```sql
-- AMAZON SALES CAPSTONE PROJECTS
-- Create database
CREATE DATABASE amazon_sales;
USE amazon_sales;

-- Create table with correct schema based on your CSV
CREATE TABLE sales (
    invoice_id VARCHAR(30) 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,
    tax_5 DECIMAL(10,2) NOT NULL,
    total DECIMAL(10,2) NOT NULL,
    date DATE NOT NULL,
    time TIME NOT NULL,
    payment VARCHAR(30) NOT NULL,
    cogs DECIMAL(10,2) NOT NULL,
    gross_margin_percentage FLOAT NOT NULL,
    gross_income DECIMAL(10,2) NOT NULL,
    rating FLOAT(2,1) NOT NULL
);
drop table sales;

create table amazon_sales
(invoice_id varchar(30) primary key not null,
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 not null,
total decimal(10,2) not null,
date date not null,
time time not null,
payment_method varchar(20) not null,
cogs decimal(10,2) not null,
gross_margin_percentage float not null,
gross_income decimal(10,2) not null,
rating decimal(3,1) not null);

-- Add new columns
ALTER TABLE amazon_sales ADD COLUMN timeofday VARCHAR(20);
ALTER TABLE amazon_sales ADD COLUMN dayname VARCHAR(20);
ALTER TABLE amazon_sales ADD COLUMN monthname VARCHAR(20);

-- Add new columns
ALTER TABLE sales ADD COLUMN timeofday VARCHAR(20);
ALTER TABLE sales ADD COLUMN dayname VARCHAR(20);
ALTER TABLE sales ADD COLUMN monthname VARCHAR(20);

-- Fill timeofday
UPDATE amazon_sales
```

```
SET timeofday = CASE
    WHEN HOUR(time) BETWEEN 5 AND 11 THEN 'Morning'
    WHEN HOUR(time) BETWEEN 12 AND 16 THEN 'Afternoon'
    WHEN HOUR(time) BETWEEN 17 AND 21 THEN 'Evening'
    ELSE 'Night'
END;

-- Fill dayname
UPDATE amazon_sales
SET dayname = DAYNAME(date);

-- Fill monthname
UPDATE amazon_sales
SET monthname = MONTHNAME(date);

-- Business Questions (SQL Queries)
--   1 What is the count of distinct cities in the dataset?
SELECT COUNT(DISTINCT city) AS distinct_cities
FROM amazon_sales;

-- 2 For each branch, what is the corresponding city?
SELECT branch, city
FROM amazon_sales GROUP BY branch, city;

-- 3 What is the count of distinct product lines in the dataset?
SELECT COUNT(DISTINCT product_line) AS distinct_product_lines
FROM amazon_sales;

-- 4 Which payment method occurs most frequently?
SELECT payment_method, COUNT(*) AS frequency
FROM amazon_sales
GROUP BY payment_method
ORDER BY frequency DESC

-- 5 Which product line has the highest sales?
SELECT product_line, SUM(quantity) AS revenue
FROM amazon_sales
GROUP BY product_line
ORDER BY revenue DESC

-- 6 How much revenue is generated each month?
SELECT monthname, SUM(total) AS revenue
FROM amazon_sales
GROUP BY monthname
ORDER BY
FIELD(monthname,'January','February','March','April','May','June',
 'July','August','September','October','November','December');

-- 7 In which month did the cost of goods sold reach its peak?
SELECT monthname, SUM(cogs) AS total_cogs
FROM amazon_sales
GROUP BY monthname
ORDER BY total_cogs DESC

-- 8 Which product line generated the highest revenue?
SELECT product_line, SUM(total) AS revenue
FROM amazon_sales
GROUP BY product_line
```

```
ORDER BY revenue DESC

-- 9 In which city was the highest revenue recorded?
SELECT city, SUM(total) AS revenue
FROM amazon_sales
GROUP BY city
ORDER BY revenue DESC

-- 10 Which product line incurred the highest Value Added Tax?
SELECT product_line, SUM(vat) AS vat
FROM amazon_sales
GROUP BY product_line
ORDER BY vat DESC

-- 11 For each product line, add a column indicating "Good" if its
sales are above average, otherwise "Bad."
WITH product_sales AS (
  SELECT product_line, SUM(total) AS revenue
  FROM amazon_sales
  GROUP BY product_line
),
avg_sales AS (
SELECT AVG(revenue) AS avg_revenue FROM product_sales
)
SELECT p.product_line, p.revenue,
CASE WHEN p.revenue > a.avg_revenue THEN 'Good' ELSE 'Bad' END AS
performance
FROM product_sales p CROSS JOIN avg_sales a;

-- 12 Identify the branch that exceeded the average number of products
sold.
WITH branch_qty AS (
SELECT branch, SUM(quantity) AS total_qty
FROM  amazon_sales
GROUP BY branch
),
avg_qty AS (
SELECT AVG(total_qty) AS avg_qty
FROM branch_qty
)
SELECT branch, total_qty
FROM branch_qty, avg_qty
WHERE branch_qty.total_qty > avg_qty.avg_qty;

-- 13 Which product line is most frequently associated with each
gender?
SELECT gender, product_line, COUNT(*) AS freq
