way convert all the sheets.



	phone	primary_pincode	gender	dob	joining_date
l.com	8960722319	110001	male	02-12-1977	19-02-2016
mail.com	8857852863	400001	male	20-09-1995	01-10-2016
ail.com	8492146048	560001	male	26-12-1986	16-08-2018
l.com	8631861914	600001	female	20-01-2002	28-07-2009
ail.com	8215391844	500001	male		05-03-2017
mail.com	8317878411	700001	male	29-09-1981	13-03-2019
		0001	male	14-01-1999	18-07-2018
		0001	male	20-09-1985	23-02-2015
		0001	female	26-08-1992	24-11-2010
		0001	male	03-04-1996	25-08-2013
		0001	female	11-04-2002	08-03-2009
		0001	male		16-11-2017
		0001	female	16-04-1980	30-11-2009
		0001	male	10-03-1980	29-08-2008

Import Data to SQL Server

So first we will import 1 table



Importing into the database "intern"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: (Max: 40MiB)

No file chosen

You may also drag and drop a file on any page.

Character set of the file:

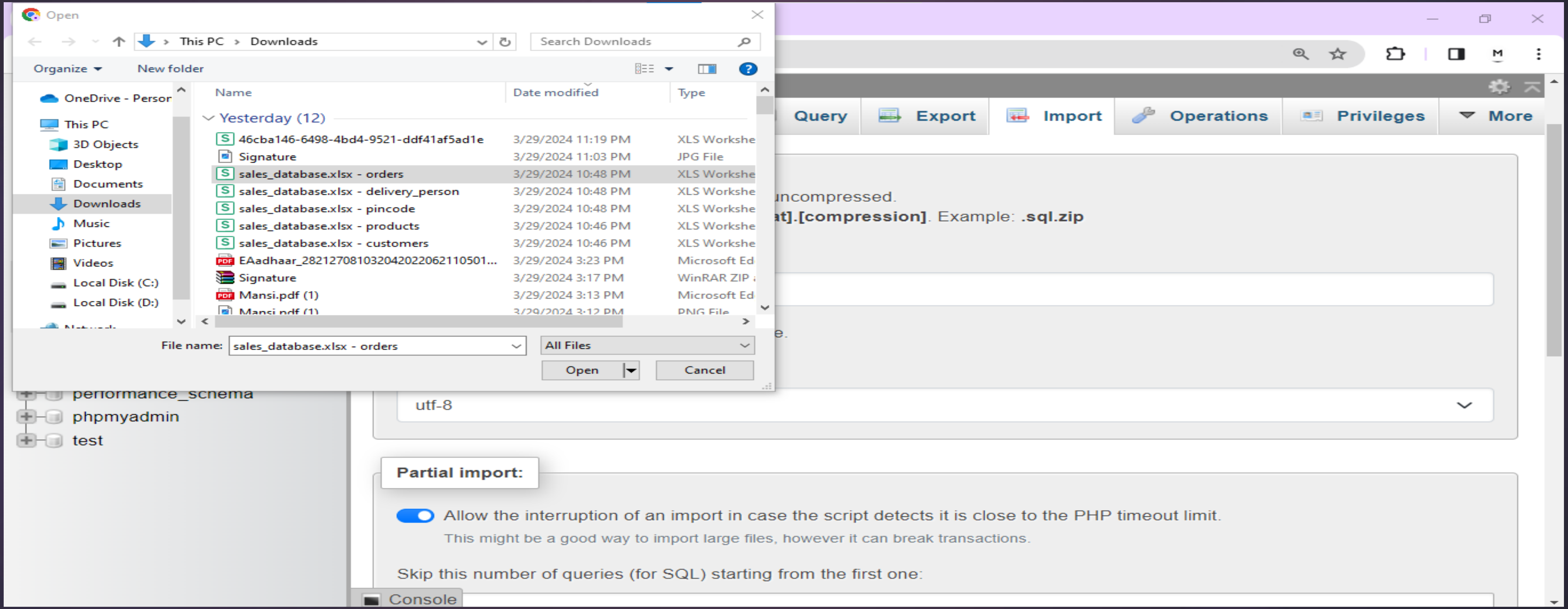
utf-8

Partial import:

ow the interruption of an import in case the script detects it is close to the PHP timeout limit.

Uploading csv file

I have impot orders table



Open

This PC > Downloads

Search Downloads

Organize ▾ New folder

Name	Date modified	Type
Yesterday (12)		
46cba146-6498-4bd4-9521-ddf41af5ad1e	3/29/2024 11:19 PM	XLS Workshe
Signature	3/29/2024 11:03 PM	JPG File
sales_database.xlsx - orders	3/29/2024 10:48 PM	XLS Workshe
sales_database.xlsx - delivery_person	3/29/2024 10:48 PM	XLS Workshe
sales_database.xlsx - pincode	3/29/2024 10:48 PM	XLS Workshe
sales_database.xlsx - products	3/29/2024 10:46 PM	XLS Workshe
sales_database.xlsx - customers	3/29/2024 10:46 PM	XLS Workshe
EAadhaar_282127081032042022062110501...	3/29/2024 3:23 PM	Microsoft Ed
Signature	3/29/2024 3:17 PM	WinRAR ZIP
Mansi.pdf (1)	3/29/2024 3:13 PM	Microsoft Ed
Mansi.pdf (1)	3/29/2024 3:12 PM	PNG File

File name: sales_database.xlsx - orders

All Files

Open Cancel

Query Export Import Operations Privileges More

uncompressed.

at].[compression]. Example: .sql.zip

utf-8

performance_schema

phpmyadmin

test

Partial import:

Allow the interruption of an import in case the script detects it is close to the PHP timeout limit.

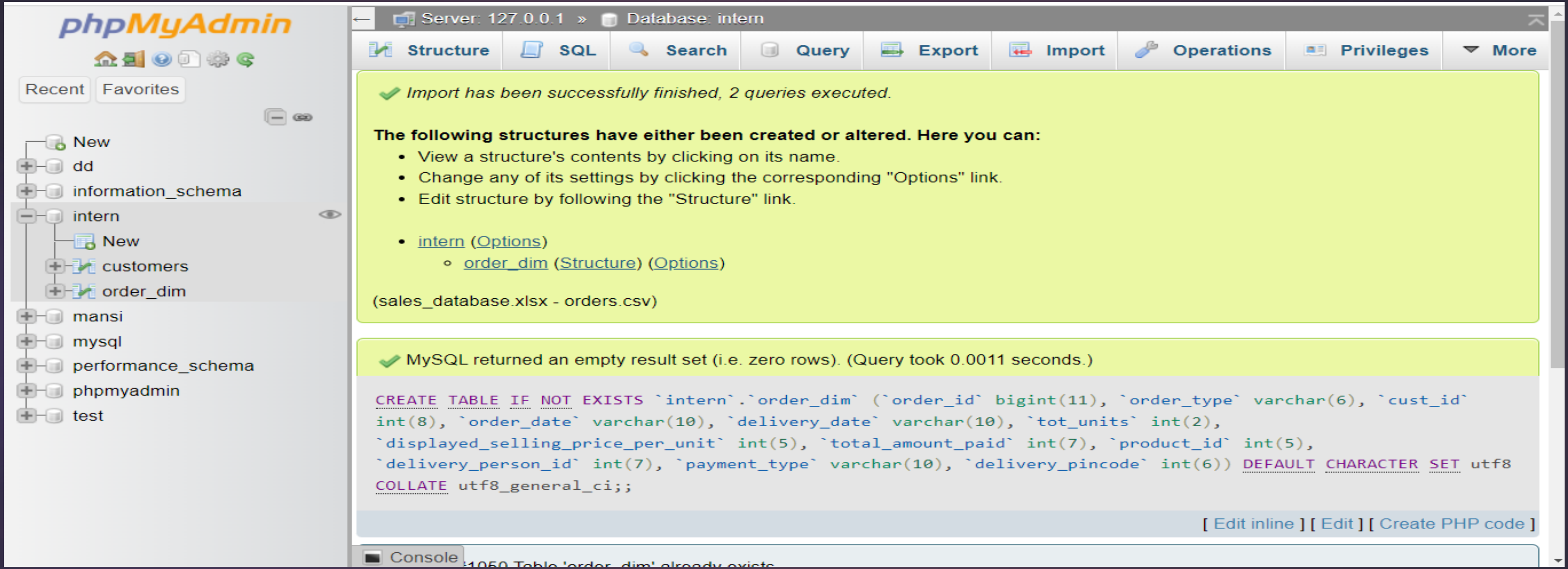
This might be a good way to import large files, however it can break transactions.

Skip this number of queries (for SQL) starting from the first one:

Console

Table added successfully

In the same way import all other tables



Import has been successfully finished, 2 queries executed.

The following structures have either been created or altered. Here you can:

- View a structure's contents by clicking on its name.
- Change any of its settings by clicking the corresponding "Options" link.
- Edit structure by following the "Structure" link.

- intern (Options)
  - order_dim (Structure) (Options)

(sales_database.xlsx - orders.csv)

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0011 seconds.)

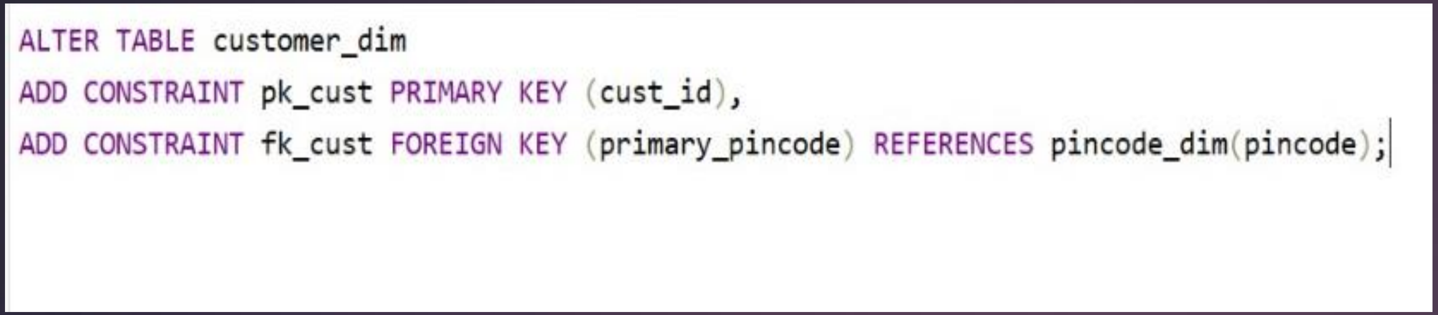
```
CREATE TABLE IF NOT EXISTS `intern`.`order_dim` (`order_id` bigint(11), `order_type` varchar(6), `cust_id`
int(8), `order_date` varchar(10), `delivery_date` varchar(10), `tot_units` int(2),
`displayed_selling_price_per_unit` int(5), `total_amount_paid` int(7), `product_id` int(5),
`delivery_person_id` int(7), `payment_type` varchar(10), `delivery_pincode` int(6)) DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci;;
```

[Edit inline] [Edit] [Create PHP code]

