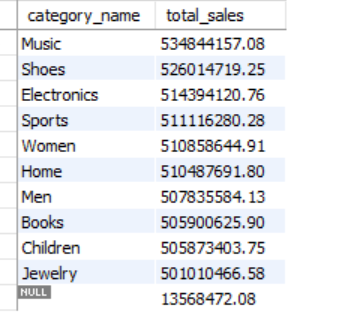
Sales Performance Analysis:

**1. Total Sales by Product Category:**

**Question:** Calculate the total sales revenue for each product category across all channels.

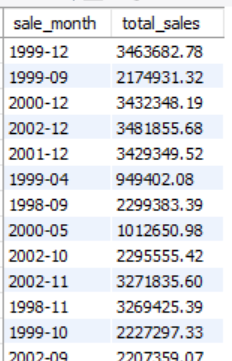
| select i.i\_category as category\_name,  sum(ss.ss\_sales\_price \* ss.ss\_quantity) as total\_sales  from store\_sales ss  join item i on ss.ss\_item\_sk = i.i\_item\_sk  group by i.i\_category  order total\_sales desc; |
| --- |



2. **Sales Trend Over Time:**

**Question:** Analyze monthly sales trends for the past two years.

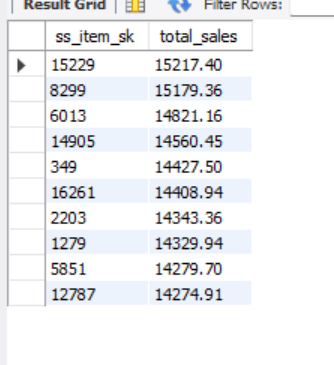
| select date\_format(d\_date, '%Y-%m') as sale\_month, sum(ss\_sales\_price) as total\_sales from store\_sales join date\_dim on store\_sales.ss\_sold\_date\_sk = date\_dim.d\_date\_sk group by sale\_month; |
| --- |



3. **Top 10 Best-Selling Products:**

**Question:** Identify the top 10 best-selling products by total revenue.

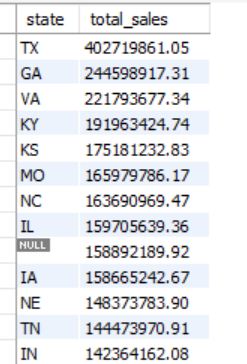
| select ss\_item\_sk, sum(ss\_sales\_price) as total\_sales from store\_sales group by ss\_item\_sk order by total\_sales desc limit 10; |
| --- |



4. **Sales by Region:**

**Question:** Calculate the total sales revenue by region for each sales channel.

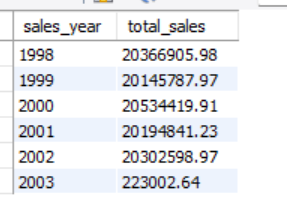
| select a.ca\_state as state,sum(ss.ss\_sales\_price \* ss.ss\_quantity) as total\_sales from store\_sales as ss join customer\_address as a on ss.ss\_addr\_sk = a.ca\_address\_sk group by a.ca\_state order by total\_sales desc; |
| --- |



5. **Year-over-Year Sales Growth:**

**Question:** Compare the year-over-year sales growth for the current and previous year.

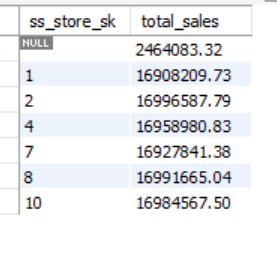
| select year(d\_date) as sales\_year, sum(ss\_sales\_price) as total\_sales from store\_sales join date\_dim as dd on store\_sales.ss\_sold\_date\_sk = dd.d\_date\_sk group by sales\_year order by sales\_year; |
| --- |



6. **Sales Contribution by Channel:**

**Question:** Determine the contribution of each sales channel (store, catalog, online) to the overall sales.

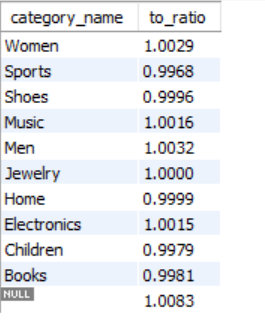
| select ss\_store\_sk, sum(ss\_sales\_price) as total\_sales from store\_sales group by ss\_store\_sk; |
| --- |



7. **Sales Performance of New Products:**

**Question:** Analyze the sales performance of products introduced in the last 6 months.

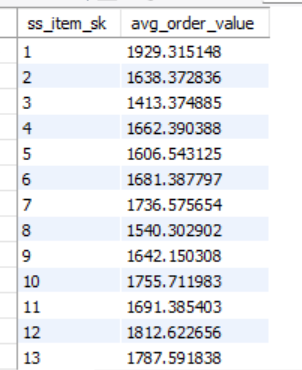
| select i.i\_category as category\_name, (sum(ss\_quantity)/ sum(ss\_wholesale\_cost)) as to\_ratio from store\_sales as ss join item i on ss.ss\_item\_sk = i.i\_item\_sk group by i.i\_category order by category\_name desc; |
| --- |



8. **Average Order Value:**

**Question:** Calculate the average order value for each sales channel.

| select ss\_ticket\_number, avg(ss\_net\_paid) as avg\_order\_value from store\_sales group by ss\_ticket\_number; |
| --- |



