

```
create database if not exists indexing;

use indexing;

show tables;

rename table `global superstore 2018 _ csv format` to superstore;

show tables;

describe superstore;

alter table superstore rename column `Row ID` to row_id;

select*from superstore

order by row_id;

-- 1)

explain analyze

select *from superstore

where row_id =21550;

-- -> (cost=409 rows=385) (actual time=16.8..19.9 rows=1 loops=1)

create index idx_row_id on superstore(row_id);

explain analyze

select *from superstore

where row_id =21550;

-- -> (cost=0.35 rows=1) (actual time=0.0915..0.0995 rows=1 loops=1)

show indexes from superstore;

describe superstore;
```

```
-- 2 ) find regionwise sales  
explain analyze  
select region,round(sum(sales),2) as sale from superstore  
group by region  
order by sum(sales)desc;  
-- -> (actual time=20.1..20.1 rows=8 loops=1) (actual time=20..20 rows=8 loops=1)
```

```
create index idx_region
```

```
on superstore (region(50));  
explain analyze  
select region,round(sum(sales),2) as sale from superstore  
group by region  
order by sum(sales)desc;
```

```
show indexes from superstore;
```

```
explain analyze  
select*from superstore  
where `customer id` ='ea-140355' and region =' south amrica';
```

```
-- 3) creating a composite index
```

```
create index idx_cust_region  
on superstore (region(50),`customer id`(30));  
explain analyze  
select *from superstore
```

```
where `customer id` ='ea-140355' and region =' south amrica';
```

```
select *from superstore;
```

```
show indexes from superstore;
```

```
drop index idx_cust_region on superstore;
```

```
-- exercise questions
```

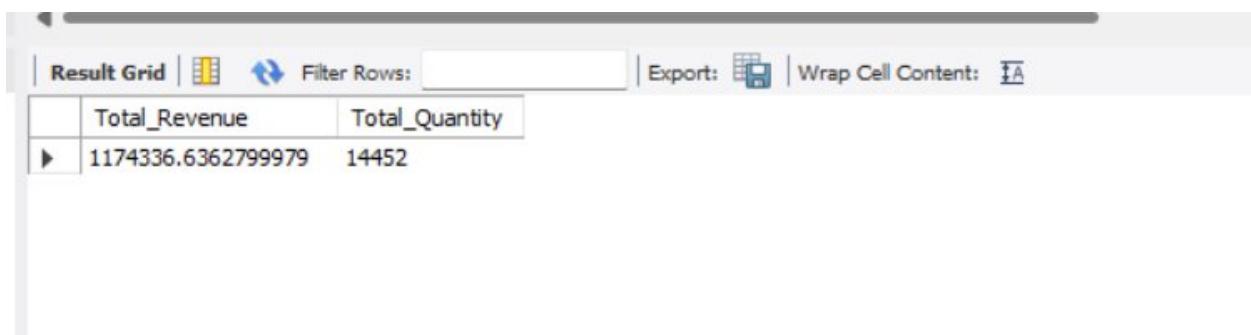
```
-- 1) find the total revenue , quantities generated
```

```
SELECT
```

```
    SUM(Sales) AS Total_Revenue,
```

```
    SUM(Quantity) AS Total_Quantity
```

```
FROM superstore;
```



The screenshot shows a MySQL Workbench result grid. The grid has two columns: 'Total_Revenue' and 'Total_Quantity'. There is one data row containing the values 1174336.6362799979 and 14452 respectively.

	Total_Revenue	Total_Quantity
▶	1174336.6362799979	14452

-- 2) find the segment wise distribution of the sales

```
SELECT segment,  
(SUM(sales)) AS total_sales  
FROM superstore  
GROUP BY segment  
ORDER BY total_sales DESC;
```

The screenshot shows a software interface for viewing database results. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below this is a table with three rows of data:

	segment	total_sales
▶	Consumer	624094.8519599998
▶	Corporate	350747.61732000054
▶	Home Office	199494.16700000007

At the bottom left of the results area, it says 'Result 44'.

-- 3)find the top 3 most profitable product

```
SELECT  
 `Product Name` AS product_name,  
 ROUND(SUM(profit), 2) AS total_profit  
FROM superstore  
GROUP BY `Product Name`  
ORDER BY total_profit DESC  
LIMIT 3;
```

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

	product_name	total_profit
▶	Sauder Classic Bookcase, Metal	2978.37
	Nokia Smart Phone, with Caller ID	2887.59
	Novimex Executive Leather Armchair, Adjustable	2523.55

-- 4) find how many orders are placed after 2016

```
SELECT COUNT(*) AS total_orders_after_2016
FROM superstore
WHERE YEAR(`Order Date`) >'31-1-2016';
```

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

	total_orders_after_2016
▶	0

Result 40 ×

-- 5) how many states from austria are under the roof of business ?

```
SELECT
COUNT(DISTINCT state) AS total_states_in_austria
```

```
FROM superstore  
WHERE country = 'Austria';
```

The screenshot shows a database query results window. At the top, there is a status bar with the number '99' and the text '^Product Name AS product_name,'. Below this is a toolbar with buttons for 'Result Grid', 'Filter Rows:', 'Export:', and 'Wrap Cell Content:'. The main area displays a single row in a grid. The first column is labeled 'total_states_in_austria' and contains the value '6'. A navigation arrow is visible to the left of the row.

