```sql
--1)Which customers placed the most orders
select top 5 concat(C.FIRSTNAME,' ',C.LASTNAME) as fullname,F.CUSTOMERID , count(*)as totalorder
from DIMCUSTOMERdata C inner join FACT F
on C.CUSTOMERID=F.CUSTOMERID
group by C.FIRSTNAME,C.LASTNAME,F.CUSTOMERID
order by totalorder DESC;
```

Results **Chart**

	A FULLNAME	# CUSTOMERID	# TOTALORDER
1	Tony Thomas	965	68
2	Gregory Thomas	360	67
3	Frank Aguilar	341	67
4	Shannon Garcia	891	67
5	Sandra Wilson	869	67

Query Details ...

Query duration   2.7s

Rows   5

Query ID   01bcebd0-0000-8af5-0...

Show more ⌄

```sql
--2)What is the average customer spend per order?
SELECT
    AVG(f.totalamount) AS avg_spend_per_order
FROM
    FACT f
JOIN
    dimcustomerdata c ON f.customerid = c.customerid
```

Results **Chart**

	# AVG_SPEND_PER_ORDER
1	450.12273653430649

Query Details ...

Query duration   53ms

Rows   1

Query ID   01bcebd8-0000-8af5-0...

```sql
--3)Determines how much customers typically spend.
SELECT
    C.FIRSTNAME,
    F.CUSTOMERID,
    AVG(TOTALAMOUNT) AS AvgSpendPerCustomer
FROM FACT F
JOIN DIMCUSTOMERDATA C ON F.CUSTOMERID = C.CUSTOMERID
GROUP BY F.CUSTOMERID , C.FIRSTNAME
ORDER BY AvgSpendPerCustomer DESC;
```

Results **Chart**

	A FIRSTNAME	# CUSTOMERID	# AVGSPENDPERCUSTOMER
1	Cory	223	557.22486141757042
2	Bradley	129	544.66492734258713
3	Rebecca	812	533.86348164050119
4	Michael	732	531.17348494942660
5	Kristen	581	526.54563803812943

Query Details ...

Query duration   32ms

Rows   1K

Query ID   01bcebdb-0000-8af4-0...

Show more ⌄

```sql
SELECT
    dc.FIRSTNAME,
    COUNT(*) AS OrderCount
FROM FACT f
JOIN DIMCUSTOMERDATA dc ON f.CUSTOMERID = dc.CUSTOMERID
GROUP BY f.CUSTOMERID,dc.FIRSTNAME
HAVING COUNT(*) = 1;
```

Results **Chart**

FIRSTNAME	ORDERCOUNT

Query produced no results

Query Details ...

Query duration   1.3s

Rows   0

Query ID   01bcebdf-0000-8af5-0...

Show more ⌄

```
26    //4)Which customers have made only one purchase?
27    SELECT
28        dc.FIRSTNAME,
29        COUNT(*) AS OrderCount
30    FROM FACT f
31    JOIN DIMCUSTOMERDATA dc ON f.CUSTOMERID = dc.CUSTOMERID
32    GROUP BY f.CUSTOMERID,dc.FIRSTNAME
33    HAVING COUNT(*) = 1;
34
```

Results | **Chart**

FIRSTNAME	ORDERCOUNT
Query produced no results	

Query Details

Query duration	1.3s
Rows	0
Query ID	01bcebdf-0000-8af5-0...

Show more ⌄

```
34
35    //5)Who are the most loyal customers based on repeat purchases?
36    SELECT
37        dc.FIRSTNAME,COUNT(*) AS TotalOrders
38    FROM FACT f
39    JOIN DIMCUSTOMERDATA dc ON f.CUSTOMERID = dc.CUSTOMERID
40    GROUP BY dc.FIRSTNAME
41    HAVING COUNT(*) > 1
42    ORDER BY TotalOrders desc
43
44
```

Results | **Chart**

	A FIRSTNAME	# TOTALORDERS
1	Michael	1153
2	James	875
3	David	840
4	Jennifer	774
5	Robert	632

Query Details

Query duration	83ms
Rows	355
Query ID	01bcebe1-0000-8af5-0...