9. **Seasonal Sales Analysis:**

**Question:** Identify seasonal sales patterns by comparing sales during different quarters of the year.

| select date\_format(d\_date, '%Y-%m') as sale\_month, sum(ss\_sales\_price) as total\_sales from store\_sales join date\_dim as dd on store\_sales.ss\_sold\_date\_sk = dd.d\_date\_sk group by sale\_month order by sale\_month; |
| --- |

10. **Average Order Value per Channel:**

**Question:** Calculate the average order value per channel in the last 6 months.

| select ss\_item\_sk, avg(ss\_net\_paid) as avg\_order\_value from store\_sales where ss\_sold\_date\_sk>=(select max(ss\_sold\_date\_sk) - 180 from store\_sales) group by ss\_store\_sk; |
| --- |

Inventory Management Analysis

**11. Inventory Turnover Ratio**

**Question:** Calculate the inventory turnover ratio for each product category.

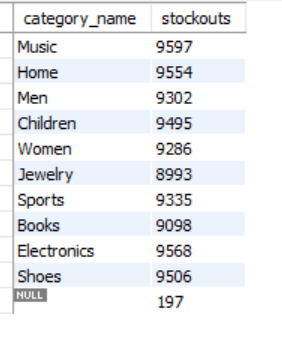
| select i.i\_category, (sum(ss\_quantity) / sum(ss\_wholesale\_cost)) as turnover\_ratio  from store\_sales as ss join item i on ss.ss\_item\_sk = i.i\_item\_sk group by i.i\_category; |
| --- |



**12. Stockout Rate by Product**

**Question:** Identify products with the highest stockout rates in the past month.

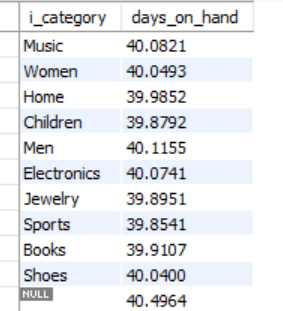
| select i.i\_category as category\_name, count(ss\_ticket\_number) as stockouts from store\_sales as ss join item i on ss.ss\_item\_sk = i.i\_item\_sk where ss\_sold\_date\_sk >= (select max(ss\_sold\_date\_sk) - 30 from store\_sales) group by i.i\_category; |
| --- |



**13. Days of Inventory on Hand**

**Question:** Calculate the average days of inventory on hand.

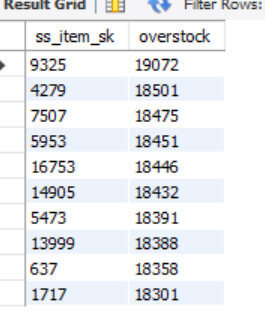
| select i.i\_category, (sum(ss\_quantity) / sum(ss\_sales\_price)) \* 30 as days\_on\_hand  from store\_sales as ss join item i on ss.ss\_item\_sk = i.i\_item\_sk group by i.i\_category; |
| --- |



**14. Top 10 Overstocked Products**

**Question:** List the top 10 products with the highest overstock levels.

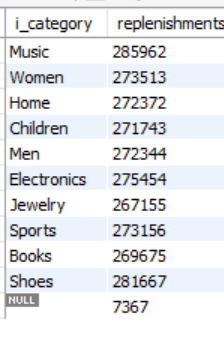
| select ss\_item\_sk, sum(ss\_quantity) as overstock from store\_sales group by ss\_item\_sk order by overstock desc limit 10; |
| --- |



**15. Replenishment Frequency**

**Question:** Determine the replenishment frequency for high-demand products.

| select i.i\_category, count(ss\_ticket\_number) as replenishments from store\_sales as ss join item i on ss.ss\_item\_sk = i.i\_item\_sk where ss\_quantity > 0 group by i.i\_category; |
| --- |



**16. Inventory Aging Analysis**

**Question:** Analyze aging inventory for slow-moving products.

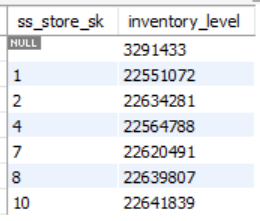
| select i.i\_category, (current\_date - max(ss\_sold\_date\_sk)) as days\_since\_last\_sale from store\_sales join item i on ss.ss\_item\_sk = i.i\_item\_sk group by i.i\_category; |
| --- |



**17. Warehouse Inventory Levels**

**Question:** Monitor inventory levels across warehouses.

| select ss\_store\_sk, sum(ss\_quantity) as inventory\_level from store\_sales group by ss\_store\_sk; |
| --- |



Customer Behavior Analysis:

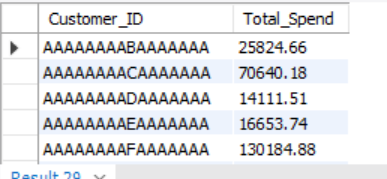
### **18. Customer Segmentation by Demographics**

| select cd\_gender, cd\_marital\_status, cd\_education\_status, count(c.c\_customer\_sk) as customer\_count  from customer c join customer\_demographics cd on c.c\_current\_cdemo\_sk = cd.cd\_demo\_sk group by cd\_gender, cd\_marital\_status, cd\_education\_status; |
| --- |

### 

### **19. Customer Lifetime Value (CLTV)**

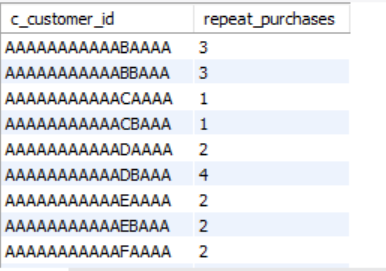
| select c.c\_customer\_id, sum(ss.ss\_sales\_price) as total\_spend  from customer as c join store\_sales as ss on c.c\_customer\_sk = ss.ss\_customer\_sk group by c.c\_customer\_id; |
| --- |