total_states_in_austria
6

-- 6) which products and subcategories are most and least profitable

```
(SELECT  
`Sub-Category` AS sub_category,  
`Product Name` AS product_name,  
SUM(Profit) AS total_profit  
FROM superstore  
GROUP BY `Sub-Category`, `Product Name`  
ORDER BY total_profit DESC  
LIMIT 5  
)  
UNION ALL
```

```

(
SELECT
`Sub-Category` AS sub_category,
`Product Name` AS product_name,
SUM(Profit) AS total_profit
FROM superstore
GROUP BY `Sub-Category`, `Product Name`
ORDER BY total_profit ASC
LIMIT 5
);

```

ORDER BY total_revenue DESC

The screenshot shows a database query results grid. The columns are labeled 'sub_category', 'product_name', and 'total_profit'. The data is ordered by total profit in descending order. The results are as follows:

sub_category	product_name	total_profit
Bookcases	Sauder Classic Bookcase, Metal	2978.3700000000003
Phones	Nokia Smart Phone, with Caller ID	2887.594
Chairs	Novimex Executive Leather Armchair, Adjustable	2523.5519999999997
Chairs	Hon Executive Leather Armchair, Adjustable	2410.2749999999996
Copiers	Brother Copy Machine, Color	1963.362
Bookcases	Ikea Library with Doors, Traditional	-1748.1749999999997

Result 33 ×

Output

-- 7) which customer segment contributes the most to the total revenue

```

SELECT segment,
SUM(sales) AS total_revenue
FROM superstore
GROUP BY segment

```

```
ORDER BY total_revenue DESC
```

```
LIMIT 1;
```

The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows', 'Export' (with icons for CSV, Excel, and PDF), 'Wrap Cell Content', and 'Fetch rows'. Below the header is a single data row:

	segment	total_revenue
▶	Consumer	624094.8519599998

At the bottom left of the grid area, it says 'Result 31'.

-- 8) what is the year-over-year growth in sales and profit

```
SELECT
```

```
    YEAR(`Order Date`) AS order_year,  
    ROUND(SUM(Sales), 2) AS total_sales,  
    ROUND(SUM(Profit), 2) AS total_profit,  
    ROUND(  
        SUM(Sales) - LAG(SUM(Sales)) OVER (ORDER BY YEAR(`Order Date`)), 2  
    ) AS sales_growth,  
    ROUND(  
        SUM(Profit) - LAG(SUM(Profit)) OVER (ORDER BY YEAR(`Order Date`)), 2  
    ) AS profit_growth  
FROM superstore  
GROUP BY YEAR(`Order Date`)  
ORDER BY order_year;
```

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

	order_year	total_sales	total_profit	sales_growth	profit_growth
▶	NULL	1174336.64	134146.22	NULL	NULL

-- 9) Which countries and cities are driving the highest sales?

SELECT

```
country,
SUM(sales) AS total_sales
FROM superstore
GROUP BY country
ORDER BY total_sales DESC;
```

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

	country	total_sales
▶	Australia	925235.8530000002
	Austria	92539.04999999999
	Argentina	57511.78327999994
	Algeria	36091.58999999999
	Angola	25554.00000000001
	Afghanistan	21673.32000000003

Result 29 ×

-- 10. What is the average delivery time from order to ship date across regions?

```

SELECT
    region,
    ROUND(AVG(DATEDIFF(`Ship Date`, `Order Date`)), 2) AS avg_delivery_days
FROM superstore
GROUP BY region
ORDER BY avg_delivery_days;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	region	avg_delivery_days		
▶	Southern Asia	NULL		
	Southern Europe	NULL		
	North Africa	NULL		
	Central Africa	NULL		
	South America	NULL		
	Western Asia	NULL		

Result 28 ×

-- 11. what is the profit distribution across order priority?

```

SELECT
    `Order Priority` AS order_priority,
    SUM(profit) AS total_profit,
    AVG(profit) AS avg_profit_per_order
FROM superstore
GROUP BY `Order Priority`
ORDER BY total_profit DESC;

```

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

	order_priority	total_profit	avg_profit_per_order
▶	Medium	73509.69276000009	32.0443298866609
	High	46576.51983999997	36.703325327029134
	Critical	9776.810679999997	34.18465272727271
	Low	4283.192999999999	31.727355555555555

Result 26 ×

-- 12. Suggest data-driven recommendations for improving profit and reducing losses.

SELECT

```
`Product Name`,  
 `Sub-Category`,  
 SUM(Profit)as Total_Profit  
FROM superstore  
GROUP BY `Product Name`, `Sub-Category`  
HAVING SUM(Profit) < 0  
ORDER BY Total_Profit ASC;
```

176 ORDER BY Total_Profit ASC;

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

	Product Name	Sub-Category	Total_Profit
▶	Ikea Library with Doors, Traditional	Bookcases	-1748.174999999997
	Panasonic Inkjet, Red	Machines	-1410.192
	Chromcraft Conference Table, with Bottom Stor...	Tables	-1335.291
	Bevis Wood Table, with Bottom Storage	Tables	-1056.807
	Lesro Wood Table, Adjustable Height	Tables	-953.442
	Safco Classic Bookcase, Pine	Bookcases	-909.9

Result 25 ×

Output ::::::::::::