Show more ⌄

```
35    //5)Who are the most loyal customers based on repeat purchases?
36    SELECT
37        dc.FIRSTNAME,COUNT(*) AS TotalOrders
38    FROM FACT f
39    JOIN DIMCUSTOMERDATA dc ON f.CUSTOMERID = dc.CUSTOMERID
40    join dimloyaltyinfo dl on dc.loyaltyprogramid=dl.loyaltyprogramid
41    where dl.loyaltyprogramid=5
42    GROUP BY f.CUSTOMERID ,dc.FIRSTNAME
43    ORDER BY TotalOrders desc
44    limit 5
45
```

Results | **Chart**

	A FIRSTNAME	# TOTALORDERS
1	Sandra	67
2	Frank	67
3	Michael	64
4	Rachel	64
5	Michael	63

Query Details

Query duration	169ms
Rows	5
Query ID	01bcebeb-0000-8af5-0...

Show more ⌄

```sql
47    --6) What are the top-selling products?
48
49    SELECT
50        DP.ProductName,
51        SUM(F.QUANTITYORDERDED) AS TotalQuantitySold
52    FROM
53        FACT F
54    JOIN
55        DIMPRODUCTDATA DP ON F.ProductID = DP.ProductID
56    GROUP BY
57        DP.ProductName
58    ORDER BY
59        TotalQuantitySold DESC
60    LIMIT 5;
61
62    --7)Which products are underperforming?
```

Results | **Chart**

	A PRODUCTNAME	# TOTALQUANTITYSOLD
1	Jacket	8186
2	Wetsuit	6683
3	Suitcase	6673
4	Hair Dryer	6613
5	Dart Board	6299

Query Details

Query duration — 3.6s

Rows — 5

Query ID — 01bcebfd-0000-8af5-0...

Show more ⌄

```sql
61    --7)Which products are underperforming?
62    SELECT
63        DP.ProductName,
64        SUM(F.QUANTITYORDERDED) AS TotalQuantitySold
65    FROM
66        FACT F
67    JOIN
68        DIMPRODUCTDATA DP ON F.ProductID = DP.ProductID
69    GROUP BY
70        DP.ProductName
71    ORDER BY
72        TotalQuantitySold ASC
73    LIMIT 5;
74
```

Results | **Chart**

	A PRODUCTNAME	# TOTALQUANTITYSOLD
1	Headphones	439
2	Washing Machine	577
3	Fan	774
4	Mattress	893
5	Hammock	932

Query Details

Query duration — 138ms

Rows — 5

Query ID — 01bcebff-0000-8af5-00...

Show more ⌄

```sql
74    --8) What are the best-selling product categories?
75    SELECT
76        DP.CATEGORY,
77        SUM(F.QUANTITYORDERDED) AS TotalQuantitySold
78    FROM
79        FACT F
80    JOIN
81        DIMPRODUCTDATA DP ON F.ProductID = DP.ProductID
82    GROUP BY
83        DP.CATEGORY
84    ORDER BY
85        TotalQuantitySold DESC
86    LIMIT 1;
```

Results | **Chart**

	A CATEGORY	# TOTALQUANTITYSOLD
1	Electronics	58932

Query Details

Query duration — 122ms

Rows — 1

Query ID — 01bcec05-0000-8af5-0...

```
89
90    --10)Which product category contributes most to revenue?
91    SELECT
92    DP.CATEGORY,
93    SUM (F. TOTALAMOUNT) AS TOTALREVENUE
94    FROM
95    FACT F
96    JOIN DIMPRODUCTDATA DP ON F.ProductID = DP.ProductID
97    GROUP BY
98    DP.CATEGORY
99    ORDER BY
100   TOTALREVENUE DESC
101   LIMIT 1;
```

↳ **Results**   ∿ Chart

A CATEGORY	# TOTALREVENUE
1 Electronics	2518072.74531621792326

```
102   --11) Which store has the highest revenue?
103   SELECT
104   DS. STORENAME,
105   SUM (F. TOTALAMOUNT) AS TOTALREVENUE
106   FROM
107   FACT F
108   JOIN dimstoredata DS ON F.STOREID = DS.STOREID
109   GROUP BY
110   DS. STORENAME
111   ORDER BY
112   TOTALREVENUE DESC
113   LIMIT 1;
114
115   --12)What is the average transaction value per store?
```