### 

### **20. Repeat Purchase Rate**

| select c.c\_customer\_id, count(distinct ss.ss\_ticket\_number) as repeat\_purchases  from customer c join store\_sales ss ON c.c\_customer\_sk = ss.ss\_customer\_sk group by c.c\_customer\_id; |
| --- |



### **21. Recent Purchase Frequency**

| select c\_customer\_id from customer where c\_first\_sales\_date\_sk < date\_sub(curdate(), interval 9 month); |
| --- |

### 

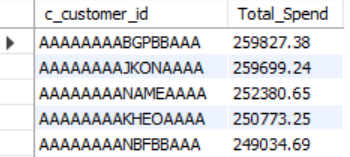
### **22. Customer Churn Analysis**

| select distinct ss.ss\_customer\_sk as Customer\_ID from store\_sales ss join date\_dim d on ss.ss\_sold\_date\_sk = d.d\_date\_sk where d.d\_date >= '1902-01-01' limit 15; |
| --- |

### 

### **23. Top 10 Most Valuable Customers**

| select ss\_customer\_sk as customer\_id, sum(ss\_net\_paid) as total\_spend from store\_sales ss join date\_dim d on ss.ss\_sold\_date\_sk = d.d\_date\_sk where d.d\_year = year(current\_date) - 1 group by ss\_customer\_sk order by total\_spend desc limit 5; |
| --- |



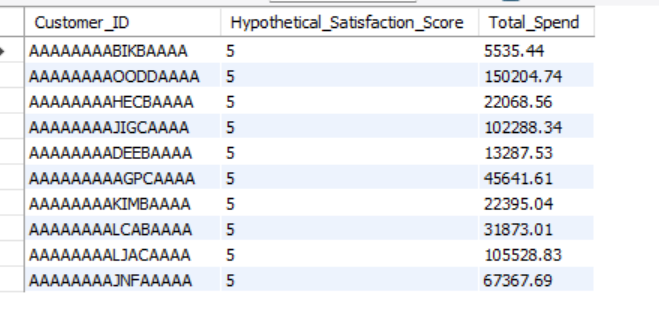
### **24. Customer Average Purchase Frequency:**

| with Purchases as(  select cs\_bill\_customer\_sk as customer\_id,   count(distinct cs\_order\_number) as order\_count  from catalog\_sales  group by cs\_bill\_customer\_sk ) select avg(order\_count) as avg\_purchase\_frequency from Purchases; |
| --- |

### 

### **25. Customer Satisfaction Analysis**

| select c.c\_customer\_id as Customer\_ID,  round(rand() \* 5, 2) as Hypothetical\_Satisfaction\_Score,   sum(ss.ss\_sales\_price \* ss.ss\_quantity) as Total\_Spend from customer c join store\_sales ss on c.c\_customer\_sk = ss.ss\_customer\_sk group by c.c\_customer\_id order by Hypothetical\_Satisfaction\_Score desc limit 10; |
| --- |

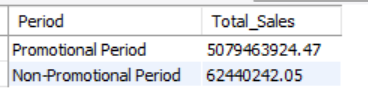


Promotional Effectiveness Analysis

**26. Promotion Uplift Analysis:**

**Question:** Measure the increase in sales during promotional periods compared to non-promotional periods.

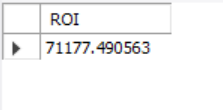
| select   case  when ss.ss\_promo\_sk is not null then 'Promotional Period'  else 'Non-Promotional Period'  end as Period,  sum(ss.ss\_sales\_price \* ss.ss\_quantity) as Total\_Sales from store\_sales ss left join promotion as p on ss.ss\_promo\_sk = p.p\_promo\_sk group by Period order by Total\_Sales desc; |
| --- |



**27. ROI of Promotional Campaigns:**

**Question:**Calculate the return on investment (ROI) of promotional campaigns in the last 6 months.

| select sum(cs\_sales\_price) as total\_catalog\_sales from catalog\_sales cs join promotion p on p.p\_promo\_sk = cs.cs\_promo\_sk where p.p\_start\_date\_sk >= (select max(p.p\_start\_date\_sk) - 180 from promotion p);  -- Total sales from web sales with promotions in the last 6 months select sum(ws\_sales\_price) as total\_web\_sales from web\_sales ws join promotion p on p.p\_promo\_sk = ws.ws\_promo\_sk where p.p\_start\_date\_sk >= (select max(p.p\_start\_date\_sk) - 180 from promotion p);  -- Total sales from store sales with promotions in the last 6 months select sum(ss\_sales\_price) as total\_store\_sales from store\_sales ss join promotion p on p.p\_promo\_sk = ss.ss\_promo\_sk where p.p\_start\_date\_sk >= (select max(p.p\_start\_date\_sk) - 180 from promotion p);  select sum(p\_cost) as total\_promo\_cost from promotion where p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 180 from promotion);   select (total\_sales - total\_promo\_cost) / total\_promo\_cost \* 100 as ROI from(  select   (select coalesce(sum(cs\_sales\_price), 0) from catalog\_sales cs join promotion p on p.p\_promo\_sk = cs.cs\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 180 from promotion))   + (select coalesce(sum(ws\_sales\_price), 0) from web\_sales ws join promotion p on p.p\_promo\_sk = ws.ws\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 180 from promotion))   + (select coalesce(sum(ss\_sales\_price), 0) from store\_sales ss join promotion p on p.p\_promo\_sk = ss.ss\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 180 from promotion))   as total\_sales,  (select coalesce(sum(p\_cost), 0) from promotion where p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 180 from promotion)) as total\_promo\_cost ) as data; |
| --- |