#1050 Table 'order_dim' already exists

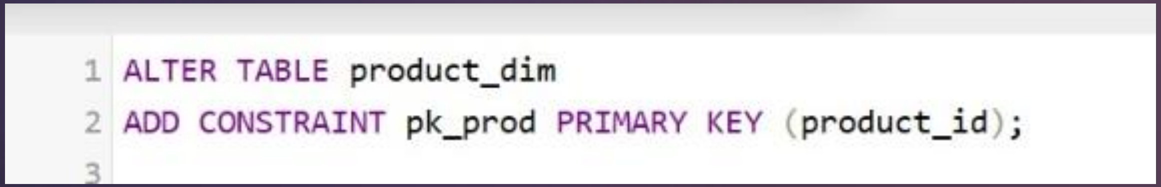
We will add constraints to the table

Customer Table



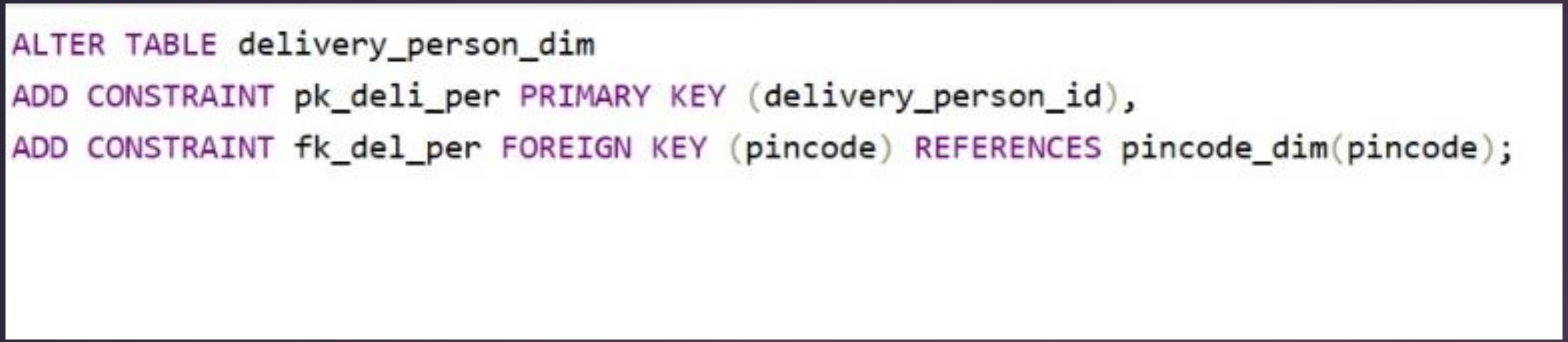
```sql
ALTER TABLE customer_dim
ADD CONSTRAINT pk_cust PRIMARY KEY (cust_id),
ADD CONSTRAINT fk_cust FOREIGN KEY (primary_pincode) REFERENCES pincode_dim(pincode);
```

Product Table



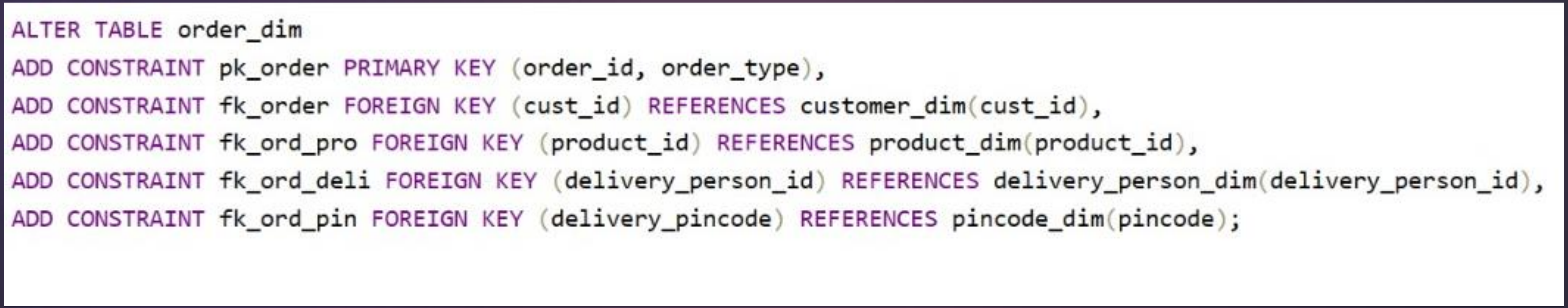
```sql
ALTER TABLE product_dim
ADD CONSTRAINT pk_prod PRIMARY KEY (product_id);
```

For delivery person



```
ALTER TABLE delivery_person_dim
ADD CONSTRAINT pk_deli_per PRIMARY KEY (delivery_person_id),
ADD CONSTRAINT fk_del_per FOREIGN KEY (pincode) REFERENCES pincode_dim(pincode);
```

Order Table

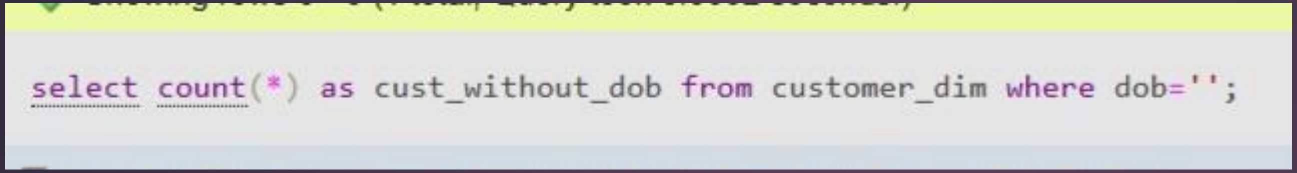


```
ALTER TABLE order_dim
ADD CONSTRAINT pk_order PRIMARY KEY (order_id, order_type),
ADD CONSTRAINT fk_order FOREIGN KEY (cust_id) REFERENCES customer_dim(cust_id),
ADD CONSTRAINT fk_ord_pro FOREIGN KEY (product_id) REFERENCES product_dim(product_id),
ADD CONSTRAINT fk_ord_deli FOREIGN KEY (delivery_person_id) REFERENCES delivery_person_dim(delivery_person_id),
ADD CONSTRAINT fk_ord_pin FOREIGN KEY (delivery_pincode) REFERENCES pincode_dim(pincode);
```

Now we will do the analysis on the data by solving some questions

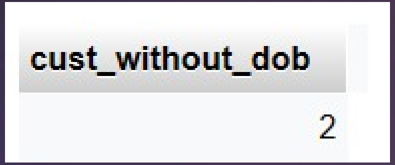
Q1 How many customers do not have DOB information available ?

Query :-



```
select count(*) as cust_without_dob from customer_dim where dob='';
```

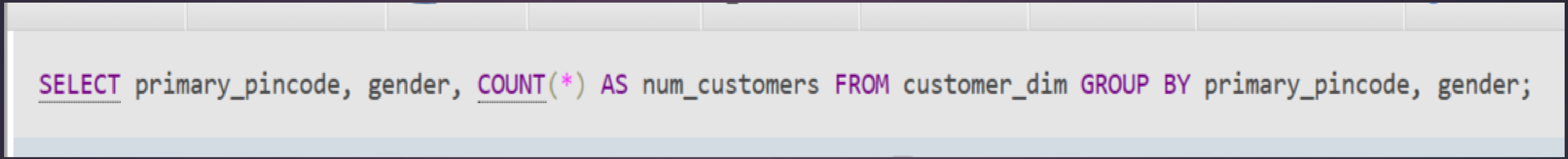
Output :-



cust_without_dob
2

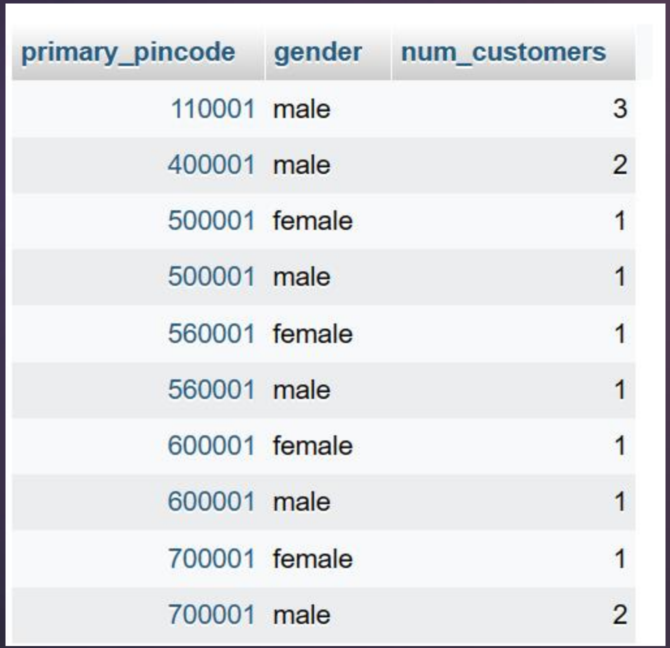
Q2 How many customers are there in each pincode and gender combination?

Query :-



```
SELECT primary_pincode, gender, COUNT(*) AS num_customers FROM customer_dim GROUP BY primary_pincode, gender;
```

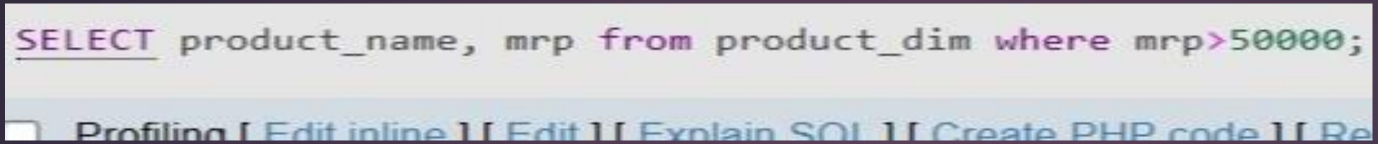
Output :-



primary_pincode	gender	num_customers
110001	male	3
400001	male	2
500001	female	1
500001	male	1
560001	female	1
560001	male	1
600001	female	1
600001	male	1
700001	female	1
700001	male	2

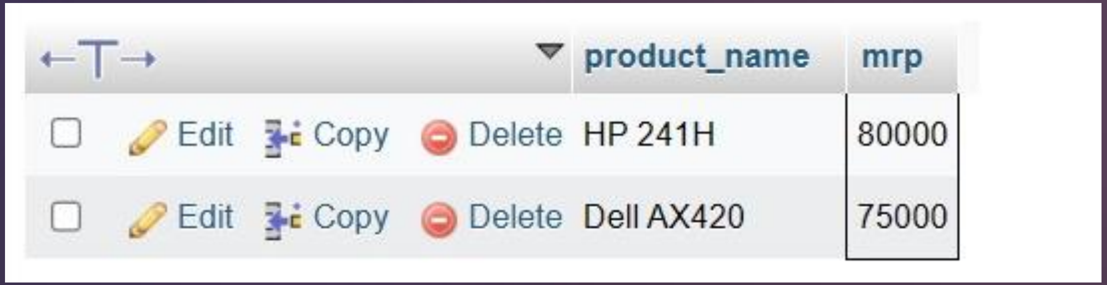
Q 3 Print product name and mrp for products which have more than 50000 MRP?

