SQL CAPSTONE PROJECT

AMAZON SALES ANALYSIS

Suryakant ranjan

OBJECTIVE

- The major aim of this project is to gain insight into the sales data of Amazon to understand the different factors that affect sales of the different branches.

ABOUT DATA:

- This dataset contains sales transactions from three different branches of Amazon, respectively located in Mandalay, Yangon and Naypyitaw. The data contains 17 columns and 1000 rows:

Column	Description	Data Type
invoice_id	Invoice of the sales made	VARCHAR(30)
branch	Branch at which sales were made	VARCHAR(5)
city	The location of the branch	VARCHAR(30)
customer_type	The type of the customer	VARCHAR(30)
gender	Gender of the customer making purchase	VARCHAR(10)
product_line	Product line of the product sold	VARCHAR(100)
unit_price	The price of each product	DECIMAL(10, 2)
quantity	The amount of the product sold	INT

VAT	The amount of tax on the purchase	FLOAT(6, 4)
total	The total cost of the purchase	DECIMAL(10, 2)
date	The date on which the purchase was made	DATE
time	The time at which the purchase was made	TIMESTAMP
payment_method	The total amount paid	DECIMAL(10, 2)
cogs	Cost Of Goods sold	DECIMAL(10, 2)
gross_margin_percentage	Gross margin percentage	FLOAT(11, 9)
gross_income	Gross Income	DECIMAL(10, 2)
rating	Rating	FLOAT(2, 1)

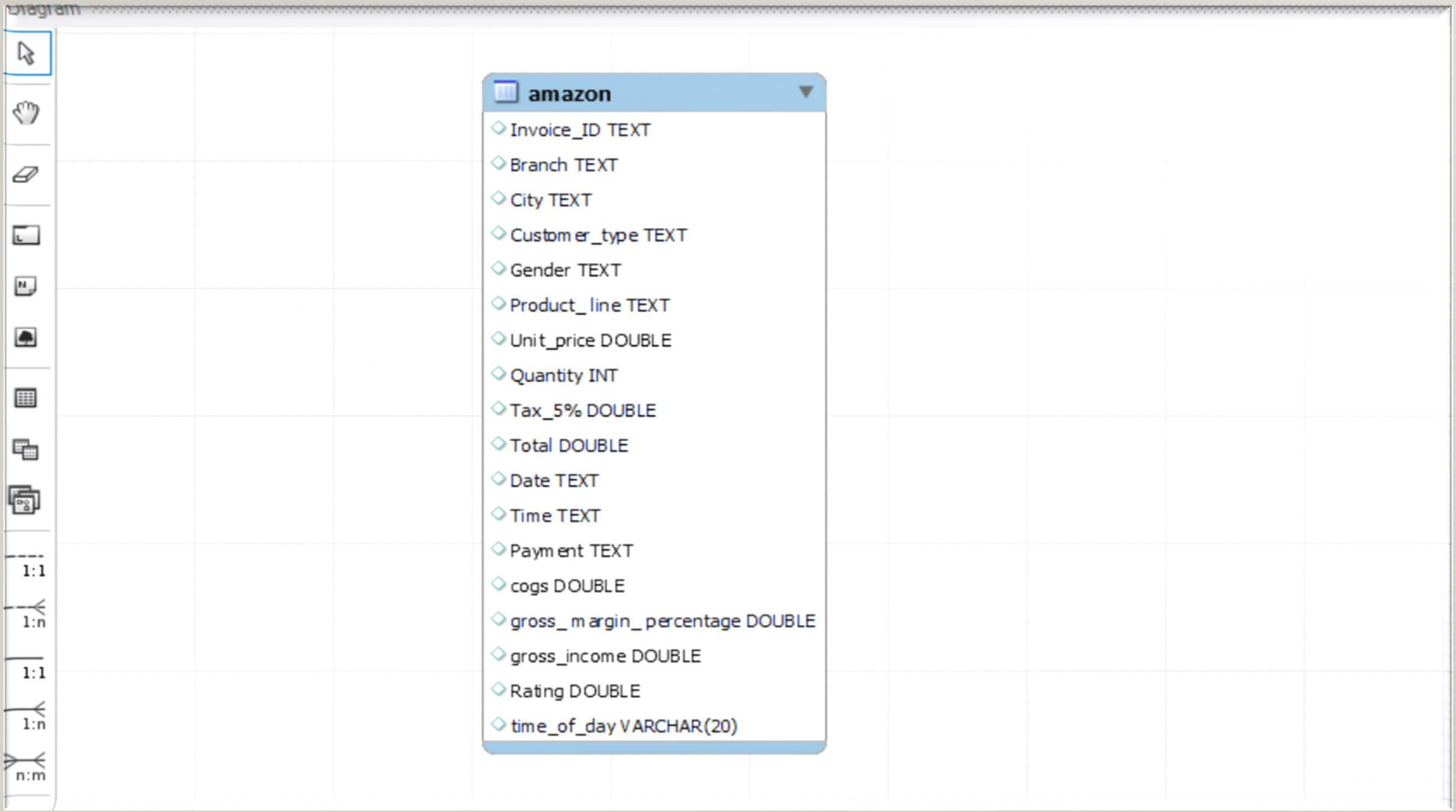


Diagram
amazon
Invoice_ID TEXT
Branch TEXT
City TEXT
Customer_type TEXT
Gender TEXT
Product_ line TEXT
Unit_price DOUBLE
Quantity INT
Tax_5% DOUBLE
Total DOUBLE
Date TEXT
Time TEXT
Payment TEXT
cogs DOUBLE
gross_ margin_ percentage DOUBLE
gross_income DOUBLE
Rating DOUBLE
time_of_day VARCHAR(20)
1:1
1:n
1:1
1:n
n:m

1) PRODUCT ANALYSIS :-

- Conduct analysis on the data to understand the different product lines, the products lines performing best and the product lines that need to be improved.