↳ **Results**   ∿ Chart

A STORENAME	# TOTALREVENUE	Query Details ⋯
1 The Chic Quest	420968.73288672652274	Query duration 406ms
		Rows 1
		Query ID 01bcec18-0000-8af5-0...
		Show more ⌄

```
115   --12)What is the average transaction value per store?
116   SELECT
117   DS. STORENAME,
118   AVG (F. TOTALAMOUNT) average_transaction
119   FROM
120   FACT F
121   JOIN dimstoredata DS ON F.STOREID = DS.STOREID
122   GROUP BY
123   DS. STORENAME
124   ORDER BY
125   average_transaction DESC
126   LIMIT 1;
```

↳ **Results**   ∿ Chart

A STORENAME	# AVERAGE_TRANSACTION
1 The Purely Merchant	473.80849186988175

```sql
127
128     --13) Which region generates the most revenue?
129      SELECT
130     DS.REGION,
131     SUM (F. TOTALAMOUNT) TOTALREVENUE
132     FROM
133      FACT F
134     JOIN dimstoredata DS ON F.STOREID = DS.STOREID
135     GROUP BY
136     DS.REGION
137     ORDER BY
138     TOTALREVENUE DESC
139      LIMIT 1;
```

⤷ Results   **∼ Chart**

REGION	TOTALREVENUE
North	6431829.15094446592412

```sql
141     --14)What is the profit margin per region?
142      SELECT
143     DS.REGION,
144     SUM (F. TOTALAMOUNT-F.SHIPPINGCOST) profit_margin
145     FROM
146      FACT F
147     JOIN dimstoredata DS ON F.STOREID = DS.STOREID
148     GROUP BY
149     DS.REGION
150     ORDER BY
151     profit_margin DESC
152      LIMIT 4;
```

⤷ Results   **∼ Chart**

REGION	PROFIT_MARGIN
North	5640483.7517226115696775
East	5283108.1050453242008957
West	4161807.8046130562825224
South	3731874.1645355483137925

```sql
154     --15)Monthly revenue trends over the last year
155     SELECT
156     DD. YEAR,
157     DD. MONTH,
158     SUM(F. TOTALAMOUNT) AS MonthlyRevenue
159     FROM FACT F
160     JOIN DIMDATE DD ON F.DATEID = DD.DATEID
161     WHERE
162     DD. YEAR = 2024
163     GROUP BY
164     DD. YEAR,
165     DD. MONTH
166     ORDER BY
167     DD.MONTH;
168     --16)Year-over-Year Sales Growth
```

⤷ Results   **∼ Chart**

YEAR	MONTH	MONTHLYREVENUE
2024	1	181104.74458264309442
2024	2	155330.68399219274780
2024	3	179125.84581944352517
2024	4	159672.39595151037965
2024	5	167348.23975519632383
2024	6	153021.02642401351885
2024	7	152700.03419525916830

Query Details ···

Query duration	1.0s
Rows	Z
Query ID	01bcec28-0000-8af5-0...

Show more ⌄

YEAR   #

100% filled

```
--17) Hourly sales distribution ?
--can't be calculated becuse of lake of data columns
--18) Profitability by product?
--can't be calculated becuse of lake of data columns
--19) Profit margin comparison between categories ?
--can't be calculated becuse of lake of data columns
```

```
196    --20) Age-wise purchase patterns
197    SELECT
198    case
199    when DATEDIFF ('year', DC.DOB, CURRENT_DATE()) BETWEEN 0 AND 18 THEN '0-18'
200    WHEN DATEDIFF ('year', DC.DOB, CURRENT_DATE()) BETWEEN 19 AND 25 THEN '19-25'
201    WHEN DATEDIFF ('year', DC.DOB, CURRENT_DATE()) BETWEEN 26 AND 45 THEN '26-45'
202    WHEN DATEDIFF ('year', DC.DOB, CURRENT_DATE()) BETWEEN 46 AND 60 THEN '46-40'
203    WHEN DATEDIFF ('year', DC.DOB, CURRENT_DATE()) > 60 THEN '60+' ELSE 'UNKNOWN'
204    END AS AgeGroup,
205    COUNT (F.OrderID) AS TotalOrders,
206    SUM (F. TOTALAMOUNT) AS TOTALREVENUE
207    FROM FACT F
208    JOIN DIMCUSTOMERDATA DC ON F.CustomerID = DC.CustomerID
209    GROUP BY
210    AgeGroup
```