Query :-



```
SELECT product_name, mrp from product_dim where mrp>50000;
```

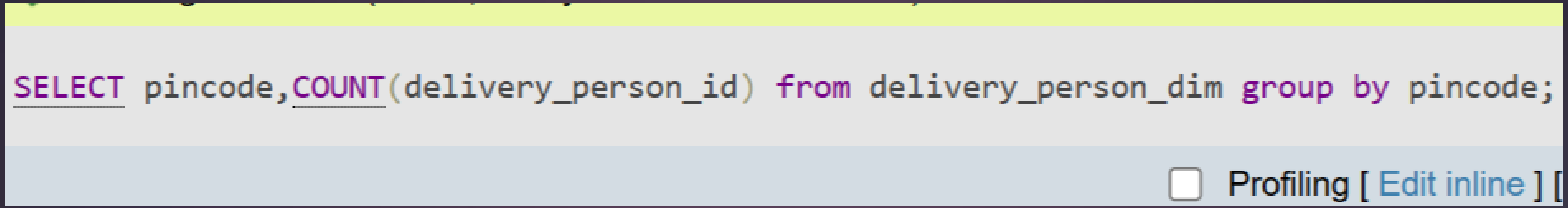
Output :-



	product_name	mrp
Edit Copy Delete	HP 241H	80000
Edit Copy Delete	Dell AX420	75000

Q4 How many delivery person are there in each pincode?

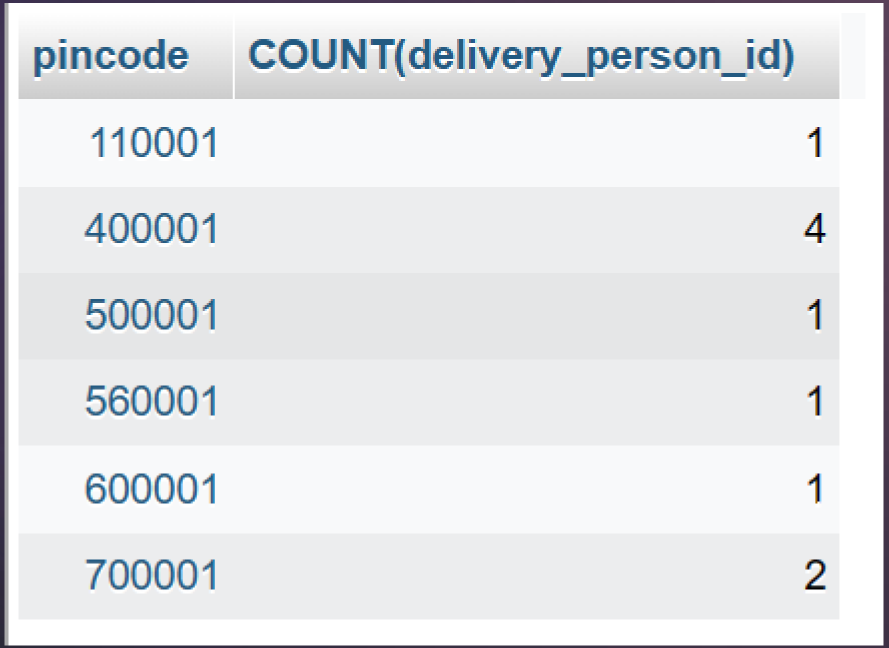
Query :-



```
SELECT pincode,COUNT(delivery_person_id) from delivery_person_dim group by pincode;
```

Profiling [Edit inline] [

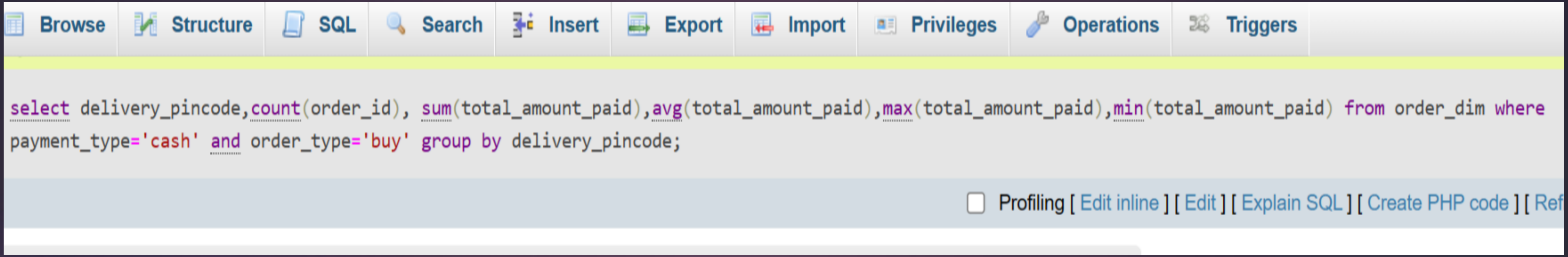
Output :-



pincode	COUNT(delivery_person_id)
110001	1
400001	4
500001	1
560001	1
600001	1
700001	2

Q5 For each Pin code, print the count of orders, sum of total amount paid, average amount paid, maximum amount paid, minimum amount paid for the transactions which were paid by 'cash'. Take only 'buy' order types

Query :-

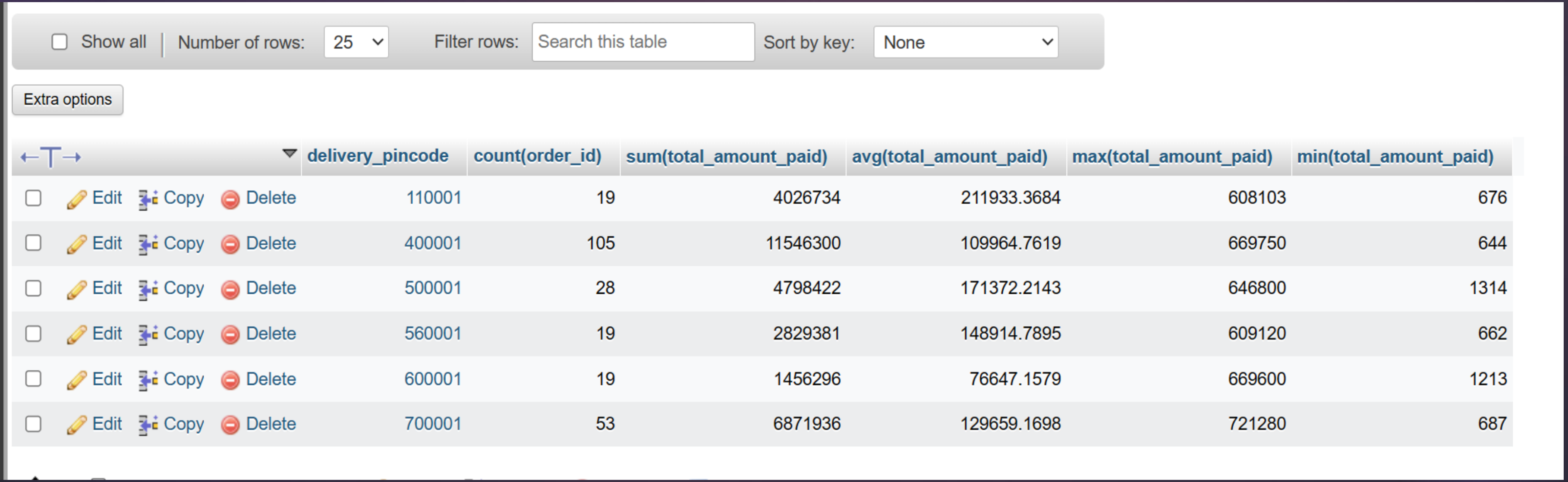


Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

```
select delivery_pincode,count(order_id), sum(total_amount_paid),avg(total_amount_paid),max(total_amount_paid),min(total_amount_paid) from order_dim where
payment_type='cash' and order_type='buy' group by delivery_pincode;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Ref

Output :-



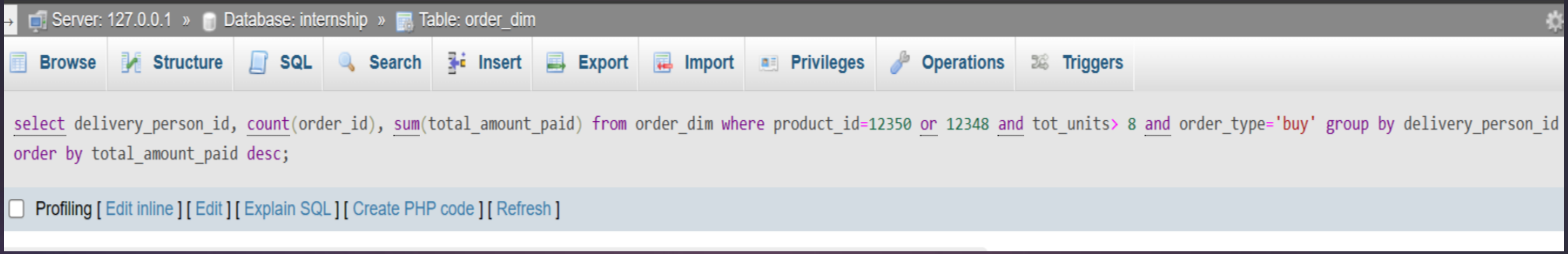
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	delivery_pincode	count(order_id)	sum(total_amount_paid)	avg(total_amount_paid)	max(total_amount_paid)	min(total_amount_paid)
Edit Copy Delete	110001	19	4026734	211933.3684	608103	676
Edit Copy Delete	400001	105	11546300	109964.7619	669750	644
Edit Copy Delete	500001	28	4798422	171372.2143	646800	1314
Edit Copy Delete	560001	19	2829381	148914.7895	609120	662
Edit Copy Delete	600001	19	1456296	76647.1579	669600	1213
Edit Copy Delete	700001	53	6871936	129659.1698	721280	687

Q6 For each delivery_person_id, print the count of orders and total amount paid for product_id = 12350 or 12348 and total units > 8. Sort the output by total amount paid in descending order. Take only 'buy' order types

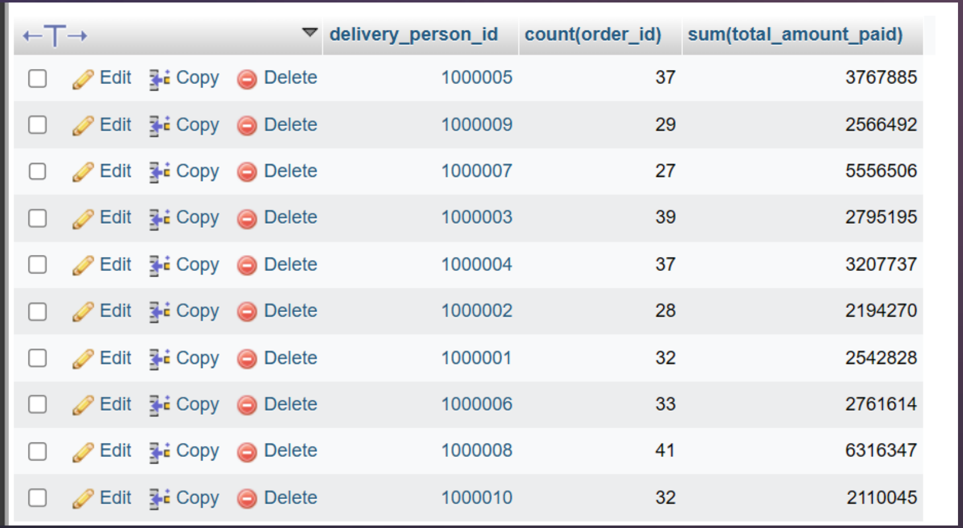
Query :-


Server: 127.0.0.1 » Database: internship » Table: order_dim