2)SALES ANALYSIS :-

- This analysis aims to answer the question of the sales trends of product. The result of this can help us measure the effectiveness of each sales strategy the business applies and what modifications are needed to gain more sales.

3) CUSTOMER ANALYSIS :-

- This analysis aims to uncover the different customer segments, purchase trends and the profitability of each customer segment.

APPROACH USED :-

1) Data Wrangling: This is the first step where inspection of data is done to make sure NULL values and missing values are detected and data replacement methods are used to replace missing or NULL values.

- 1.1 Build a database

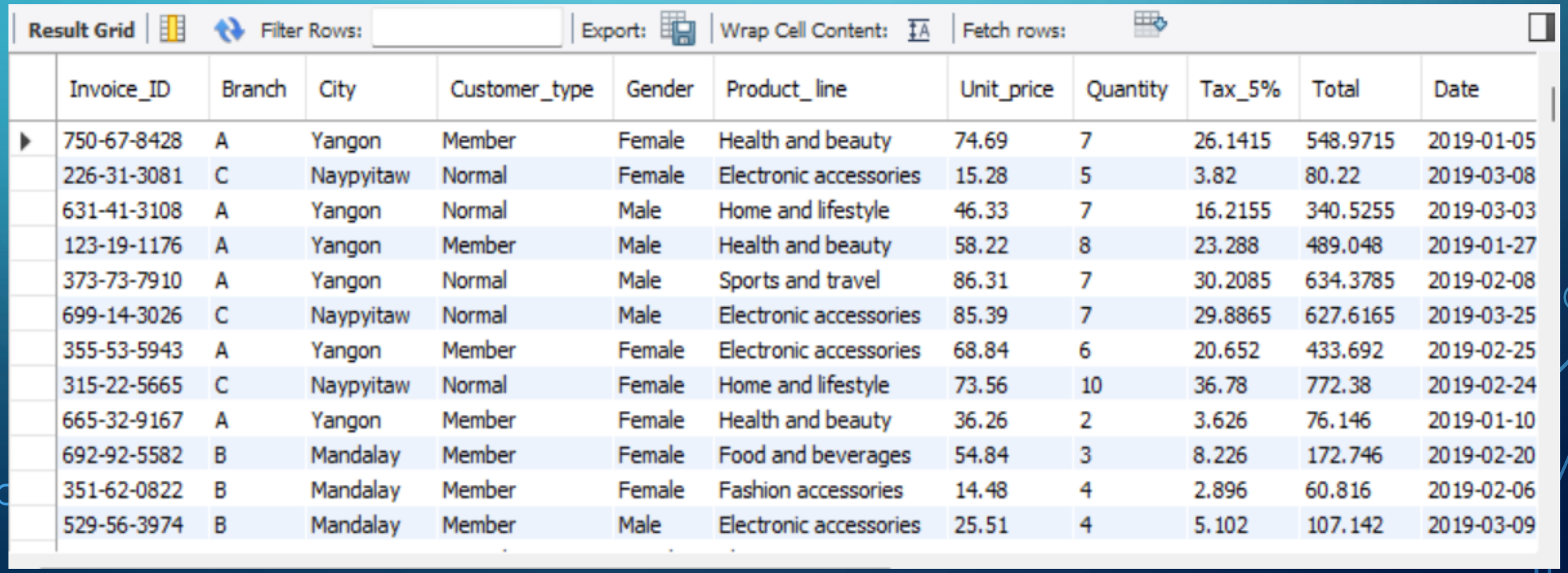
```
create database capstone_project;
```

1.2 Create a table and insert the data.

```
Create table if not exists amazon (
 invoice_id  varchar(30) Not null primary Key,
 branch  varchar (5) not null,
    city varchar (30) not null,
    customer_type varchar (30) not null,
    gender  varchar(10) not null,
    product_line varchar (100) not null,
    unit_price  decimal(10,2) not null,
    quantity int not null,
    VAT  float(6,4) not null,
    total decimal (12,4) not null,
    date datetime not null,
    time TIME not null,
    payment_method varchar (15)  not null,
    cogs  decimal (10, 2) not null,
    gross_margin_pct float (11,9),
    gross_income decimal(12, 4) not null,
    rating float (2, 1)
);
```