Results | **Chart**

AGEGROUP	TOTALORDERS	TOTALREVENUE
1 0-18	15799	7103988.54766069677783
2 19-25	6539	2953739.74091477436755
3 26-45	16632	7475191.25968317941198
4 46-40	8687	3918129.58402025938786

```
213    --21. How does the yearly revenue trend compare across different years?
214    SELECT
215        DD.YEAR,
216        SUM(F.TOTALAMOUNT) AS YearlyRevenue
217    FROM FACT F
218    JOIN DIMDATE DD ON F.DATEID = DD.DATEID
219    GROUP BY DD.YEAR
220    order BY DD.YEAR
221    --22. What is the peak sales hour in a typical day?
```

Results | **Chart**

YEAR	YEARLYREVENUE
1 2014	2038234.14454108630287
2 2015	2042731.95440325523147
3 2016	2046147.02035117216021
4 2017	2033833.94590551106841
5 2018	1996634.44598978320121
6 2019	2080212.98349031928106

```
221  --22. What is the peak sales hour in a typical day?
222  --can't be calculated
223  --23. Whichstore has the highest number of transactions per day?
224  SELECT
225  DD.DATE,
226  DS. STORENAME,
227  COUNT (F. ORDERID) AS TransactionNumber
228  FROM FACT F
229  join dimstoredata DS ON F.STOREID = DS. STOREID
230  join DIMDATE DD ON F.DATEID = DD.DATEID
231  GROUP BY
232  DD.DATE,
233  DS.STORENAME
234  ORDER BY
235  TransactionNumber DESC
236  LIMIT 1;
```

Results **Chart**

	DATE	STORENAME	TRANSACTIONNUMBER
1	2016-12-19	The Stylish Deals	5

Query Details · · ·

Query duration 211ms

Rows 1

Query ID 01bcec83-0000-8c5b-0...

```
237  --24. What is the total discount amount given to customers in the last quarter?
238  SELECT
239  SUM(F.DISCOUNTAMOUNT) AS TotalDiscount
240  from FACT F
241  join DIMDATE DD ON F.DATEID = DD.DATEID
242  WHERE
243  DD.year = 2024 --(SELECT MAX(Year) FROM DIMDATE) for dynamic latest year
244  AND DD.QUARTER = (SELECT MAX(QUARTER) FROM DIMDATE WHERE YEAR = 2024);
245  --subquery for dynamic latest quarter]
246  --25. Which product has the highest return rate?
```

Results **Chart**

	TOTALDISCOUNT
1	null

Query Details · · ·

Query duration 31ms

Rows 1

Query ID 01bcec96-0000-8c5a-0...

Show more ⌄

```
--25. Which product has the highest return rate?
--can't be calculated
```

```
249  --26. How does the revenue vary across weekdays versus weekends?
250  select d.isweekend,
251  sum(f.totalamount)
252  from fact f
253  join dimDate d on f.dateid=d.dateid
254  group by d.isweekend
255
```

Results **Chart**

	ISWEEKEND	SUM(F.TOTALAMOUNT)
1	FALSE	15388883.51789068764661
2	TRUE	6062165.61438822229861

Query Details · · ·

Query duration 2.8s

Rows 2

Query ID 01bceca5-0000-8c5a-0...