```
select delivery_person_id, count(order_id), sum(total_amount_paid) from order_dim where product_id=12350 or 12348 and tot_units> 8 and order_type='buy' group by delivery_person_id
order by total_amount_paid desc;
```

Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

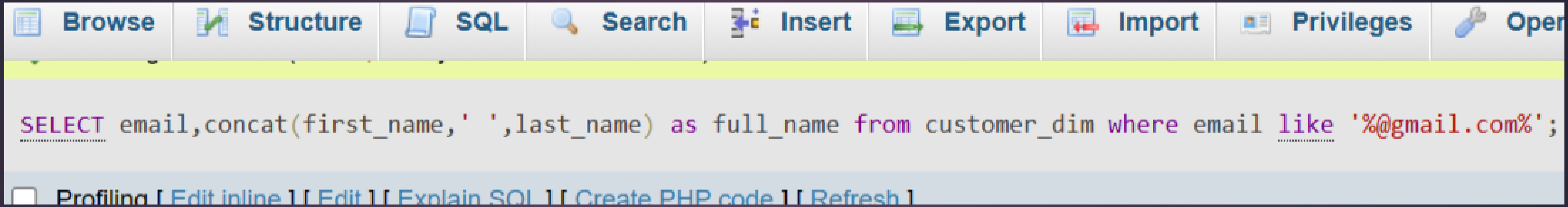
Output :-



delivery_person_id	count(order_id)	sum(total_amount_paid)
1000005	37	3767885
1000009	29	2566492
1000007	27	5556506
1000003	39	2795195
1000004	37	3207737
1000002	28	2194270
1000001	32	2542828
1000006	33	2761614
1000008	41	6316347
1000010	32	2110045

Q7 Print the Full names (first name plus last name) for customers that have email on "gmail.com"?

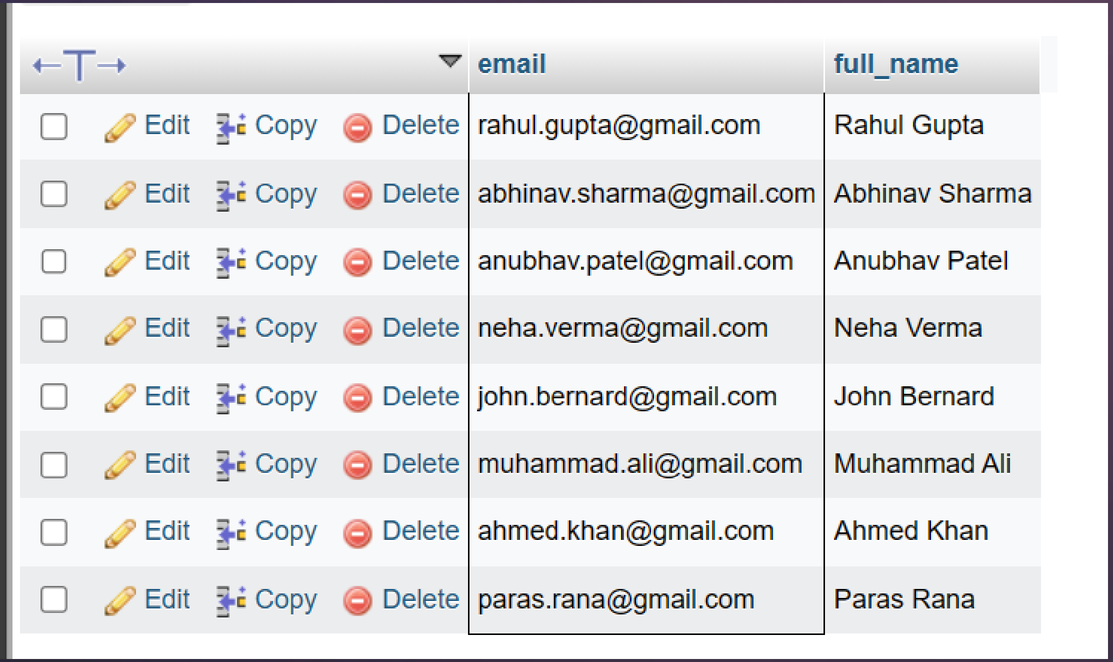
Query :-



```
SELECT email,concat(first_name,' ',last_name) as full_name from customer_dim where email like '%@gmail.com%';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

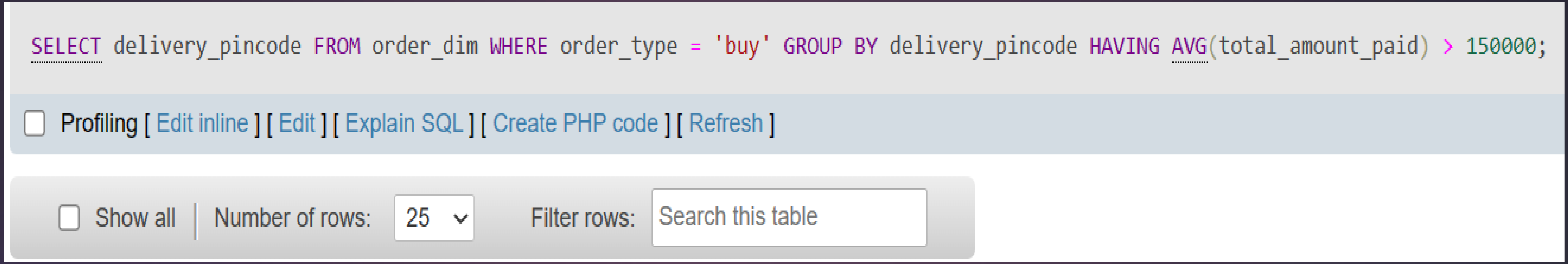
Output :-



	email	full_name
Edit Copy Delete	rahul.gupta@gmail.com	Rahul Gupta
Edit Copy Delete	abhinav.sharma@gmail.com	Abhinav Sharma
Edit Copy Delete	anubhav.patel@gmail.com	Anubhav Patel
Edit Copy Delete	neha.verma@gmail.com	Neha Verma
Edit Copy Delete	john.bernard@gmail.com	John Bernard
Edit Copy Delete	muhammad.ali@gmail.com	Muhammad Ali
Edit Copy Delete	ahmed.khan@gmail.com	Ahmed Khan
Edit Copy Delete	paras.rana@gmail.com	Paras Rana

Q8 Which pincode has average amount paid more than 150,000? Take only 'buy' order types

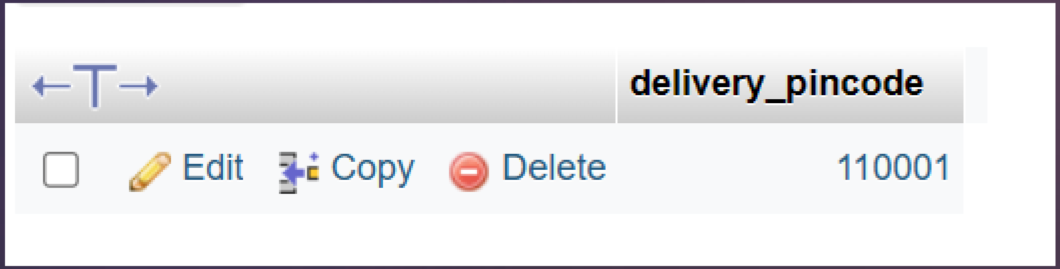
Query :-



```
SELECT delivery_pincode FROM order_dim WHERE order_type = 'buy' GROUP BY delivery_pincode HAVING AVG(total_amount_paid) > 150000;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table

Output :-

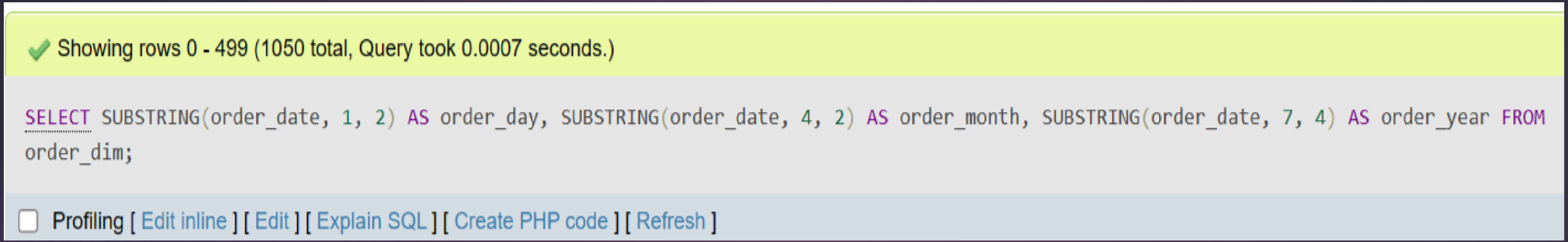


	delivery_pincode
Edit Copy Delete	110001

Q9 Create following columns from order_dim data -

- order_date
- Order day
- Order month
- Order year

Query :-

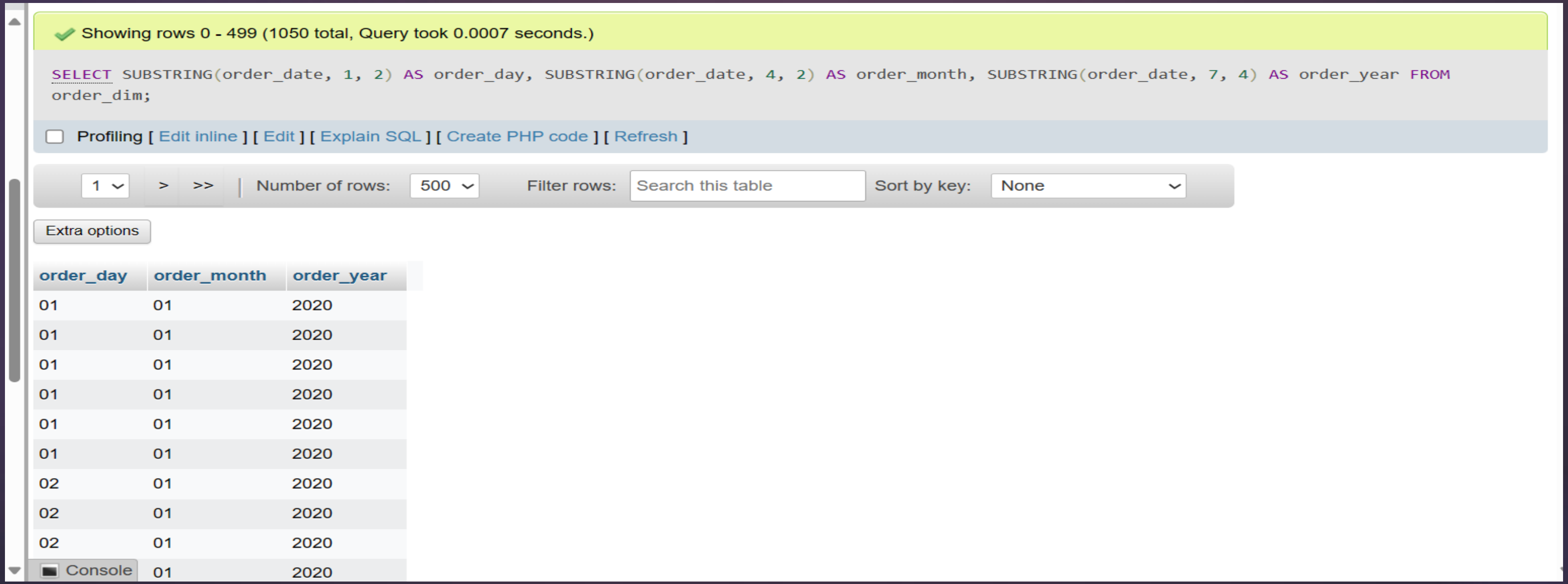


Showing rows 0 - 499 (1050 total, Query took 0.0007 seconds.)