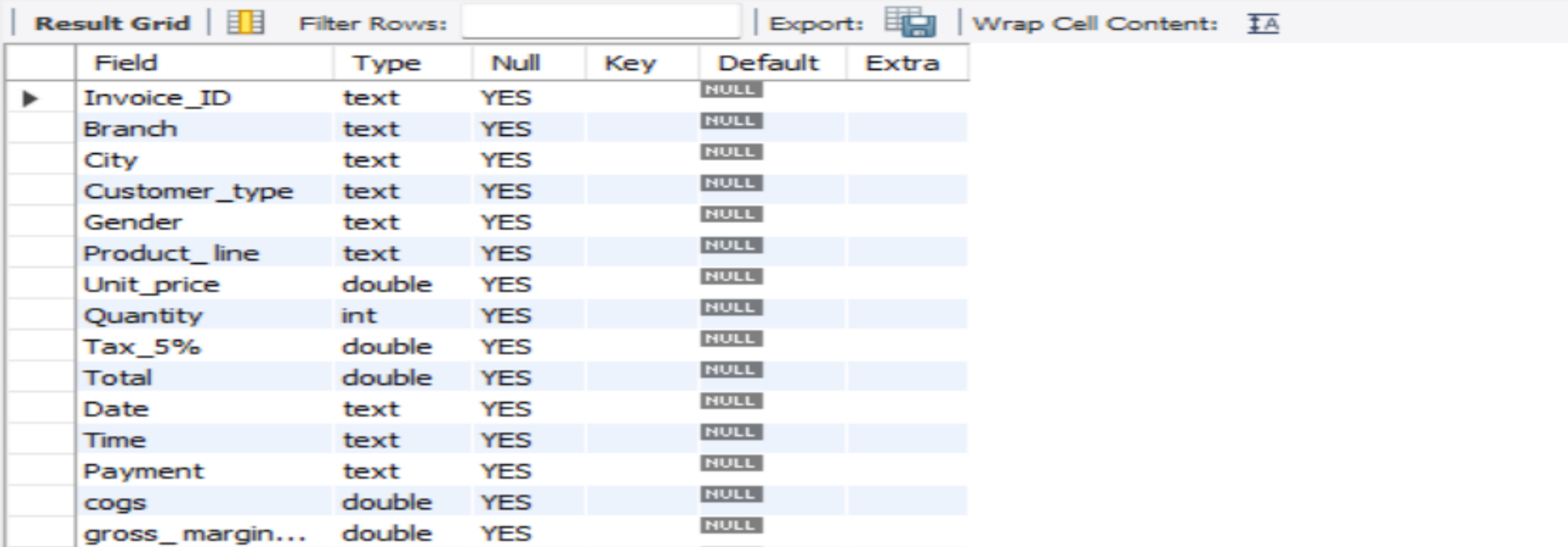
Import Data from CSV

After creating a database, import data from a CSV file.


Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Invoice_ID	Branch	City	Customer_type	Gender	Product_line	Unit_price	Quantity	Tax_5%	Total	Date
750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	2019-01-05
226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.82	80.22	2019-03-08
631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	2019-03-03
123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.288	489.048	2019-01-27
373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2019-02-08
699-14-3026	C	Naypyitaw	Normal	Male	Electronic accessories	85.39	7	29.8865	627.6165	2019-03-25
355-53-5943	A	Yangon	Member	Female	Electronic accessories	68.84	6	20.652	433.692	2019-02-25
315-22-5665	C	Naypyitaw	Normal	Female	Home and lifestyle	73.56	10	36.78	772.38	2019-02-24
665-32-9167	A	Yangon	Member	Female	Health and beauty	36.26	2	3.626	76.146	2019-01-10
692-92-5582	B	Mandalay	Member	Female	Food and beverages	54.84	3	8.226	172.746	2019-02-20
351-62-0822	B	Mandalay	Member	Female	Fashion accessories	14.48	4	2.896	60.816	2019-02-06
529-56-3974	B	Mandalay	Member	Male	Electronic accessories	25.51	4	5.102	107.142	2019-03-09

1.3 SELECT COLUMNS WITH NULL VALUES IN THEM. THERE ARE NO NULL VALUES IN OUR DATABASE AS IN CREATING THE TABLES, WE SET NOT NULL FOR EACH FIELD, HENCE NULL VALUES ARE FILTERED OUT.


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Invoice_ID	text	YES		NULL	
Branch	text	YES		NULL	
City	text	YES		NULL	
Customer_type	text	YES		NULL	
Gender	text	YES		NULL	
Product_ line	text	YES		NULL	
Unit_price	double	YES		NULL	
Quantity	int	YES		NULL	
Tax_5%	double	YES		NULL	
Total	double	YES		NULL	
Date	text	YES		NULL	
Time	text	YES		NULL	
Payment	text	YES		NULL	
cogs	double	YES		NULL	
gross_ margin...	double	YES		NULL	

2) FEATURE ENGINEERING: THIS WILL HELP US GENERATE SOME NEW COLUMNS FROM EXISTING ONES.

- 2.1 Add a new column named time_of_day to give insight of sales in the Morning, Afternoon and Evening. This will help answer the question on which part of the day most sales are made.

QUERY :-

```
SELECT
 time,
 (CASE
 WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"
 WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"
 ELSE "Evening"
 END) AS time_of_day
FROM amazon;

ALTER TABLE amazon ADD COLUMN time_of_day VARCHAR(20);

UPDATE amazon
SET time_of_day = (
 CASE
  WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"
        WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"
        ELSE "Evening"
    END
);
```

OUTPUT :-

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ntity	Tax_5%	Total	Date	Time	Payment	cogs	gross_ margin_ percentage	gross_income	Rating	time_of_day
	26.1415	548.9715	2019-01-05	13:08:00	Ewallet	522.83	4.761904762	26.1415	9.1	Afternoon
	3.82	80.22	2019-03-08	10:29:00	Cash	76.4	4.761904762	3.82	9.6	Morning
	16.2155	340.5255	2019-03-03	13:23:00	Credit card	324.31	4.761904762	16.2155	7.4	Afternoon
	23.288	489.048	2019-01-27	20:33:00	Ewallet	465.76	4.761904762	23.288	8.4	Evening
	30.2085	634.3785	2019-02-08	10:37:00	Ewallet	604.17	4.761904762	30.2085	5.3	Morning
	29.8865	627.6165	2019-03-25	18:30:00	Ewallet	597.73	4.761904762	29.8865	4.1	Evening
	20.652	433.692	2019-02-25	14:36:00	Ewallet	413.04	4.761904762	20.652	5.8	Afternoon
	36.78	772.38	2019-02-24	11:38:00	Ewallet	735.6	4.761904762	36.78	8	Morning
	3.626	76.146	2019-01-10	17:15:00	Credit card	72.52	4.761904762	3.626	7.2	Evening
	8.226	172.746	2019-02-20	13:27:00	Credit card	164.52	4.761904762	8.226	5.9	Afternoon
	2.896	60.816	2019-02-06	18:07:00	Ewallet	57.92	4.761904762	2.896	4.5	Evening
	5.102	107.142	2019-03-09	17:03:00	Cash	102.04	4.761904762	5.102	6.8	Evening
	11.7375	246.4875	2019-02-12	10:25:00	Ewallet	234.75	4.761904762	11.7375	7.1	Morning
	21.595	453.495	2019-02-07	16:48:00	Ewallet	431.9	4.761904762	21.595	8.2	Evening

- 2.2 Add a new column named dayname that contains the extracted days of the week on which the given transaction took place (Mon, Tue, Wed, Thur, Fri). This will help answer the question on which week of the day each branch is busiest.