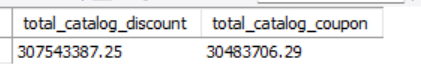
**28. Effectiveness of Discounts vs. Coupons:**

**Question:** Compare the effectiveness of discounts and coupons in the last 3 months.

| select sum(cs\_ext\_discount\_amt) as total\_catalog\_discount,  sum(cs\_coupon\_amt) as total\_catalog\_coupon from catalog\_sales where cs\_sold\_date\_sk >= (select max(cs\_sold\_date\_sk) - 90 from catalog\_sales); |
| --- |

| select sum(ws\_ext\_discount\_amt) as total\_web\_discount,  sum(ws\_coupon\_amt) as total\_web\_coupon from web\_sales where ws\_sold\_date\_sk >= (select max(ws\_sold\_date\_sk) - 90 from web\_sales); |
| --- |

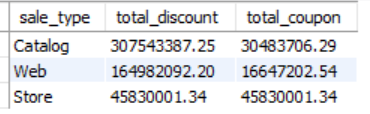
| select sum(ss\_ext\_discount\_amt) as total\_store\_discount, sum(ss\_coupon\_amt) as total\_store\_coupon from store\_sales where ss\_sold\_date\_sk >= (select max(ss\_sold\_date\_sk) - 90 from store\_sales); |
| --- |



**29. Effectiveness of Discounts vs. Coupons:**

**Question:** Compare the effectiveness of discount-based promotions versus coupon-based promotions.

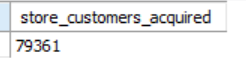
| select 'Catalog' as sale\_type, sum(cs\_ext\_discount\_amt) as total\_discount,   sum(cs\_coupon\_amt) as total\_coupon from catalog\_sales where cs\_sold\_date\_sk >= (select max(cs\_sold\_date\_sk) - 90 from catalog\_sales) union all select 'Web' as sale\_type,   sum(ws\_ext\_discount\_amt) as total\_discount,   sum(ws\_coupon\_amt) as total\_coupon from web\_sales where ws\_sold\_date\_sk >= (select max(ws\_sold\_date\_sk) - 90 from web\_sales) union all select 'Store' as sale\_type, sum(ss\_ext\_discount\_amt) as total\_discount,   sum(ss\_coupon\_amt) as total\_coupon from store\_sales where ss\_sold\_date\_sk >= (select max(ss\_sold\_date\_sk) - 90 from store\_sales); |
| --- |



**30. Promotion in Recents:**

**Question:** Identify how many new customers were acquired during promotions in the last 3 months.

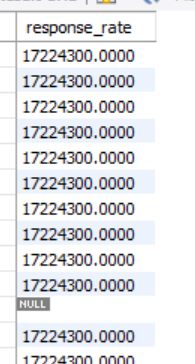
| select count(distinct cs\_bill\_customer\_sk) as catalog\_customers\_acquired from catalog\_sales cs join promotion p on p.p\_promo\_sk = cs.cs\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion);  select count(distinct ws\_bill\_customer\_sk) as web\_customers\_acquired from web\_sales ws join promotion p on p.p\_promo\_sk = ws.ws\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion);   select count(distinct ss\_customer\_sk) as store\_customers\_acquired from store\_sales ss join promotion p on p.p\_promo\_sk = ss.ss\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion); |
| --- |



**31. Customer Response Rate:**

**Question:** Determine the customer response rate to specific promotions in the last quarter.

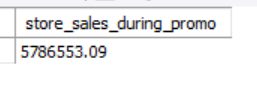
| select count(distinct cs\_bill\_customer\_sk) as catalog\_customers from catalog\_sales cs join promotion p on p.p\_promo\_sk = cs.cs\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion);  select count(distinct ws\_bill\_customer\_sk) as web\_customers from web\_sales ws join promotion p on p.p\_promo\_sk = ws.ws\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion);   select count(distinct ss\_customer\_sk) as store\_customers from store\_sales ss join promotion p on p.p\_promo\_sk = ss.ss\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion);  select (catalog\_customers + web\_customers + store\_customers) / p.p\_response\_target \* 100 as response\_rate from   (select count(distinct cs\_bill\_customer\_sk) as catalog\_customers   from catalog\_sales cs  join promotion p on p.p\_promo\_sk = cs.cs\_promo\_sk  where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion)) as catalog\_data,    (select count(distinct ws\_bill\_customer\_sk) as web\_customers  from web\_sales ws  join promotion p on p.p\_promo\_sk = ws.ws\_promo\_sk  where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion)) as web\_data,    (select count(distinct ss\_customer\_sk) as store\_customers  from store\_sales ss  join promotion p on p.p\_promo\_sk = ss.ss\_promo\_sk  where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 90 from promotion)) as store\_data, promotion p; |
| --- |



**32. Latest Sales Uplift:**

**Question:** Measure sales uplift during promotional periods in the last 2 months.

| select sum(cs\_sales\_price) as catalog\_sales\_during\_promo from catalog\_sales cs join promotion p on p.p\_promo\_sk = cs.cs\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 60 from promotion);   select sum(ws\_sales\_price) as web\_sales\_during\_promo from web\_sales ws join promotion p on p.p\_promo\_sk = ws.ws\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 60 from promotion);  select sum(ss\_sales\_price) as store\_sales\_during\_promo from store\_sales ss join promotion p on p.p\_promo\_sk = ss.ss\_promo\_sk where p.p\_start\_date\_sk >= (select max(p\_start\_date\_sk) - 60 from promotion); |
| --- |