```
SELECT SUBSTRING(order_date, 1, 2) AS order_day, SUBSTRING(order_date, 4, 2) AS order_month, SUBSTRING(order_date, 7, 4) AS order_year FROM
order_dim;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Output :-



Showing rows 0 - 499 (1050 total, Query took 0.0007 seconds.)

```
SELECT SUBSTRING(order_date, 1, 2) AS order_day, SUBSTRING(order_date, 4, 2) AS order_month, SUBSTRING(order_date, 7, 4) AS order_year FROM
order_dim;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Number of rows: 500 Filter rows: Search this table Sort by key: None

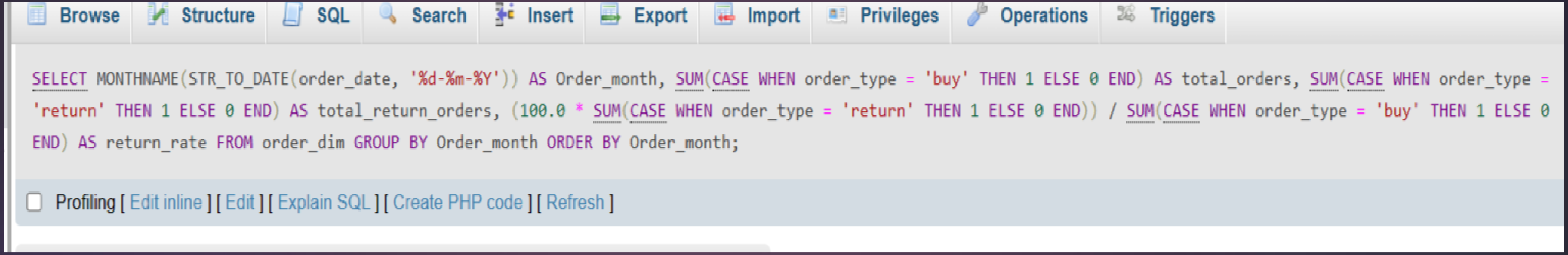
Extra options

order_day	order_month	order_year
01	01	2020
01	01	2020
01	01	2020
01	01	2020
01	01	2020
01	01	2020
02	01	2020
02	01	2020
02	01	2020
	01	2020

Console

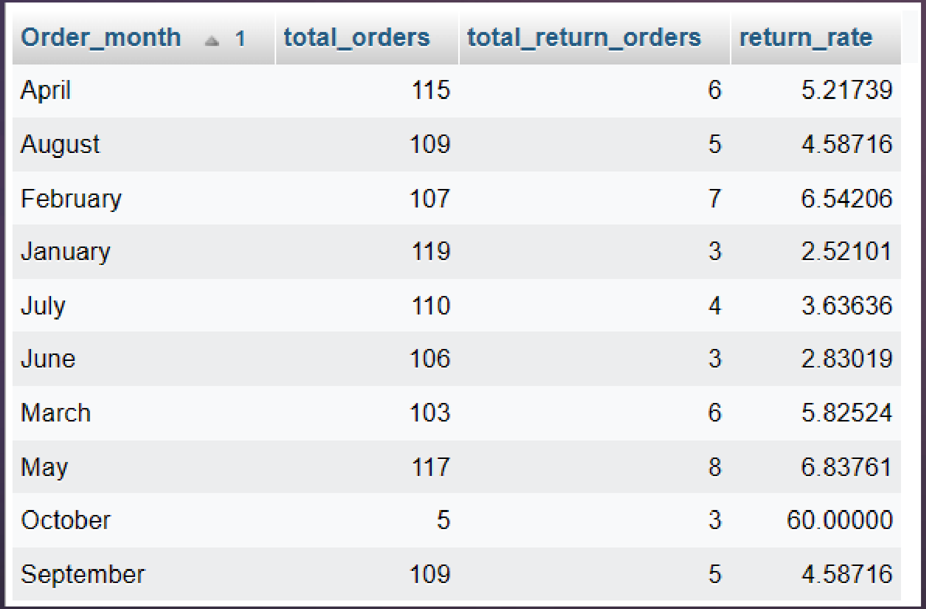
Q10 How many total orders were there in each month and how many of them were returned? Add a column for return rate too. return rate = (100.0 * total return orders) / total buy orders Hint: You will need to combine SUM() with CASE WHEN

Query :-


Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

```
SELECT MONTHNAME(STR_TO_DATE(order_date, '%d-%m-%Y')) AS Order_month, SUM(CASE WHEN order_type = 'buy' THEN 1 ELSE 0 END) AS total_orders, SUM(CASE WHEN order_type =
'return' THEN 1 ELSE 0 END) AS total_return_orders, (100.0 * SUM(CASE WHEN order_type = 'return' THEN 1 ELSE 0 END)) / SUM(CASE WHEN order_type = 'buy' THEN 1 ELSE 0
END) AS return_rate FROM order_dim GROUP BY Order_month ORDER BY Order_month;
```

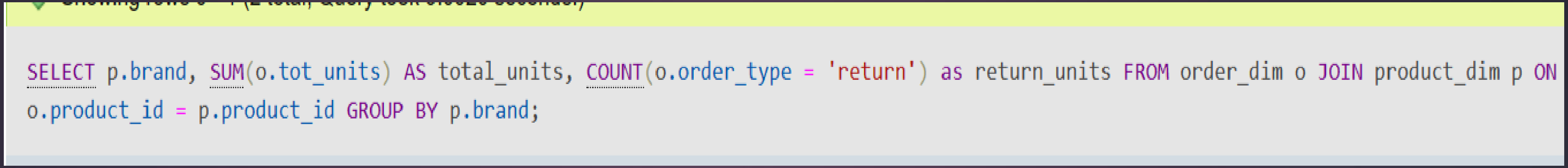
Output :-



Order_month ▲ 1	total_orders	total_return_orders	return_rate
April	115	6	5.21739
August	109	5	4.58716
February	107	7	6.54206
January	119	3	2.52101
July	110	4	3.63636
June	106	3	2.83019
March	103	6	5.82524
May	117	8	6.83761
October	5	3	60.00000
September	109	5	4.58716

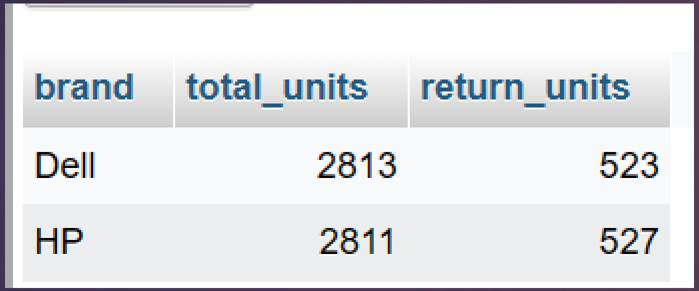
Q11 How many units have been sold by each brand? Also get total returned units for each brand.

Query :-



```
SELECT p.brand, SUM(o.tot_units) AS total_units, COUNT(o.order_type = 'return') as return_units FROM order_dim o JOIN product_dim p ON
o.product_id = p.product_id GROUP BY p.brand;
```

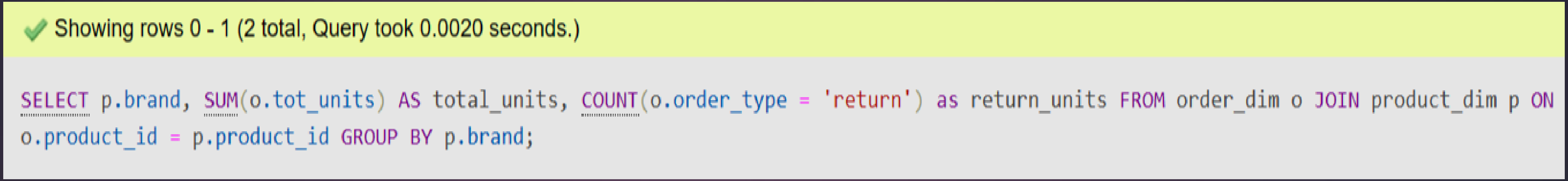
Output :-



brand	total_units	return_units
Dell	2813	523
HP	2811	527

Q12 How many distinct customers and delivery boys are there in each state?

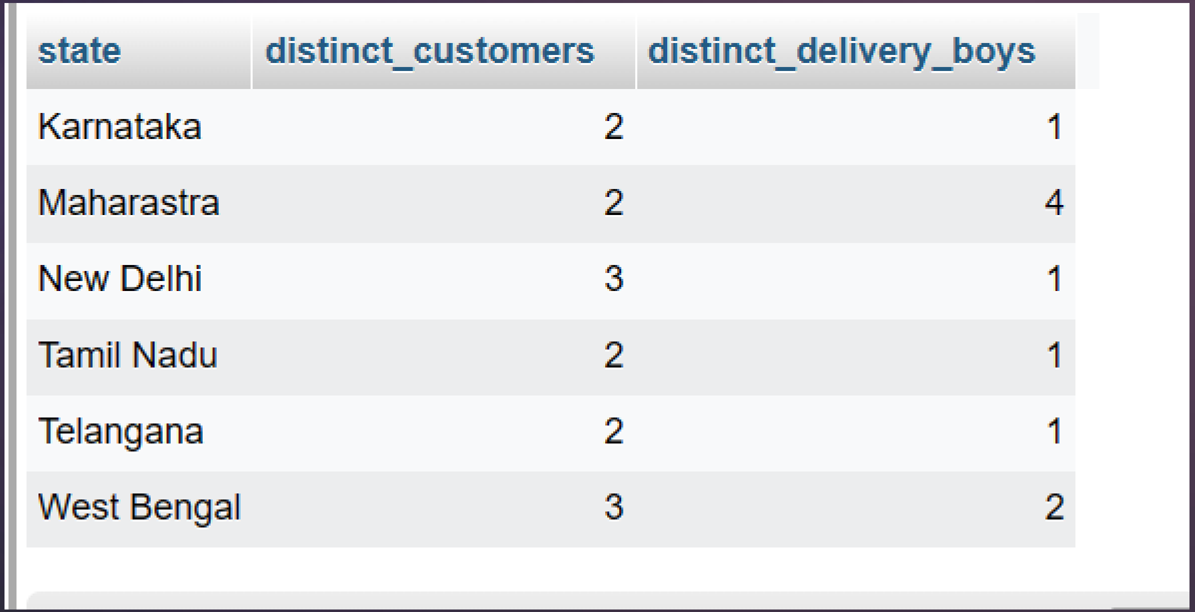
Query :-



✓ Showing rows 0 - 1 (2 total, Query took 0.0020 seconds.)