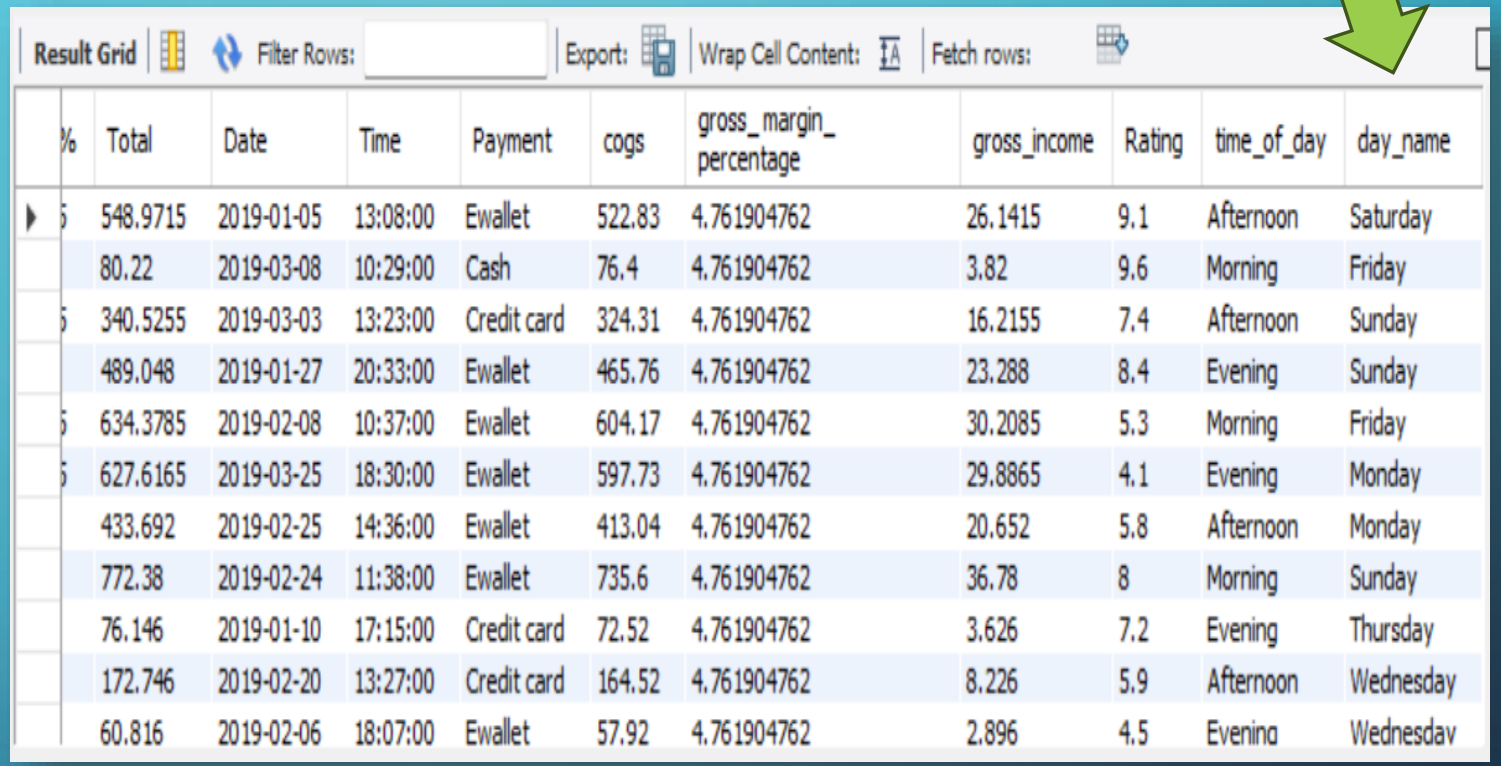
QUERY :-

```
select
 date,
    dayname(date)
from amazon;

alter table amazon add column day_name varchar(10);

update  amazon
set day_name = dayname(date);
```

OUTPUT :-



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

%	Total	Date	Time	Payment	cogs	gross_margin_percentage	gross_income	Rating	time_of_day	day_name
5	548.9715	2019-01-05	13:08:00	Ewallet	522.83	4.761904762	26.1415	9.1	Afternoon	Saturday
	80.22	2019-03-08	10:29:00	Cash	76.4	4.761904762	3.82	9.6	Morning	Friday
5	340.5255	2019-03-03	13:23:00	Credit card	324.31	4.761904762	16.2155	7.4	Afternoon	Sunday
	489.048	2019-01-27	20:33:00	Ewallet	465.76	4.761904762	23.288	8.4	Evening	Sunday
5	634.3785	2019-02-08	10:37:00	Ewallet	604.17	4.761904762	30.2085	5.3	Morning	Friday
5	627.6165	2019-03-25	18:30:00	Ewallet	597.73	4.761904762	29.8865	4.1	Evening	Monday
	433.692	2019-02-25	14:36:00	Ewallet	413.04	4.761904762	20.652	5.8	Afternoon	Monday
	772.38	2019-02-24	11:38:00	Ewallet	735.6	4.761904762	36.78	8	Morning	Sunday
	76.146	2019-01-10	17:15:00	Credit card	72.52	4.761904762	3.626	7.2	Evening	Thursday
	172.746	2019-02-20	13:27:00	Credit card	164.52	4.761904762	8.226	5.9	Afternoon	Wednesday
	60.816	2019-02-06	18:07:00	Ewallet	57.92	4.761904762	2.896	4.5	Evening	Wednesday