Show more ⌄

```
256   --27. What is the best-selling product for each region?
257   WITH RegionalProduct AS (
258       SELECT
259           DS.REGION,
260           DP.PRODUCTNAME,
261           SUM(F.QUANTITYORDERDED) AS TotalQuantitySold,
262           ROW_NUMBER() OVER (
263               PARTITION BY DS.REGION
264               ORDER BY SUM(F.QUANTITYORDERDED) DESC
265           ) AS rn
266       FROM FACT F
267       JOIN DIMPRODUCTDATA DP ON F.ProductID = DP.ProductID
268       JOIN DIMSTOREDATA DS ON F.STOREID = DS.STOREID
269       GROUP BY DS.REGION, DP.PRODUCTNAME
270   )
271   SELECT
272       REGION,
273       PRODUCTNAME,
274       TotalQuantitySold
275   FROM RegionalProduct
276   WHERE rn = 1
277   ORDER BY REGION;
278
```

↳ Results   ∿ Chart   🔍 ▥ ↧ ◨ 🕐

	REGION	PRODUCTNAME	TOTALQUANTITYSOLD
1	East	Jacket	2369
2	North	Jacket	2437
3	South	Jacket	1408
4	West	Jacket	1972

Query Details   ⋯
Query duration   248ms
Rows   4
Query ID   01bceccd-0000-8c5b-0…

```
278
279   --28. Which customer segment generates the highest revenue?
280    SELECT
281   DLI.PROGRAMNAME,
282   SUM (F.TOTALAMOUNT) AS TOTALREVENUE
283   from FACT F
284   join DIMCUSTOMERDATA DC ON F.CustomerID = DC.CustomerID
285   join DIMLOYALTYINFO DLI ON DC.LoyaltyProgramID = DLI.LoyaltyProgramID
286   GROUP BY
287   DLI.PROGRAMNAME
288   ORDER BY
289   TOTALREVENUE DESC;
290   --29. What is the profit-to-revenue ratio for different product categories?
291   --can't be calculated
292   --30. Which product saw the highest percentage increase in sales compared to the previous year
```

↳ Results   ∿ Chart   🔍 ▥ ↧ ◨ 🕐

	PROGRAMNAME	TOTALREVENUE
1	Diamond VIP	4788776.59669616579330
2	Bronze Rewards	4589967.08838019946978
3	Platinum Elite	4273402.27855286614160
4	Gold Club	4039095.47100852136814

Query Details   ⋯
Query duration   1.9s
Rows   5
Query ID   01bcecd7-0000-8c5b-0…

```sql
-- 30) Which product saw the highest percentage increase in sales compared to the previous year?
WITH ProductYear AS (
SELECT DP.PRODUCTNAME,
DD.Year,
SUM(FO.QUANTITYORDERDED) AS YearlyQuantity
from FACT FO
JOIN DIMDATE DD ON FO.DATEID= DD.DATEID
JOIN DIMPRODUCTDATA DP ON FO.ProductID = DP.ProductID
GROUP BY
DP.PRODUCTNAME,
DD. YEAR ),
ProductPreviousYear AS (
SELECT
PRODUCTNAME,
YEAR,
YearlyQuantity,
LAG (YearlyQuantity, 1) OVER (PARTITION BY PRODUCTNAME ORDER BY YEAR) AS PreviousYearQuantity FROM
ProductYear
)
SELECT
YEAR,
PRODUCTNAME,
YearlyQuantity,
PreviousYearQuantity,
(YearlyQuantity - PreviousYearQuantity) *100.0/ (PreviousYearQuantity) AS PercentageIncrease
from ProductPreviousYear
WHERE PreviousYearQuantity IS NOT NULL AND PreviousYearQuantity != 0
ORDER BY PercentageIncrease DESC;
```

sults ~ Chart

YEAR	PRODUCTNAME	YEARLYQUANTITY	PREVIOUSYEARQUANTITY	PERCENTAGEINCREASE
2016	Mattress	147	38	286.842105
2018	Showerhead	157	52	201.923077
2018	Headphones	104	37	181.081081
2015	Headphones	58	22	163.636364
2015	Smartphone	150	59	154.237288

Query Details ...

Query duration   2.2s

Rows   1.5K

Query ID   01bd0864-0000-8c59-...

Show more ⌄