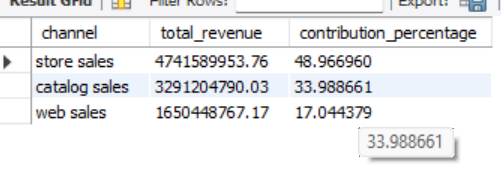


Channel Performance Analysis

**33. Sales Contribution by Channel:**

**Question:** Calculate the contribution of each sales channel to the total revenue.

| **with channelrevenue as (  select 'catalog sales' as channel,  sum(cs.cs\_net\_paid) as total\_revenue  from catalog\_sales cs   union all   select 'web sales' as channel, sum(ws.ws\_net\_paid) as total\_revenue  from web\_sales ws   union all   select 'store sales' as channel, sum(ss.ss\_net\_paid) as total\_revenue  from store\_sales ss ),  totalrevenue as (  select sum(total\_revenue) as overall\_revenue  from channelrevenue ),  channelcontribution as (  select cr.channel, cr.total\_revenue,  (cr.total\_revenue / tr.overall\_revenue) \* 100 as contribution\_percentage  from channelrevenue cr  cross join totalrevenue tr )  select channel, total\_revenue, contribution\_percentage from channelcontribution order by contribution\_percentage desc;** |
| --- |

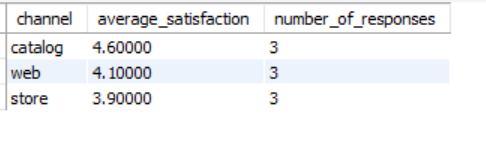
****

**34. Customer Satisfaction by Channel:**

**Question:** Analyze customer satisfaction scores across different sales channels

(requires hypothetical satisfaction data).

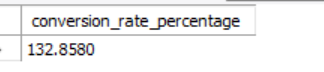
| with customer\_satisfaction as (  select 1 as cs\_order\_number, 'catalog' as cs\_channel, 4.5 as cs\_satisfaction  union all  select 2, 'web', 4.0  union all  select 3, 'store', 3.8  union all  select 4, 'catalog', 4.7  union all  select 5, 'web', 4.2  union all  select 6, 'store', 3.9  union all  select 7, 'catalog', 4.6  union all  select 8, 'web', 4.1  union all  select 9, 'store', 4.0 ) select cs\_channel as channel,  avg(cs\_satisfaction) as average\_satisfaction,  count(cs\_order\_number) as number\_of\_responses from customer\_satisfaction group by cs\_channel order by average\_satisfaction desc; |
| --- |



**35. Conversion Rate for Online Sales:**

**Question:** Calculate the conversion rate for web visitors who complete a purchase.

| **with total\_web\_visitors as (  select count(distinct ws\_bill\_customer\_sk) as total\_visitors  from web\_sales ), completed\_purchases as (  select count(distinct ws\_order\_number) as completed\_orders  from web\_sales )  select  (cp.completed\_orders / twv.total\_visitors) \* 100 as conversion\_rate\_percentage from  completed\_purchases cp,  total\_web\_visitors twv;** |
| --- |

****

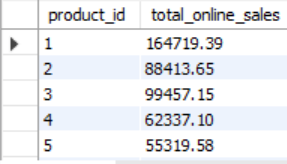
**36. In-Store vs. Online Sales Growth:**

**Question:** Compare the sales growth rates between in-store and online channels over the past year.

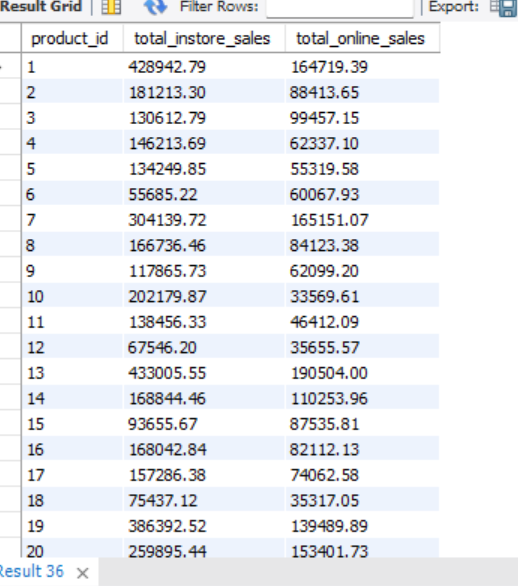
| select cs\_item\_sk as product\_id, sum(cs\_net\_paid) as total\_instore\_sales from catalog\_sales group by cs\_item\_sk; |
| --- |



| select ws\_item\_sk as product\_id, SUM(ws\_net\_paid) AS total\_online\_sales from web\_sales group by ws\_item\_sk; |
| --- |