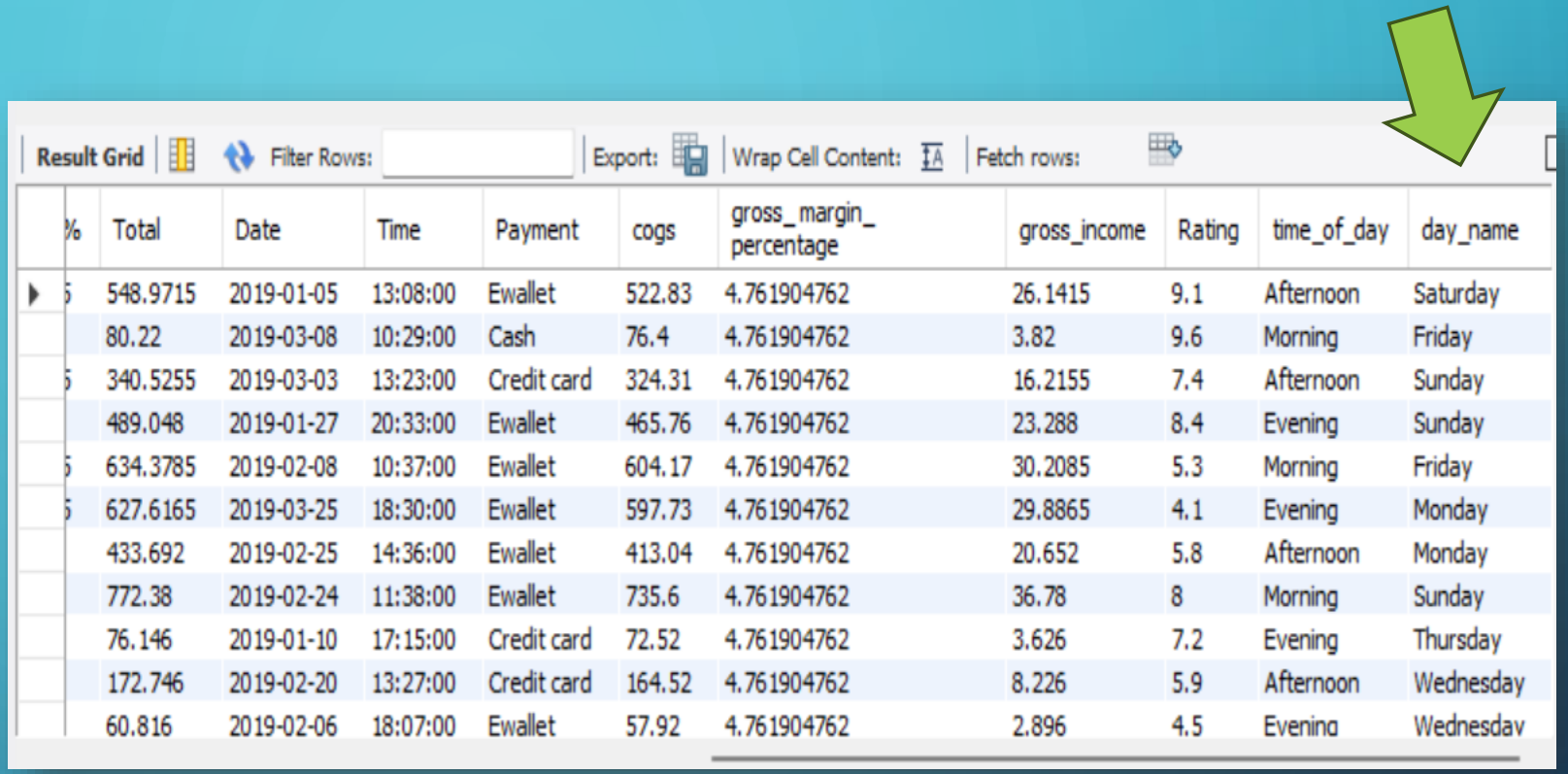
- 2.3 Add a new column named monthname that contains the extracted months of the year on which the given transaction took place (Jan, Feb, Mar). Help determine which month of the year has the most sales and profit.

QUERY :-

```
select date,
    monthname(date)
    from amazon;
alter table amazon add column month_name varchar(20);


update amazon
set month_name = monthname(date);
```

OUTPUT :-



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

%	Total	Date	Time	Payment	cogs	gross_margin_percentage	gross_income	Rating	time_of_day	day_name
5	548.9715	2019-01-05	13:08:00	Ewallet	522.83	4.761904762	26.1415	9.1	Afternoon	Saturday
	80.22	2019-03-08	10:29:00	Cash	76.4	4.761904762	3.82	9.6	Morning	Friday
5	340.5255	2019-03-03	13:23:00	Credit card	324.31	4.761904762	16.2155	7.4	Afternoon	Sunday
	489.048	2019-01-27	20:33:00	Ewallet	465.76	4.761904762	23.288	8.4	Evening	Sunday
5	634.3785	2019-02-08	10:37:00	Ewallet	604.17	4.761904762	30.2085	5.3	Morning	Friday
5	627.6165	2019-03-25	18:30:00	Ewallet	597.73	4.761904762	29.8865	4.1	Evening	Monday
	433.692	2019-02-25	14:36:00	Ewallet	413.04	4.761904762	20.652	5.8	Afternoon	Monday
	772.38	2019-02-24	11:38:00	Ewallet	735.6	4.761904762	36.78	8	Morning	Sunday
	76.146	2019-01-10	17:15:00	Credit card	72.52	4.761904762	3.626	7.2	Evening	Thursday
	172.746	2019-02-20	13:27:00	Credit card	164.52	4.761904762	8.226	5.9	Afternoon	Wednesday
	60.816	2019-02-06	18:07:00	Ewallet	57.92	4.761904762	2.896	4.5	Evening	Wednesday

3. **Exploratory Data Analysis (EDA):** Exploratory data analysis is done to answer the listed questions and aims of this project.