```
SELECT p.brand, SUM(o.tot_units) AS total_units, COUNT(o.order_type = 'return') as return_units FROM order_dim o JOIN product_dim p ON
o.product_id = p.product_id GROUP BY p.brand;
```

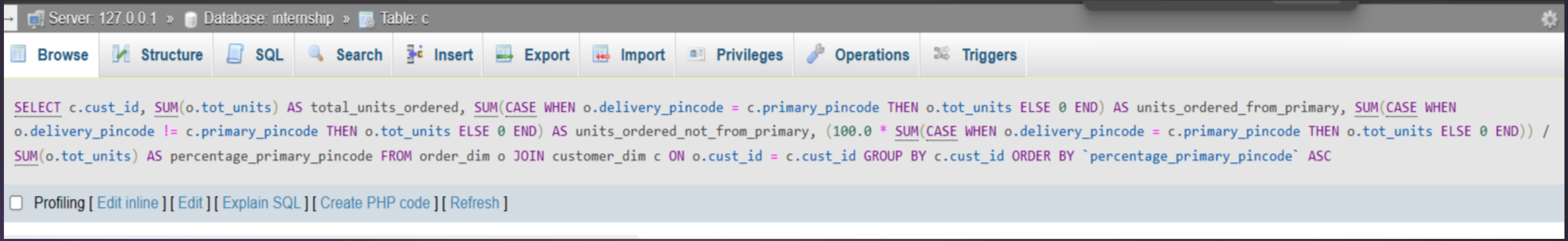
Output :-



state	distinct_customers	distinct_delivery_boys
Karnataka	2	1
Maharastra	2	4
New Delhi	3	1
Tamil Nadu	2	1
Telangana	2	1
West Bengal	3	2

Q13 For every customer, print how many total units were ordered, how many units were ordered from their primary_pincode and how many were ordered not from the primary_pincode. Also calulate the percentage of total units which were ordered from primary_pincode(remember to multiply the numerator by 100.0). Sort by the percentage column in descending order.

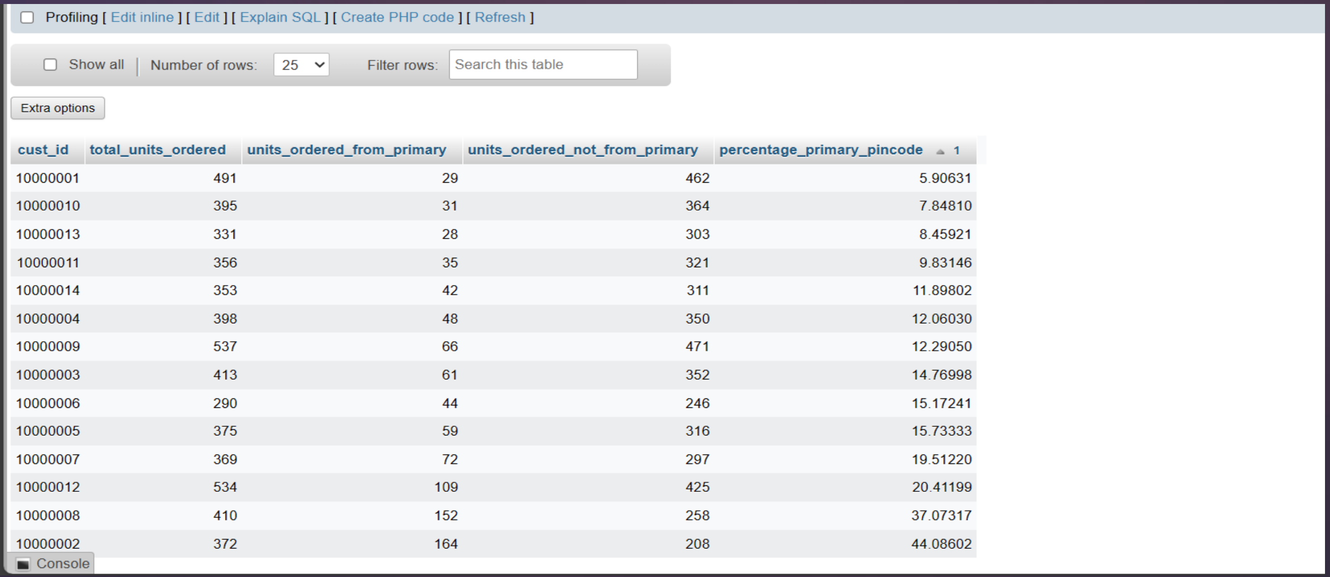
Query :-



Server: 127.0.0.1 » Database: internship » Table: c
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

```sql
SELECT c.cust_id, SUM(o.tot_units) AS total_units_ordered, SUM(CASE WHEN o.delivery_pincode = c.primary_pincode THEN o.tot_units ELSE 0 END) AS units_ordered_from_primary, SUM(CASE WHEN
o.delivery_pincode != c.primary_pincode THEN o.tot_units ELSE 0 END) AS units_ordered_not_from_primary, (100.0 * SUM(CASE WHEN o.delivery_pincode = c.primary_pincode THEN o.tot_units ELSE 0 END)) /
SUM(o.tot_units) AS percentage_primary_pincode FROM order_dim o JOIN customer_dim c ON o.cust_id = c.cust_id GROUP BY c.cust_id ORDER BY `percentage_primary_pincode` ASC
```

Output :-

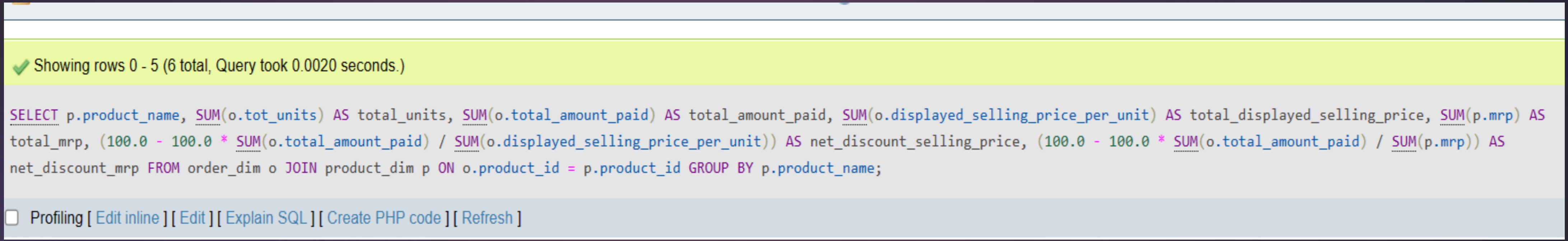


Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Extra options
Console

cust_id	total_units_ordered	units_ordered_from_primary	units_ordered_not_from_primary	percentage_primary_pincode 1
10000001	491	29	462	5.90631
10000010	395	31	364	7.84810
10000013	331	28	303	8.45921
10000011	356	35	321	9.83146
10000014	353	42	311	11.89802
10000004	398	48	350	12.06030
10000009	537	66	471	12.29050
10000003	413	61	352	14.76998
10000006	290	44	246	15.17241
10000005	375	59	316	15.73333
10000007	369	72	297	19.51220
10000012	534	109	425	20.41199
10000008	410	152	258	37.07317
10000002	372	164	208	44.08602

Q14 For each product name, print the sum of number of units, total amount paid, total displayed selling price, total mrp of these units, and finally the net discount from selling price. (i.e. 100.0 - 100.0 * total amount paid / total displayed selling price) & the net discount from mrp (i.e. 100.0 - 100.0 * total amount paid / total mrp)

Query :-

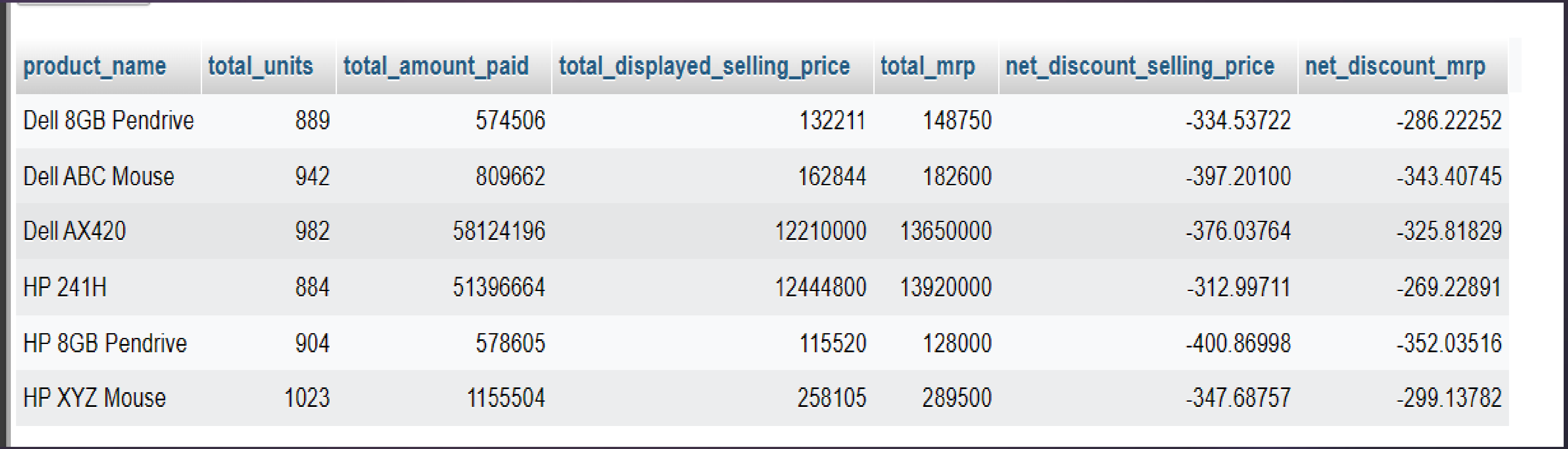


Showing rows 0 - 5 (6 total, Query took 0.0020 seconds.)