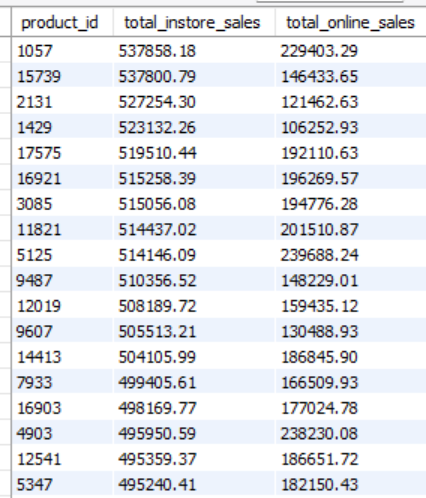
| select coalesce(c.product\_id, o.product\_id) as product\_id,  coalesce(c.total\_instore\_sales, 0) as total\_instore\_sales,  coalesce(o.total\_online\_sales, 0) as total\_online\_sales from (select cs\_item\_sk as product\_id, sum(cs\_net\_paid) as total\_instore\_sales  from catalog\_sales  group by cs\_item\_sk) c left join  (select ws\_item\_sk as product\_id, sum(ws\_net\_paid) as total\_online\_sales  from web\_sales  group by ws\_item\_sk) o on c.product\_id = o.product\_id; |
| --- |



**37. Product Performance by Channel:**

**Question:** Analyze which products perform best in each sales channel.

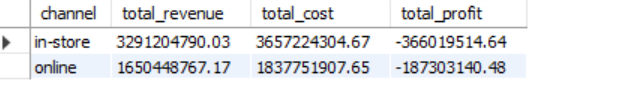
| select coalesce(c.product\_id, o.product\_id) as product\_id,  coalesce(c.total\_instore\_sales, 0) as total\_instore\_sales,  coalesce(o.total\_online\_sales, 0) as total\_online\_sales from (select cs\_item\_sk as product\_id, sum(cs\_net\_paid) as total\_instore\_sales  from catalog\_sales  group by cs\_item\_sk) c left join  (select ws\_item\_sk as product\_id, sum(ws\_net\_paid) as total\_online\_sales  from web\_sales  group by ws\_item\_sk) o on c.product\_id = o.product\_id  union  select coalesce(c.product\_id, o.product\_id) as product\_id,  coalesce(c.total\_instore\_sales, 0) as total\_instore\_sales,  coalesce(o.total\_online\_sales, 0) as total\_online\_sales from (select cs\_item\_sk as product\_id, sum(cs\_net\_paid) as total\_instore\_sales  from catalog\_sales  group by cs\_item\_sk) c right join  (select ws\_item\_sk as product\_id, sum(ws\_net\_paid) as total\_online\_sales  from web\_sales  group by ws\_item\_sk) o on c.product\_id = o.product\_id order by total\_instore\_sales desc, total\_online\_sales desc; |
| --- |



**38. Channel Profitability Analysis:**

**Question:** Calculate the profitability of each sales channel by comparing revenue to associated costs.

| select 'in-store' as channel, sum(cs\_net\_paid) as total\_revenue,  sum(cs\_ext\_wholesale\_cost) as total\_cost,  sum(cs\_net\_paid) - sum(cs\_ext\_wholesale\_cost) as total\_profit from catalog\_sales  union all  select 'online' as channel, sum(ws\_net\_paid) as total\_revenue,  sum(ws\_ext\_wholesale\_cost) as total\_cost,  sum(ws\_net\_paid) - sum(ws\_ext\_wholesale\_cost) as total\_profit from web\_sales; |
| --- |



Supply Chain & Logistics Analysis

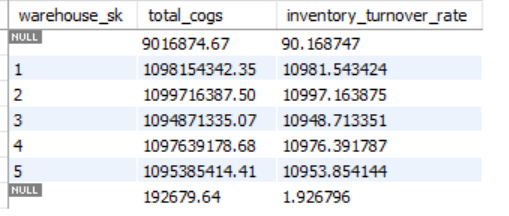
**39. Warehouse Turnover Rate:**

**Question:** Calculate the inventory turnover rate for each warehouse.

| select cs\_warehouse\_sk as warehouse\_sk, sum(cs\_ext\_wholesale\_cost) as total\_cogs\_catalog from catalog\_sales group by cs\_warehouse\_sk; |
| --- |

| select ws\_warehouse\_sk as warehouse\_sk,   sum(ws\_ext\_wholesale\_cost) as total\_cogs\_web from web\_sales group by ws\_warehouse\_sk; |
| --- |