Business Questions To Answer :-

1) WHAT IS THE COUNT OF DISTINCT CITIES IN THE DATASET?

QUERY :-

```
SELECT COUNT(DISTINCT(CITY)) FROM AMAZON;
```

OUTPUT :-

	COUNT(DISTINCT(CITY))
▶	3

2) FOR EACH BRANCH, WHAT IS THE CORRESPONDING CITY?

QUERY :-

```
SELECT distinct(BRANCH) ,CITY from AMAZON;
```

OUTPUT :-

	BRANCH	CITY
▶	A	Yangon
	C	Naypyitaw
	B	Mandalay

3) WHAT IS THE COUNT OF DISTINCT PRODUCT LINES IN THE DATASET?

QUERY :-

```
SELECT COUNT(DISTINCT(PRODUCT_LINE))
AS "UNIQUE PRODUCT LINE" FROM AMAZON;
```

OUTPUT :-

	UNIQUE PRODUCT LINE
▶	6

4)WHICH PAYMENT METHOD OCCURS MOST FREQUENTLY?

QUERY :-

```
SELECT PAYMENT, COUNT(PAYMENT) AS PAYMENT_COUNT FROM AMAZON
GROUP BY PAYMENT
ORDER BY PAYMENT_COUNT DESC
LIMIT 1;
```

OUTPUT :-

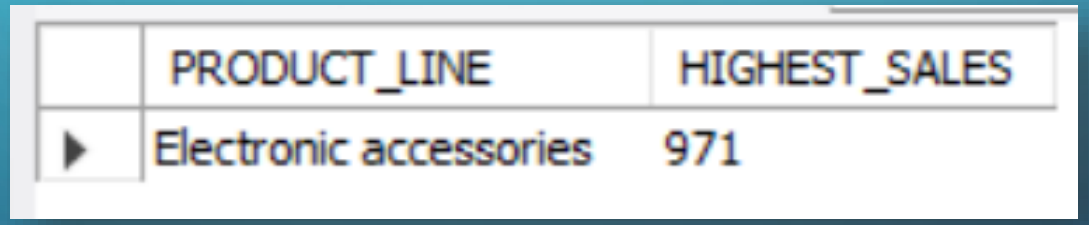
	PAYMENT	PAYMENT_COUNT
▶	Ewallet	345

5) WHICH PRODUCT LINE HAS THE HIGHEST SALES?

QUERY :-

```
SELECT PRODUCT_LINE ,SUM(QUANTITY) AS HIGHEST_SALES
FROM AMAZON
GROUP BY PRODUCT_LINE
ORDER BY HIGHEST_SALES DESC
LIMIT 1;
```

OUTPUT :-



	PRODUCT_LINE	HIGHEST_SALES
▶	Electronic accessories	971

6) HOW MUCH REVENUE IS GENERATED EACH MONTH?

QUERY :-

```
SELECT MONTH_NAME AS MONTH ,ROUND(SUM(TOTAL)) AS "TOTAL REVENUE"
FROM AMAZON
GROUP BY MONTH_NAME;
```

OUTPUT :-

MONTH	TOTAL REVENUE
January	116292
March	109456
February	97219

7) IN WHICH MONTH DID THE COST OF GOODS SOLD REACH ITS PEAK?

QUERY :-

```
SELECT MONTH_NAME ,ROUND(SUM(COGS)) AS COST_OF_GOODS FROM AMAZON
GROUP BY MONTH_NAME
ORDER BY COST_OF_GOODS DESC
LIMIT 1;
```

OUTPUT :-

	MONTH_NAME	COST_OF_GOODS
▶	January	110754

8) WHICH PRODUCT LINE GENERATED THE HIGHEST REVENUE?

QUERY :-

```
SELECT PRODUCT_LINE ,ROUND(SUM(TOTAL)) AS TOTAL FROM AMAZON
GROUP BY PRODUCT_LINE
ORDER BY TOTAL DESC
LIMIT 1;
```

OUTPUT :-

	PRODUCT_LINE	TOTAL
▶	Food and beverages	56145

9) IN WHICH CITY WAS THE HIGHEST REVENUE RECORDED?

QUERY :-

```
SELECT CITY , ROUND(SUM(TOTAL)) AS TOTAL FROM AMAZON
GROUP BY CITY
ORDER BY TOTAL DESC
LIMIT 1;
```

OUTPUT :-

	PRODUCT_LINE	TOTAL
▶	Food and beverages	56145

10) WHICH PRODUCT LINE INCURRED THE HIGHEST VALUE ADDED TAX?

QUERY :-

```
SELECT PRODUCT_LINE ,ROUND(SUM(TAX)) AS TAX
FROM AMAZON
GROUP BY PRODUCT_LINE
ORDER BY TAX DESC
LIMIT 1;
```

OUTPUT :-

	PRODUCT_LINE	TAX
▶	Food and beverages	2674

11) FOR EACH PRODUCT LINE, ADD A COLUMN INDICATING "GOOD" IF ITS SALES ARE ABOVE AVERAGE, OTHERWISE "BAD."

