```
CREATE DATABASE smartphones_analysis;
USE smartphones_analysis;
SELECT * FROM smartphone;

#1. What is the total transaction amount for each brand?

SELECT Brands, SUM(Price) AS Total_Amount
FROM smartphone
GROUP BY Brands, Price ORDER BY price DESC;


#2. Identify the top 3 most expensive phones for each operating system
type?

SELECT *
FROM (
   SELECT Phone_name, Operating_System_Type, Price,
          ROW_NUMBER() OVER (PARTITION BY Operating_System_Type ORDER BY
Price DESC) AS rankk
   FROM smartphone
) ranked_phones
WHERE rankk <= 3;


#3. Find the percentage of phones with 128 GB internal storage that
support 5G?

SELECT
   ROUND((COUNT(CASE WHEN Internal_Storage = '128 GB' AND 5G_Availability
= 'Yes' THEN 1 END) * 100.0 / COUNT(*)), 2)
   AS Percentage_5G
FROM smartphone


#4. Find the brand with the highest average battery capacity, but only for
phones that have a selfie camera resolution greater than 16MP?

SELECT Brands, Round(AVG(Battery_Capacity),2) AS Avg_Battery_Capacity
FROM smartphone
WHERE Selfie_Camera > 16
GROUP BY Brands
ORDER BY Avg_Battery_Capacity DESC
LIMIT 1;


#5.Identify brands that sell phones in most of the price range?

SELECT Brands, COUNT(DISTINCT Price_Range) AS Price_Range_Count
FROM smartphone
GROUP BY Brands
HAVING COUNT(DISTINCT Price_Range) >= (
   SELECT MAX(Price_Range_Count)
   FROM (
       SELECT Brands, COUNT(DISTINCT Price_Range) AS Price_Range_Count
```

```
       FROM smartphone
       GROUP BY Brands
   ) AS Brand_Price_Ranges
) - 1;
```

#6. For each brand, calculate the total number of phones and the average price, but only include brands that have more than 10 models?

```
SELECT Brands, COUNT(*) AS Total_Phones, Round(AVG(Price),2) AS Avg_Price
FROM smartphone
GROUP BY Brands
HAVING COUNT(*) > 10;
```

#7. Find the phone with the highest price-to-battery capacity ratio for each country of origin?

```
SELECT *
FROM (
   SELECT Phone_name, Country_of_Origin,
          ROUND(Price / Battery_Capacity, 2) AS Price_Battery_Ratio,
          ROW_NUMBER() OVER (PARTITION BY Country_of_Origin ORDER BY Price
/ Battery_Capacity DESC) AS rankk
   FROM smartphone
) ranked_phones
WHERE rankk = 1;
```

#8. Calculate the cumulative total of phones sold by each brand, ordered by the brand's total mobile count?

```
SELECT Brands, Total_Mobile,
       SUM(Total_Mobile) OVER (ORDER BY Total_Mobile DESC) AS
Cumulative_Total
FROM smartphone
ORDER BY Total_Mobile DESC;
```

#9. Find the total number of phones per country and the average price per country, but exclude countries that have fewer than 5 phone models?

```
SELECT Country_of_Origin, COUNT(*) AS Total_Phones, Round(AVG(Price),2) AS
Avg_Price
FROM smartphone
GROUP BY Country_of_Origin
HAVING COUNT(*) >= 5;
```

#10. Determine the most common USB type for each price range?

```
SELECT Price_Range, USB_Type
FROM (
   SELECT Price_Range, USB_Type,
```

```
          ROW_NUMBER() OVER (PARTITION BY Price_Range ORDER BY COUNT(*)
DESC) AS rankk
   FROM smartphone
   GROUP BY Price_Range, USB_Type
) ranked_usb
WHERE rankk = 1;
```

#11. Identify phones that are priced above the average price of phones with the same battery capacity range?

```
SELECT p.Phone_name, p.Price, p.Battery_Capacity_Range
FROM smartphone p
JOIN (
   SELECT Battery_Capacity_Range, AVG(Price) AS Avg_Price
   FROM smartphone
   GROUP BY Battery_Capacity_Range
) avg_price ON p.Battery_Capacity_Range = avg_price.Battery_Capacity_Range
WHERE p.Price > avg_price.Avg_Price;
```

#12. Compare the average price of phones with and without 5G across different price ranges using a view?

```
CREATE VIEW Avg_Price_5G_Comparison AS
SELECT Price_Range,
       AVG(CASE WHEN 5G_Availability = 'Yes' THEN Price ELSE NULL END) AS
Avg_Price_5G,
       AVG(CASE WHEN 5G_Availability = 'No' THEN Price ELSE NULL END) AS
Avg_Price_No_5G
FROM smartphone
GROUP BY Price_Range;
```

#13. Calculate the percentage of phones with high battery capacity per brand using a temporary table?

```
CREATE TEMPORARY TABLE HighBatteryPhones AS
SELECT Brands, COUNT(*) AS Total_Phones,
       COUNT(CASE WHEN Battery_Capacity > 5000 THEN 1 END) AS
High_Battery_Phones
FROM smartphone
GROUP BY Brands;
SELECT Brands,
       Round((High_Battery_Phones * 100.0 / Total_Phones),2) AS
High_Battery_Percentage
FROM HighBatteryPhones;
```

#14. Identify brands that offer the most 5G phones using a CTE and DENSE_RANK?

```
WITH FiveGPhones AS (
    SELECT Brands, COUNT(*) AS FiveG_Count
```

```
    FROM smartphone
    WHERE 5G_Availability = 'Yes'
    GROUP BY Brands
)
SELECT Brands, FiveG_Count,
       DENSE_RANK() OVER (ORDER BY FiveG_Count DESC) AS Brand_Rank
FROM FiveGPhones;
```

#15. Calculate the median price of phones per brand using a CTE?

```
WITH BrandPrices AS (
    SELECT Brands, Price,
           ROW_NUMBER() OVER (PARTITION BY Brands ORDER BY Price) AS
RowNum,
           COUNT(*) OVER (PARTITION BY Brands) AS Total_Count
    FROM smartphone
)
SELECT Brands, ROUND(AVG(Price), 2) AS Median_Price
FROM BrandPrices
WHERE RowNum IN (FLOOR(Total_Count / 2), CEIL(Total_Count / 2 + 1))
GROUP BY Brands;
```

#16. Identify the most common USB type per country using a CTE and DENSE_RANK?

```
WITH USBCount AS (
    SELECT Country_of_Origin, USB_Type, COUNT(*) AS USB_Type_Count
    FROM smartphone
    GROUP BY Country_of_Origin, USB_Type
)
SELECT Country_of_Origin, USB_Type, USB_Type_Count
FROM (
    SELECT Country_of_Origin, USB_Type, USB_Type_Count,
           DENSE_RANK() OVER (PARTITION BY Country_of_Origin ORDER BY
USB_Type_Count DESC) AS USB_Rank
    FROM USBCount
) RankedUSBs
WHERE USB_Rank = 1;
```