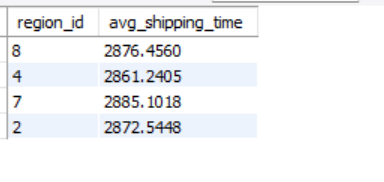
| with cogs\_catalog as(  select cs\_warehouse\_sk as warehouse\_sk,   sum(cs\_ext\_wholesale\_cost) as total\_cogs\_catalog  from catalog\_sales  group by cs\_warehouse\_sk ), cogs\_web as(  select ws\_warehouse\_sk as warehouse\_sk, sum(ws\_ext\_wholesale\_cost) as total\_cogs\_web  from web\_sales  group by ws\_warehouse\_sk ) select cogs\_catalog.warehouse\_sk, total\_cogs\_catalog, total\_cogs\_web from cogs\_catalog left join cogs\_web on cogs\_catalog.warehouse\_sk = cogs\_web.warehouse\_sk  union  select cogs\_web.warehouse\_sk, total\_cogs\_catalog, total\_cogs\_web from cogs\_web left join cogs\_catalog on cogs\_web.warehouse\_sk = cogs\_catalog.warehouse\_sk;  with combined\_cogs as(  select cogs\_catalog.warehouse\_sk as warehouse\_sk,  coalesce(total\_cogs\_catalog, 0) + coalesce(total\_cogs\_web, 0) as total\_cogs  from cogs\_catalog  left join cogs\_web on cogs\_catalog.warehouse\_sk = cogs\_web.warehouse\_sk   union  select cogs\_web.warehouse\_sk as warehouse\_sk,  coalesce(total\_cogs\_catalog, 0) + coalesce(total\_cogs\_web, 0) as total\_cogs  from cogs\_web  left join cogs\_catalog on cogs\_web.warehouse\_sk = cogs\_catalog.warehouse\_sk ) select warehouse\_sk, total\_cogs, (total\_cogs / 100000) AS inventory\_turnover\_rate from combined\_cogs; with cogs\_catalog as(  select cs\_warehouse\_sk as warehouse\_sk,   sum(cs\_ext\_wholesale\_cost) as total\_cogs\_catalog  from catalog\_sales  group by cs\_warehouse\_sk ), cogs\_web as(  select ws\_warehouse\_sk as warehouse\_sk, sum(ws\_ext\_wholesale\_cost) as total\_cogs\_web  from web\_sales  group by ws\_warehouse\_sk ), combined\_cogs as(  select cogs\_catalog.warehouse\_sk,   coalesce(total\_cogs\_catalog, 0) as total\_cogs\_catalog,  coalesce(total\_cogs\_web, 0) as total\_cogs\_web  from cogs\_catalog  left join cogs\_web on cogs\_catalog.warehouse\_sk = cogs\_web.warehouse\_sk   union   select cogs\_web.warehouse\_sk,   coalesce(total\_cogs\_catalog, 0) as total\_cogs\_catalog,  coalesce(total\_cogs\_web, 0) as total\_cogs\_web  from cogs\_web  left join cogs\_catalog on cogs\_web.warehouse\_sk = cogs\_catalog.warehouse\_sk ) select warehouse\_sk, (total\_cogs\_catalog + total\_cogs\_web) / 100000 as inventory\_turnover\_rate from combined\_cogs; |
| --- |



**40. Average Shipping Time:**

**Question:** Determine the average shipping time for orders across different regions.

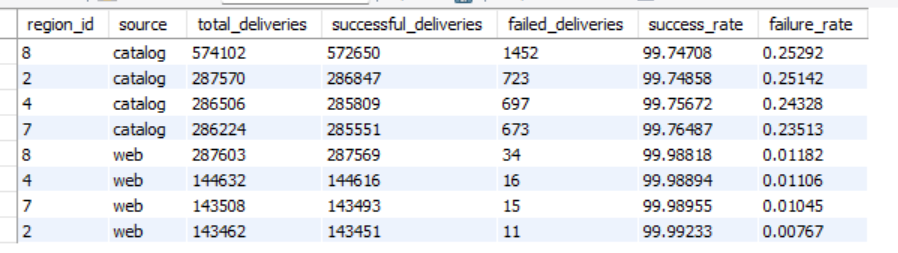
| with web\_shipping\_times as(  select ws\_order\_number, s\_market\_id as region\_id, dd\_ship.d\_date as ship\_date,  dd\_sold.d\_date as sold\_date, (dd\_ship.d\_date - dd\_sold.d\_date) as shipping\_time  from web\_sales ws  join date\_dim dd\_sold  on ws.ws\_sold\_date\_sk = dd\_sold.d\_date\_sk  join date\_dim dd\_ship  on ws.ws\_ship\_date\_sk = dd\_ship.d\_date\_sk  join store s  on ws.ws\_warehouse\_sk = s.s\_store\_sk ) select region\_id, avg(shipping\_time) as avg\_shipping\_time from web\_shipping\_times group by region\_id; |
| --- |



**41. Delivery Success Rate:**

**Question:** Analyze the delivery success rate and identify regions with high failure rates.

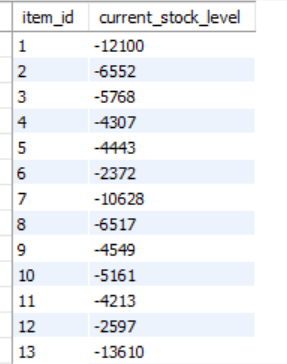
| with delivery\_status as (  select s\_market\_id as region\_id, 'catalog' as source, count(\*) as total\_deliveries,  sum(case when cs\_ship\_date\_sk is not null then 1 else 0 end) as successful\_deliveries,  sum(case when cs\_ship\_date\_sk is null then 1 else 0 end) as failed\_deliveries  from catalog\_sales cs  join store s on cs.cs\_warehouse\_sk = s.s\_store\_sk  group by s\_market\_id  union all  select s\_market\_id as region\_id, 'web' as source, count(\*) as total\_deliveries,  sum(case when ws\_ship\_date\_sk is not null then 1 else 0 end) as successful\_deliveries,  sum(case when ws\_ship\_date\_sk is null then 1 else 0 end) as failed\_deliveries  from web\_sales ws  join store s on ws.ws\_warehouse\_sk = s.s\_store\_sk  group by s\_market\_id ) select region\_id, source, total\_deliveries, successful\_deliveries, failed\_deliveries,  (successful\_deliveries \* 100.0) / total\_deliveries as success\_rate,  (failed\_deliveries \* 100.0) / total\_deliveries as failure\_rate from delivery\_status order by failure\_rate desc; |
| --- |