QUERY :-

```
SELECT ROUND(AVG(TOTAL)) FROM AMAZON;
select PRODUCT_LINE,
(case
    when 323 < ROUND(AVG(TOTAL)) then "good"
        ELSE "bad"
    end) as indicating
from AMAZON
GROUP BY PRODUCT_LINE;

ALTER TABLE AMAZON ADD COLUMN SALES varchar(20);

UPDATE AMAZON
    SET SALES = (case
    when 323 < TOTAL then "good"
    ELSE "bad"
    end);
```

OUTPUT :-

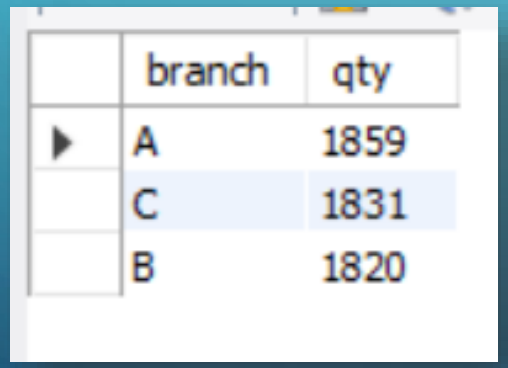
PRODUCT_LINE	indicating
Health and beauty	good
Electronic accessories	bad
Home and lifestyle	good
Sports and travel	good
Food and beverages	bad
Fashion accessories	bad

12) IDENTIFY THE BRANCH THAT EXCEEDED THE AVERAGE NUMBER OF PRODUCTS SOLD.

QUERY :-

```
select branch ,SUM(QUANTITY) FROM AMAZON
WHERE QUANTITY > (SELECT avg(QUANTITY) FROM AMAZON)
GROUP BY BRANCH;
```

OUTPUT :-



	branch	qty
▶	A	1859
	C	1831
	B	1820

13) WHICH PRODUCT LINE IS MOST FREQUENTLY ASSOCIATED WITH EACH GENDER?

QUERY :-

```
SELECT  GENDER,PRODUCT_LINE,COUNT(GENDER) AS GENDER_COUNT
FROM AMAZON
GROUP BY GENDER,PRODUCT_LINE
ORDER BY GENDER_COUNT DESC;
```

OUTPUT :-

gender	product_line	total_count
Female	Fashion accessories	96
Female	Food and beverages	90
Male	Health and beauty	88
Female	Sports and travel	88
Male	Electronic accessories	86
Female	Electronic accessories	84

14) CALCULATE THE AVERAGE RATING FOR EACH PRODUCT LINE.

QUERY :-

```
SELECT PRODUCT_LINE ,ROUND(AVG(RATING),2) AS "AVG RATING"
FROM AMAZON
GROUP BY PRODUCT_LINE;
```

OUTPUT :-

PRODUCT_LINE	AVG RATING
Health and beauty	7
Electronic accessories	6.92
Home and lifestyle	6.84
Sports and travel	6.92
Food and beverages	7.11

15) COUNT THE SALES OCCURRENCES FOR EACH TIME OF DAY ON EVERY WEEKDAY.

QUERY :-

```
SELECT DAY_NAME ,TIME_OF_DAY ,COUNT(*) AS SALES
FROM AMAZON
GROUP BY DAY_NAME,TIME_OF_DAY;
```

OUTPUT :-

DAY_NAME	TIME_OF_DAY	SALES
Saturday	Afternoon	55
Friday	Morning	29
Sunday	Afternoon	53
Sunday	Evening	58
Monday	Evening	56
Monday	Afternoon	48

16) IDENTIFY THE CUSTOMER TYPE CONTRIBUTING THE HIGHEST REVENUE.

QUERY :-

```
SELECT CUSTOMER_TYPE ,ROUND(SUM(TOTAL),2) AS REVENUE
FROM AMAZON
GROUP BY CUSTOMER_TYPE
ORDER BY REVENUE
LIMIT 1;
```

OUTPUT :-

	CUSTOMER_TYPE	REVENUE
▶	Normal	158743.31

17) DETERMINE THE CITY WITH THE HIGHEST VAT PERCENTAGE.

QUERY :-

```
SELECT CITY ,ROUND(SUM(TAX),2) AS VAT
FROM AMAZON
GROUP BY CITY
ORDER BY VAT DESC
LIMIT 1;
```

OUTPUT :-

	CITY	VAT
▶	Naypyitaw	5265.18

18) IDENTIFY THE CUSTOMER TYPE WITH THE HIGHEST VAT PAYMENTS.

QUERY :-

```
SELECT CUSTOMER_TYPE ,ROUND(SUM(TAX),2) AS VAT
FROM AMAZON
GROUP BY CUSTOMER_TYPE
ORDER BY CUSTOMER_TYPE DESC
LIMIT 1;
```

OUTPUT :-

	CUSTOMER_TYPE	VAT
▶	Normal	7559.21

19) WHAT IS THE COUNT OF DISTINCT CUSTOMER TYPES IN THE DATASET?

QUERY :-

```
SELECT COUNT(DISTINCT(CUSTOMER_TYPE))
AS "DISTINCT CUSTOMER TYPE"
FROM AMAZON;
```

OUTPUT :-

	DISTINCT CUSTOMER TYPE
▶	2

20) WHAT IS THE COUNT OF DISTINCT PAYMENT METHODS IN THE DATASET?