FROM  amazon_sales
GROUP BY gender, product_line
HAVING COUNT(*) = (
SELECT MAX(c) FROM (
SELECT gender g, product_line pl, COUNT(*) c
FROM  amazon_sales GROUP BY gender, product_line
) t WHERE t.g =  amazon_sales.gender
);

-- 14 Calculate the average rating for each product line.
SELECT product_line, AVG(rating) AS avg_rating
```

```
FROM amazon_sales
GROUP BY product_line
ORDER BY avg_rating DESC;

-- 15 Count the sales occurrences for each time of day on every
weekday.
SELECT dayname, timeofday, COUNT(*) AS sales_count
FROM amazon_sales
GROUP BY dayname, timeofday
ORDER BY dayname, timeofday;

-- 16 Identify the customer type contributing the highest revenue.
SELECT customer_type, SUM(total) AS revenue
FROM  amazon_sales
GROUP BY customer_type
ORDER BY revenue DESC

-- 17 Determine the city with the highest VAT percentage.
SELECT city, AVG(vat/total)*100 AS avg_vat_percent
FROM amazon_sales
GROUP BY city
ORDER BY avg_vat_percent DESC

-- 18 Identify the customer type with the highest VAT payments.
SELECT customer_type, SUM(vat) AS total_vat
FROM amazon_sales
GROUP BY customer_type
ORDER BY total_vat DESC

-- 19 What is the count of distinct customer types in the dataset?
SELECT COUNT(DISTINCT customer_type)
FROM  amazon_sales;

-- 20 What is the count of distinct payment methods in the dataset?
SELECT COUNT(DISTINCT payment_method)
FROM amazon_sales;

-- 21 Which customer type occurs most frequently?
SELECT customer_type, COUNT(*) AS freq
FROM amazon_sales
GROUP BY customer_type
ORDER BY freq DESC

-- 22 Identify the customer type with the highest purchase frequency.
SELECT customer_type, COUNT(*) AS tx_count
FROM amazon_sales
GROUP BY customer_type
ORDER BY tx_count DESC

-- 23 Determine the predominant gender among customers.
SELECT gender, COUNT(*) AS freq
FROM amazon_sales
GROUP BY gender
ORDER BY freq DESC

-- 24 Examine the distribution of genders within each branch.
SELECT branch, gender, COUNT(*) AS count
FROM amazon_sales
```

```
GROUP BY branch, gender
ORDER BY branch, count DESC;

-- 25 Identify the time of day when customers provide the most ratings.
SELECT timeofday, COUNT(rating) AS rating_count
FROM amazon_sales
GROUP BY timeofday
ORDER BY rating_count DESC

-- 26 Determine the time of day with the highest customer ratings for
each branch.
SELECT branch, timeofday, AVG(rating) AS avg_rating
FROM amazon_sales
GROUP BY branch, timeofday
HAVING AVG(rating) = (
SELECT MAX(avg_r) FROM (
SELECT branch b, timeofday t, AVG(rating) avg_r
FROM amazon_sales GROUP BY branch, timeofday
) sub WHERE sub.b = amazon_sales.branch
);

-- 27 Identify the day of the week with the highest average ratings.
SELECT dayname, AVG(rating) AS avg_rating
FROM amazon_sales
GROUP BY dayname
ORDER BY avg_rating DESC

-- 28 Determine the day of the week with the highest average ratings
for each branch.
SELECT branch, dayname, AVG(rating) AS avg_rating
FROM amazon_sales
GROUP BY branch, dayname
HAVING AVG(rating) = (
SELECT MAX(avg_r) FROM (
SELECT branch b, dayname d, AVG(rating) avg_r
FROM amazon_sales GROUP BY branch, dayname
) sub WHERE sub.b = amazon_sales.branch
);

-- key finding from amazon sales dataset
 ### Product Analysis ###

 -- Highest Sales Product Line : Electronic accessories  -- 971 unit
solds

 -- Highest Revenue Product Line: food and beverages -- 56144,96

-- Lowest Sales Product Line: health and beauty - 854 units sold

-- Lowest Revenue Product Line : health and beauty --49193.84

#### Sales Analysis: ####

-- Month With Highest Revenue: january -- 116292.11

-- City & Branch With Highest Revenue: city:naypyitaw branch :c -
110568.86
```

-- Month With Lowest Revenue: february 97219.58

-- City & Branch With Lowest Revenue: city:mandalay branch:b - 106198.00

-- Peak Sales Time Of Day: Afternoon

-- Peak Sales Day Of Week: Saturday

Customer Analysis:

-- Most Predominant Gender: Female -501

-- Most Predominant Customer Type: Member - 501

-- Highest Revenue Gender: Female --167883.26

-- Highest Revenue Customer Type: member - 164223.81

-- Most Popular Product Line (Male): Health and beauty - 88

-- Most Popular Product Line (Female): Fashion Accessories - 96

-- Distribution Of Members Based On Gender: female -261 & male -240

-- Sales Male: - 2641 units

-- Sales Female: - 2869 units