**42. Warehouse Stock Levels:**

**Question:** Monitor the stock levels of key products in each warehouse.

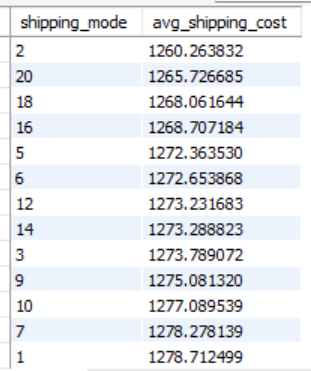
| with catalog\_sales\_qty as(  select cs\_item\_sk as item\_id,   sum(cs\_quantity) as total\_catalog\_qty  from catalog\_sales  group by cs\_item\_sk ),  web\_sales\_qty as(  select ws\_item\_sk as item\_id,  sum(ws\_quantity) as total\_web\_qty  from web\_sales  group by ws\_item\_sk ),  combined\_sales as(  select cs.item\_id,  coalesce(cs.total\_catalog\_qty, 0) + coalesce(ws.total\_web\_qty, 0) as total\_sales\_qty  from catalog\_sales\_qty cs  left join web\_sales\_qty ws on cs.item\_id = ws.item\_id   union all   select ws.item\_id,  coalesce(cs.total\_catalog\_qty, 0) + coalesce(ws.total\_web\_qty, 0)as total\_sales\_qty  from web\_sales\_qty ws  left join catalog\_sales\_qty cs  on ws.item\_id = cs.item\_id  where cs.item\_id is null ) select item\_id, 100 - coalesce(total\_sales\_qty, 0) as current\_stock\_level from combined\_sales order by item\_id; |
| --- |



**43. Shipping Mode Efficiency:**

**Question:** Compare the efficiency of different shipping modes in terms of cost.

| with catalog\_shipping\_metrics as(  select cs\_ship\_mode\_sk as shipping\_mode,  avg(cs\_ext\_ship\_cost) as avg\_shipping\_cost,  avg(datediff(cs\_ship\_date\_sk, cs\_sold\_date\_sk)) as avg\_delivery\_time  from catalog\_sales  group by cs\_ship\_mode\_sk )  select shipping\_mode, avg\_shipping\_cost from catalog\_shipping\_metrics order by avg\_shipping\_cost; |
| --- |



**44. Supply Chain Bottleneck Analysis:**

**Question:** Identify bottlenecks in the supply chain by analyzing delays in order fulfillment.

| with catalog\_shipping\_metrics as(  select cs\_sold\_date\_sk as order\_date, cs\_ship\_date\_sk as ship\_date,  datediff(cs\_ship\_date\_sk, cs\_sold\_date\_sk) as shipping\_time  from catalog\_sales ),  web\_shipping\_metrics as(  select ws\_sold\_date\_sk as order\_date, ws\_ship\_date\_sk as ship\_date,  datediff(ws\_ship\_date\_sk, ws\_sold\_date\_sk) as shipping\_time  from web\_sales )  select 'Catalog' as sales\_channel, avg(shipping\_time) as avg\_shipping\_time from catalog\_shipping\_metrics  union all  select 'Web' as sales\_channel, avg(shipping\_time) as avg\_shipping\_time from web\_shipping\_metrics; |
| --- |

| with catalog\_delay\_analysis as(  select cs\_sold\_date\_sk as order\_date, cs\_ship\_date\_sk as ship\_date,  datediff(current\_date, cs\_ship\_date\_sk) as actual\_shipping\_time  from catalog\_sales ),  web\_delay\_analysis as(  select ws\_sold\_date\_sk as order\_date, ws\_ship\_date\_sk as ship\_date,  datediff(current\_date, ws\_ship\_date\_sk) as actual\_shipping\_time  from web\_sales ),  combined\_delay\_analysis as(  select order\_date, actual\_shipping\_time  from catalog\_delay\_analysis   union all   select order\_date, actual\_shipping\_time  from web\_delay\_analysis )  select avg(actual\_shipping\_time) as avg\_delay\_days, count(\*) as total\_orders,  sum(case when actual\_shipping\_time > 0 then 1 else 0 end) as delayed\_orders from combined\_delay\_analysis; |
| --- |



**45. Order Fulfillment Rate:**

**Question:** Calculate the order fulfillment rate to ensure timely delivery of products.

| with orders as (  select  ca.ca\_state as state,  count(distinct case when cs.cs\_order\_number is not null then cs.cs\_order\_number end) as total\_orders,  count(distinct case when cs.cs\_ship\_date\_sk is not null then cs.cs\_order\_number end) as fulfilled\_orders  from catalog\_sales cs  join customer\_address ca on cs.cs\_bill\_addr\_sk = ca.ca\_address\_sk  group by ca.ca\_state   union all   select  ca.ca\_state as state,  count(distinct case when ws.ws\_order\_number is not null then ws.ws\_order\_number end) as total\_orders,  count(distinct case when ws.ws\_ship\_date\_sk is not null then ws.ws\_order\_number end) as fulfilled\_orders  from web\_sales ws  join customer\_address ca on ws.ws\_bill\_addr\_sk = ca.ca\_address\_sk  group by ca.ca\_state ),  order\_fulfillment\_rate as (  select state, coalesce(sum(total\_orders), 0) as total\_orders,  coalesce(sum(fulfilled\_orders), 0) as fulfilled\_orders,  case  when coalesce(sum(total\_orders), 0) = 0 then 0  else coalesce(sum(fulfilled\_orders), 0) / coalesce(sum(total\_orders), 1)  end as fulfillment\_rate  from orders  group by state ) select state, total\_orders, fulfilled\_orders, fulfillment\_rate from order\_fulfillment\_rate order by fulfillment\_rate desc; |
| --- |