QUERY :-

```
SELECT COUNT(DISTINCT(PAYMENT))
AS "PAYMENT METHOD"
FROM AMAZON;
```

OUTPUT :-

	PAYMENT METHOD
▶	3

21) WHICH CUSTOMER TYPE OCCURS MOST FREQUENTLY?

QUERY :-

```
SELECT GENDER , COUNT(GENDER) AS CUSTOMER
FROM AMAZON
GROUP BY GENDER
ORDER BY CUSTOMER DESC
LIMIT 1;
```

OUTPUT :-

	GENDER	CUSTOMER
▶	Female	501

22) IDENTIFY THE CUSTOMER TYPE WITH THE HIGHEST PURCHASE FREQUENCY.

QUERY :-

```
SELECT CUSTOMER_TYPE , ROUND(SUM(TOTAL),2) AS "HIGHEST PURCHASE"
FROM AMAZON
GROUP BY CUSTOMER_TYPE
ORDER BY SUM(TOTAL) DESC
LIMIT 1;
```

OUTPUT :-

	CUSTOMER_TYPE	HIGHEST PURCHASE
▶	Member	164223.44

23) DETERMINE THE PREDOMINANT GENDER AMONG CUSTOMERS.

QUERY :-

```
SELECT GENDER,COUNT(GENDER) AS "GENDER COUNT"
FROM AMAZON
GROUP BY GENDER
ORDER BY COUNT(GENDER) DESC
LIMIT 1;
```

OUTPUT :-

	GENDER	GENDER COUNT
▶	Female	501

24) EXAMINE THE DISTRIBUTION OF GENDERS WITHIN EACH BRANCH.

QUERY :-

```
SELECT BRANCH ,GENDER,COUNT(GENDER) AS "GENDER COUNT"
FROM AMAZON
GROUP BY BRANCH,GENDER
ORDER BY BRANCH;
```

OUTPUT :-

BRANCH	GENDER	GENDER COUNT
A	Female	161
A	Male	179
B	Female	162
B	Male	170
C	Female	178
C	Male	150

25) IDENTIFY THE TIME OF DAY WHEN CUSTOMERS PROVIDE THE MOST RATINGS.

QUERY :-

```
SELECT TIME_OF_DAY ,COUNT(RATING) AS "RATING COUNT"
FROM AMAZON
GROUP BY TIME_OF_DAY
ORDER BY COUNT(RATING) DESC
LIMIT 1;
```

OUTPUT :-

	TIME_OF_DAY	RATING COUNT
▶	Evening	432

26) DETERMINE THE TIME OF DAY WITH THE HIGHEST CUSTOMER RATINGS FOR EACH BRANCH.

QUERY :-

```
SELECT TIME_OF_DAY,BRANCH ,COUNT(RATING) AS RATING_COUNT FROM AMAZON
WHERE BRANCH = "A"
GROUP BY TIME_OF_DAY,BRANCH
ORDER BY RATING_COUNT DESC
LIMIT 1;
SELECT TIME_OF_DAY,BRANCH ,COUNT(RATING) AS RATING_COUNT FROM AMAZON
WHERE BRANCH = "B"
GROUP BY TIME_OF_DAY,BRANCH
ORDER BY RATING_COUNT DESC
LIMIT 1;
SELECT TIME_OF_DAY,BRANCH ,COUNT(RATING) AS RATING_COUNT FROM AMAZON
WHERE BRANCH = "C"
GROUP BY TIME_OF_DAY,BRANCH
ORDER BY RATING_COUNT DESC
LIMIT 1;
```

OUTPUT :-

	TIME_OF_DAY	BRANCH	RATING_COUNT
▶	Evening	A	141

	TIME_OF_DAY	BRANCH	RATING_COUNT
▶	Evening	B	148

	TIME_OF_DAY	BRANCH	RATING_COUNT
▶	Evening	C	143

27) IDENTIFY THE DAY OF THE WEEK WITH THE HIGHEST AVERAGE RATINGS.

QUERY :-

```
SELECT DAY_NAME AS "DAY OF THE WEEK",
ROUND(AVG(RATING),2) AS AVG_RATING FROM AMAZON
GROUP BY DAY_NAME
ORDER BY AVG_RATING DESC
LIMIT 1;
```

OUTPUT :-

	DAY OF THE WEEK	AVG_RATING
▶	Monday	7.15

28) DETERMINE THE DAY OF THE WEEK WITH THE HIGHEST AVERAGE RATINGS FOR EACH BRANCH.

QUERY :-

```
SELECT BRANCH ,DAY_NAME AS "DAY OF THE WEEK",
ROUND(AVG(RATING),2) AS AVG_RATING FROM AMAZON
WHERE BRANCH = "A"
GROUP BY BRANCH ,DAY_NAME
ORDER BY AVG_RATING DESC
LIMIT 1;
SELECT BRANCH ,DAY_NAME AS "DAY OF THE WEEK",
ROUND(AVG(RATING),2) AS AVG_RATING FROM AMAZON
WHERE BRANCH = "B"
GROUP BY BRANCH ,DAY_NAME
ORDER BY AVG_RATING DESC
LIMIT 1;
SELECT BRANCH ,DAY_NAME AS "DAY OF THE WEEK",
ROUND(AVG(RATING),2) AS AVG_RATING FROM AMAZON
WHERE BRANCH = "C"
GROUP BY BRANCH ,DAY_NAME
ORDER BY AVG_RATING DESC
LIMIT 1;
```

OUTPUT :-

	BRANCH	DAY OF THE WEEK	AVG_RATING
▶	A	Friday	7.31

	BRANCH	DAY OF THE WEEK	AVG_RATING
▶	B	Monday	7.34

	BRANCH	DAY OF THE WEEK	AVG_RATING
▶	C	Friday	7.28

THANK YOU

SURYAKANT RANJAN