```
SELECT p.product_name, SUM(o.tot_units) AS total_units, SUM(o.total_amount_paid) AS total_amount_paid, SUM(o.displayed_selling_price_per_unit) AS total_displayed_selling_price, SUM(p.mrp) AS
total_mrp, (100.0 - 100.0 * SUM(o.total_amount_paid) / SUM(o.displayed_selling_price_per_unit)) AS net_discount_selling_price, (100.0 - 100.0 * SUM(o.total_amount_paid) / SUM(p.mrp)) AS
net_discount_mrp FROM order_dim o JOIN product_dim p ON o.product_id = p.product_id GROUP BY p.product_name;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

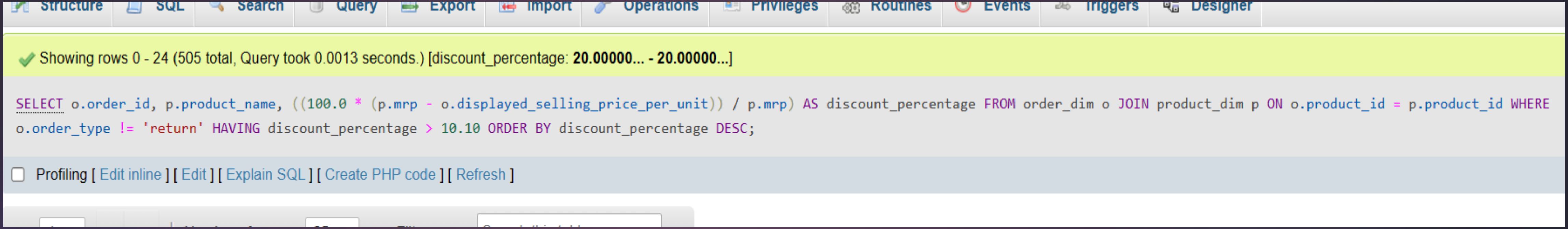
Output :-



product_name	total_units	total_amount_paid	total_displayed_selling_price	total_mrp	net_discount_selling_price	net_discount_mrp
Dell 8GB Pendrive	889	574506	132211	148750	-334.53722	-286.22252
Dell ABC Mouse	942	809662	162844	182600	-397.20100	-343.40745
Dell AX420	982	58124196	12210000	13650000	-376.03764	-325.81829
HP 241H	884	51396664	12444800	13920000	-312.99711	-269.22891
HP 8GB Pendrive	904	578605	115520	128000	-400.86998	-352.03516
HP XYZ Mouse	1023	1155504	258105	289500	-347.68757	-299.13782

Q15 For every order_id (exclude returns), get the product name and calculate the discount percentage from selling price. Sort by highest discount and print only those rows where discount percentage was above 10.10%.

Query :-

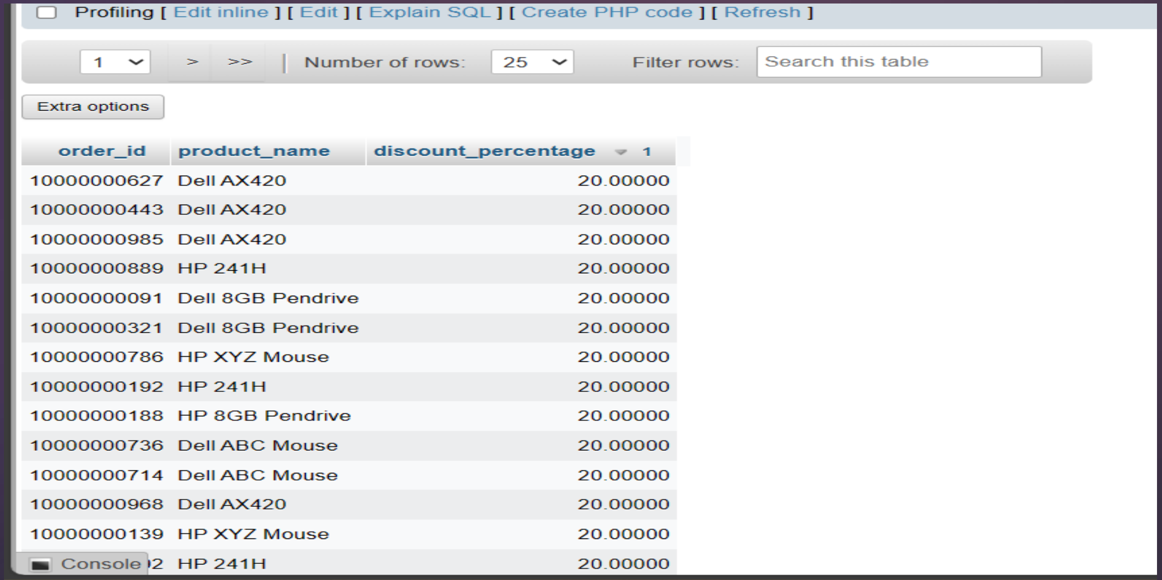


✓ Showing rows 0 - 24 (505 total, Query took 0.0013 seconds.) [discount_percentage: **20.00000... - 20.00000...**]

```sql
SELECT o.order_id, p.product_name, ((100.0 * (p.mrp - o.displayed_selling_price_per_unit)) / p.mrp) AS discount_percentage FROM order_dim o JOIN product_dim p ON o.product_id = p.product_id WHERE o.order_type != 'return' HAVING discount_percentage > 10.10 ORDER BY discount_percentage DESC;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

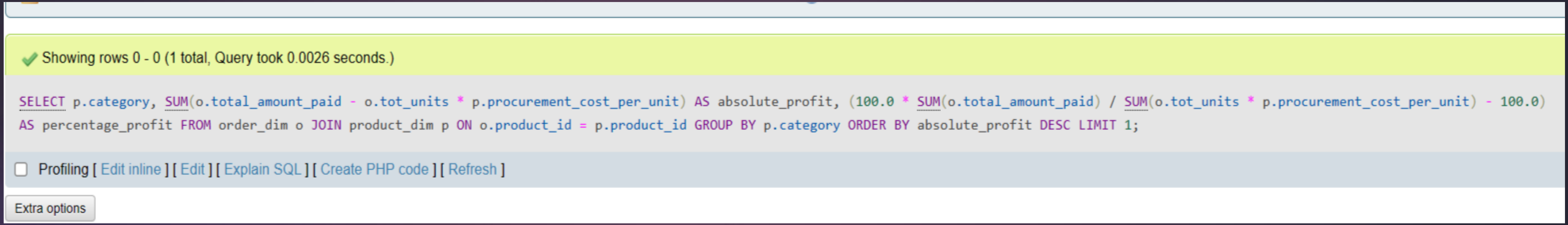
Output :-


Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1 > >> | Number of rows: 25 Filter rows: Search this table
Extra options

order_id	product_name	discount_percentage ▾ 1
10000000627	Dell AX420	20.00000
10000000443	Dell AX420	20.00000
10000000985	Dell AX420	20.00000
10000000889	HP 241H	20.00000
10000000091	Dell 8GB Pendrive	20.00000
10000000321	Dell 8GB Pendrive	20.00000
10000000786	HP XYZ Mouse	20.00000
10000000192	HP 241H	20.00000
10000000188	HP 8GB Pendrive	20.00000
10000000736	Dell ABC Mouse	20.00000
10000000714	Dell ABC Mouse	20.00000
10000000968	Dell AX420	20.00000
10000000139	HP XYZ Mouse	20.00000
Console)2	HP 241H	20.00000

Q16 Using the per unit procurement cost in product_dim, find which product category has made the most profit in both absolute amount and percentage
Absolute Profit = Total Amt Sold - Total Procurement Cost
Percentage Profit = 100.0 * Total Amt Sold / Total Procurement Cost - 100.0

Query :-

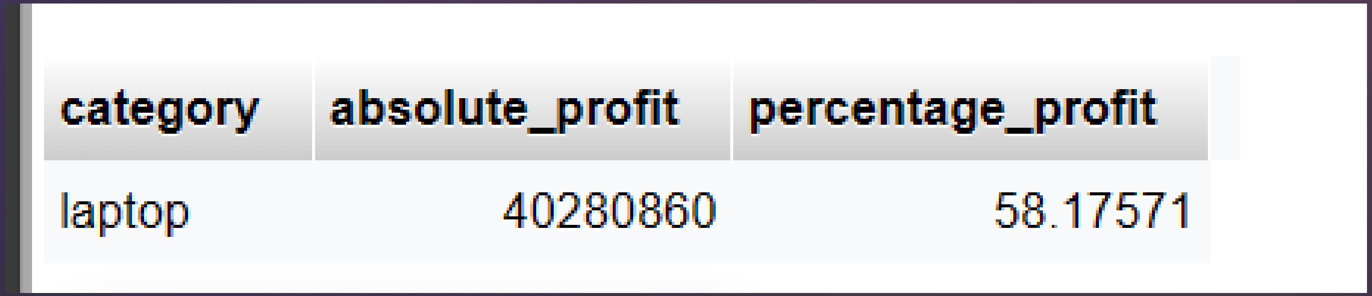

Showing rows 0 - 0 (1 total, Query took 0.0026 seconds.)

```
SELECT p.category, SUM(o.total_amount_paid - o.tot_units * p.procurement_cost_per_unit) AS absolute_profit, (100.0 * SUM(o.total_amount_paid) / SUM(o.tot_units * p.procurement_cost_per_unit) - 100.0)
AS percentage_profit FROM order_dim o JOIN product_dim p ON o.product_id = p.product_id GROUP BY p.category ORDER BY absolute_profit DESC LIMIT 1;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

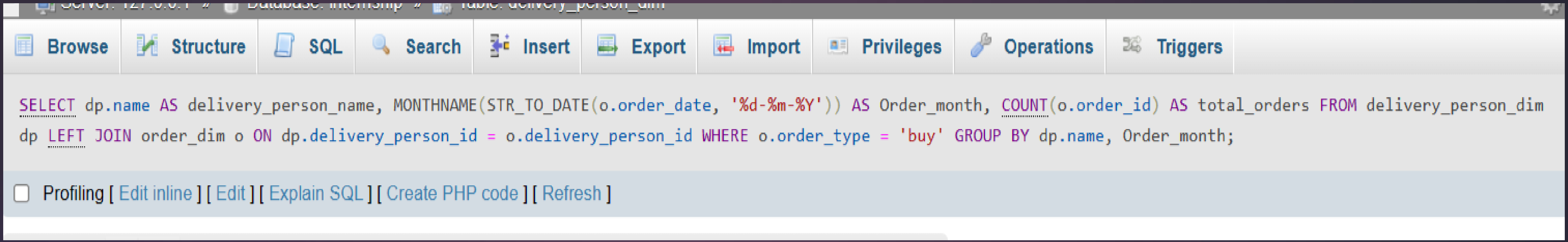
Output :-



category	absolute_profit	percentage_profit
laptop	40280860	58.17571

Q17 For every delivery person(use their name), print the total number of order ids (exclude returns) by month in separate columns i.e. there should be one row for each delivery_person_id and 12 columns for every month in the year

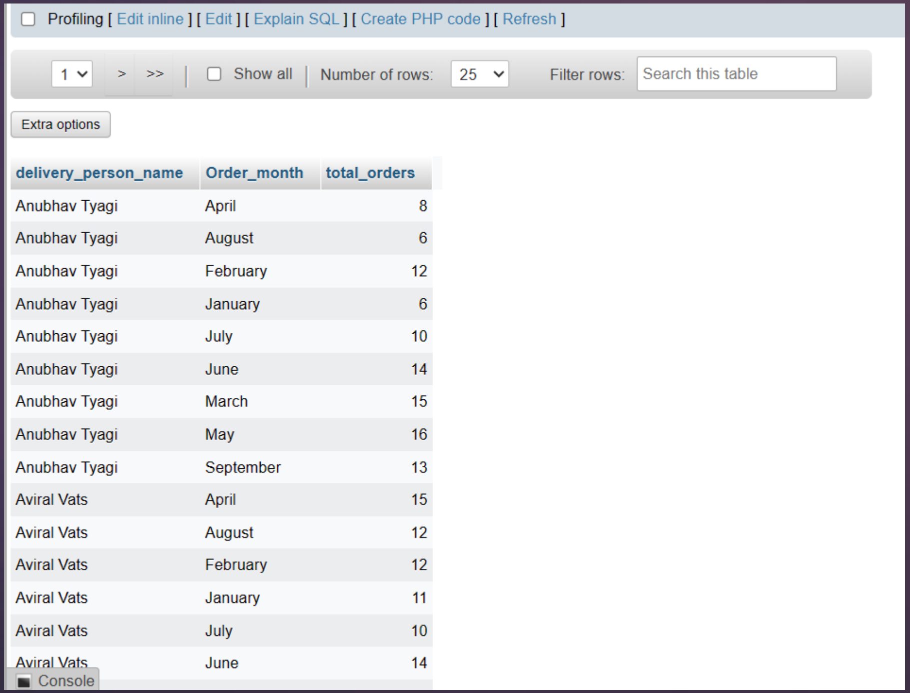
Query :-


Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers

```sql
SELECT dp.name AS delivery_person_name, MONTHNAME(STR_TO_DATE(o.order_date, '%d-%m-%Y')) AS Order_month, COUNT(o.order_id) AS total_orders FROM delivery_person_dim
dp LEFT JOIN order_dim o ON dp.delivery_person_id = o.delivery_person_id WHERE o.order_type = 'buy' GROUP BY dp.name, Order_month;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

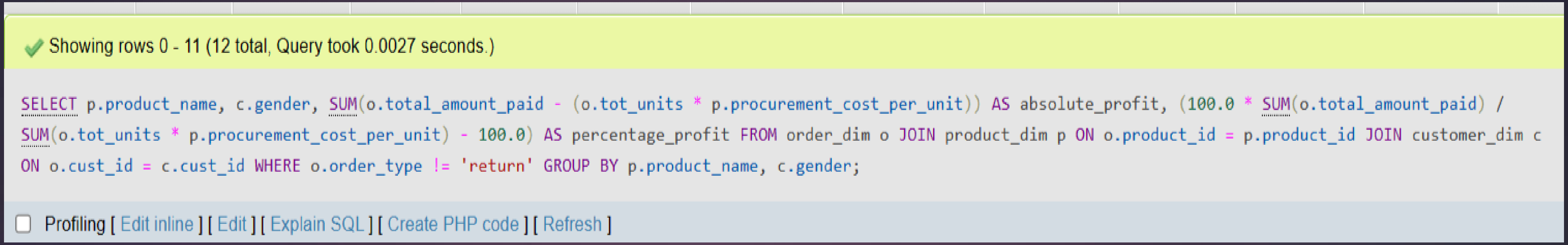
Output :-


Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1
> >>
Show all
Number of rows: 25
Filter rows: Search this table
Extra options
Console

delivery_person_name	Order_month	total_orders
Anubhav Tyagi	April	8
Anubhav Tyagi	August	6
Anubhav Tyagi	February	12
Anubhav Tyagi	January	6
Anubhav Tyagi	July	10
Anubhav Tyagi	June	14
Anubhav Tyagi	March	15
Anubhav Tyagi	May	16
Anubhav Tyagi	September	13
Aviral Vats	April	15
Aviral Vats	August	12
Aviral Vats	February	12
Aviral Vats	January	11
Aviral Vats	July	10
Aviral Vats	June	14

Q18 For each gender - male and female - find the absolute and percentage profit (like in Q15) by product name

Query :-

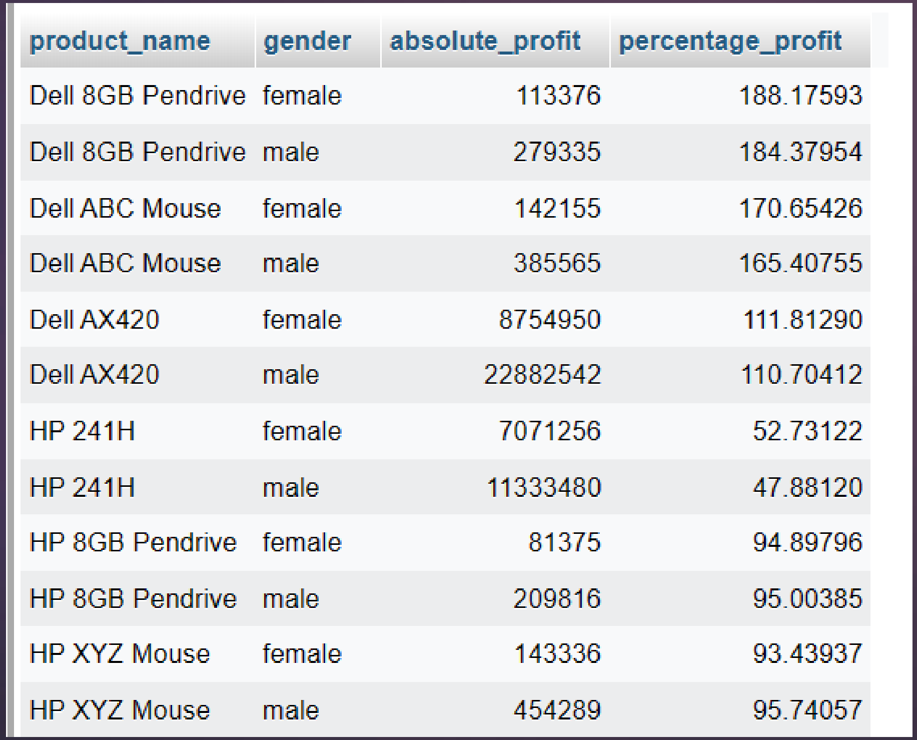


✓ Showing rows 0 - 11 (12 total, Query took 0.0027 seconds.)

```
SELECT p.product_name, c.gender, SUM(o.total_amount_paid - (o.tot_units * p.procurement_cost_per_unit)) AS absolute_profit, (100.0 * SUM(o.total_amount_paid) /
SUM(o.tot_units * p.procurement_cost_per_unit) - 100.0) AS percentage_profit FROM order_dim o JOIN product_dim p ON o.product_id = p.product_id JOIN customer_dim c
ON o.cust_id = c.cust_id WHERE o.order_type != 'return' GROUP BY p.product_name, c.gender;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

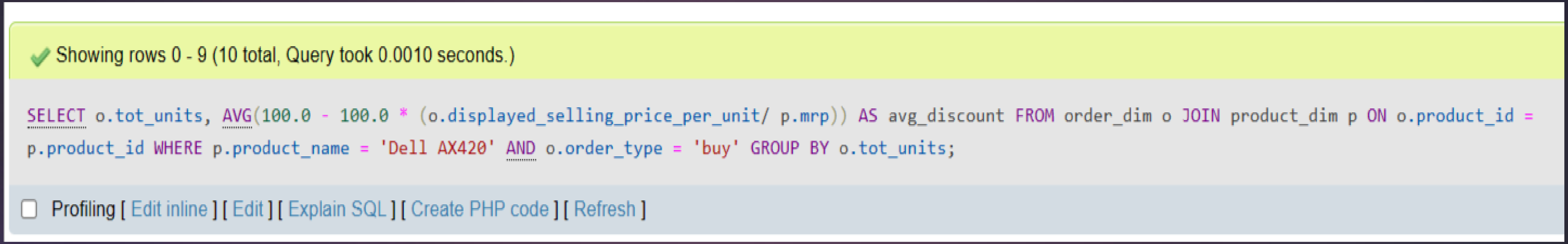
Output :-



product_name	gender	absolute_profit	percentage_profit
Dell 8GB Pendrive	female	113376	188.17593
Dell 8GB Pendrive	male	279335	184.37954
Dell ABC Mouse	female	142155	170.65426
Dell ABC Mouse	male	385565	165.40755
Dell AX420	female	8754950	111.81290
Dell AX420	male	22882542	110.70412
HP 241H	female	7071256	52.73122
HP 241H	male	11333480	47.88120
HP 8GB Pendrive	female	81375	94.89796
HP 8GB Pendrive	male	209816	95.00385
HP XYZ Mouse	female	143336	93.43937
HP XYZ Mouse	male	454289	95.74057

Q19 Generally the more numbers of units you buy, the more discount seller will give you. For 'Dell AX420' is there a relationship between number of units ordered and average discount from selling price? Take only 'buy' order types

Query :-

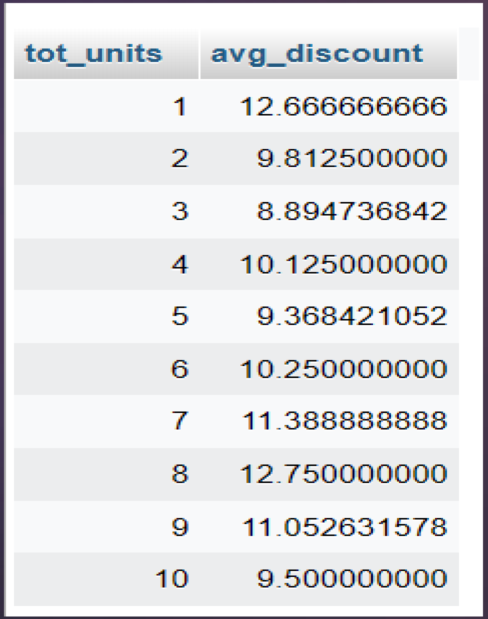


Showing rows 0 - 9 (10 total, Query took 0.0010 seconds.)

```
SELECT o.tot_units, AVG(100.0 - 100.0 * (o.displayed_selling_price_per_unit/ p.mrp)) AS avg_discount FROM order_dim o JOIN product_dim p ON o.product_id = p.product_id WHERE p.product_name = 'Dell AX420' AND o.order_type = 'buy' GROUP BY o.tot_units;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Output :-



tot_units	avg_discount
1	12.666666666
2	9.812500000
3	8.894736842
4	10.125000000
5	9.368421052
6	10.250000000
7	11.388888888
8	12.750000000
9	11.052631578
10	9.500000000

Conclusion

There were 5 Columns and using that analysis is done by SQL query.

By leveraging insights from the "Customers" table, fostering stronger customer relationships.

The "Products" table enables efficient inventory management.

"Pincode" table allow for streamlined logistics.

"Delivery Person" table ensures accurate delivery of orders.

Overall, the comprehensive framework provided by these tables enables the delivery service to adapt to market dynamics, optimize resource allocation, and deliver exceptional service quality.

Thank You!!