

End Of Season

SALE

SALE

SALE

**UP TO
40%
OFF**



SALE

1-WRITE A SQL QUERY TO RETRIEVE ALL COLUMNS FOR SALES MADE ON '2022-11-05':

```
SELECT * FROM SALES  
WHERE SALE_DATE='2022-11-05'
```

	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
▶	180	2022-11-05	10:47:00	117	Male	41	Clothing	3	300	129	900
	240	2022-11-05	11:49:00	95	Female	23	Beauty	1	300	123	300
	1256	2022-11-05	09:58:00	29	Male	23	Clothing	2	500	190	1000
	1587	2022-11-05	20:06:00	140	Female	40	Beauty	4	300	105	1200
	1819	2022-11-05	20:44:00	83	Female	35	Beauty	2	50	14	100
	943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
	1896	2022-11-05	20:19:00	87	Female	30	Electronics	2	25	31	50
	1137	2022-11-05	22:34:00	104	Male	46	Beauty	2	500	145	1000
	856	2022-11-05	17:43:00	102	Male	54	Electronics	4	30	9	120
	214	2022-11-05	16:31:00	53	Male	20	Beauty	2	30	8	60
	1265	2022-11-05	14:35:00	86	Male	55	Clothing	3	300	111	900

SALE

2-WRITE A SQL QUERY TO RETRIEVE ALL TRANSACTIONS WHERE THE CATEGORY IS 'CLOTHING' AND THE QUANTITY SOLD IS MORE THAN 4 IN THE MONTH OF NOV-2022:

SELECT

*

FROM SALES

WHERE


CATEGORY = 'CLOTHING'

AND

MONTH(SALE_DATE)=11 AND YEAR(SALE_DATE)=2022

AND

QUANTITY >= 4



Result Grid Filter Rows: Export: Wrap Cell Content:											
transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale	
1484	2022-11-23	09:29:00	22	Female	19	Clothing	4	300	147	1200	
64	2022-11-15	06:34:00	7	Male	49	Clothing	4	25	9	100	
284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	21	200	
1885	2022-11-09	07:32:00	148	Female	52	Clothing	4	30	11	120	
547	2022-11-14	07:36:00	3	Male	63	Clothing	4	500	250	2000	
159	2022-11-10	21:30:00	42	Male	26	Clothing	4	50	24	200	
699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16	120	
1259	2022-11-03	17:31:00	105	Female	45	Clothing	4	50	21	200	
146	2022-11-10	22:01:00	74	Male	38	Clothing	4	50	49	200	
1476	2022-11-11	22:27:00	130	Female	27	Clothing	4	500	555	2000	
1296	2022-11-26	20:42:00	45	Female	22	Clothing	4	300	342	1200	
1696	2022-11-21	17:59:00	24	Female	50	Clothing	4	50	55	200	
1497	2022-11-19	21:44:00	109	Male	41	Clothing	4	30	32	120	
735	2022-11-26	21:38:00	153	Female	64	Clothing	4	500	515	2000	

SALE

3-WRITE A SQL QUERY TO CALCULATE THE TOTAL SALES (TOTAL_SALE) FOR EACH CATEGORY.:



```
SELECT SUM(TOTAL_SALE) AS TOTAL_SALE,CATEGORY  
FROM SALES  
GROUP BY CATEGORY
```

Result Grid			Filter Rows:
	total_sale	category	
▶	309995	Clothing	
	286790	Beauty	
	311445	Electronics	

SALE

4-WRITE A SQL QUERY TO FIND THE AVERAGE AGE OF CUSTOMERS WHO PURCHASED ITEMS FROM THE 'BEAUTY' CATEGORY.:

```
SELECT AVG(AGE) AS AVERAGE_AGE,CATEGORY FROM  
SALES  
WHERE CATEGORY='BEAUTY'
```



Result Grid   Filter Rows		
	average_age	category
▶	40.4157	Beauty


SALE


5-WRITE A SQL QUERY TO FIND ALL TRANSACTIONS WHERE THE TOTAL_SALE IS GREATER THAN 1000.:

```
SELECT * FROM SALES WHERE TOTAL_SALE >1000
```

Result Grid

  Filter Rows:

Export: 

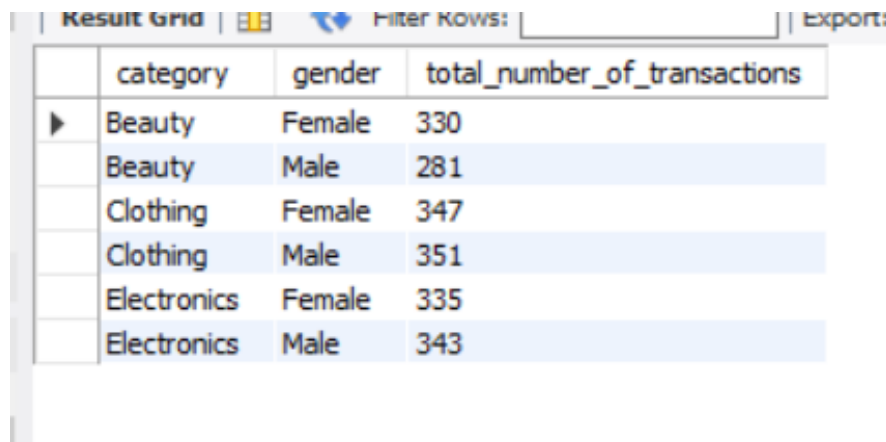
Wrap Cell Content: 

	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
	522	2022-07-09	11:00:00	52	Male	46	Beauty	3	500	145	1500
	559	2022-12-12	10:48:00	5	Female	40	Clothing	4	300	84	1200
	1522	2022-11-14	08:35:00	48	Male	46	Beauty	3	500	235	1500
	1559	2022-08-20	07:40:00	49	Female	40	Clothing	4	300	144	1200
	421	2022-04-08	08:43:00	66	Female	37	Clothing	3	500	235	1500
	1421	2022-01-17	07:07:00	59	Female	37	Clothing	3	500	185	1500
	484	2022-03-13	07:52:00	135	Female	19	Clothing	4	300	75	1200
	1484	2022-11-23	09:29:00	22	Female	19	Clothing	4	300	147	1200
	15	2022-07-01	11:50:00	75	Female	42	Electronics	4	500	210	2000
	743	2022-08-07	07:54:00	55	Female	34	Beauty	4	500	260	2000
	1015	2022-03-09	11:53:00	94	Female	42	Electronics	4	500	200	2000
	1743	2022-10-26	09:37:00	47	Female	34	Beauty	4	500	250	2000
	742	2022-03-10	06:08:00	37	Female	38	Electronics	4	500	105	2000

SALE

6-WRITE A SQL QUERY TO FIND THE TOTAL NUMBER OF TRANSACTIONS (TRANSACTION_ID) MADE BY EACH GENDER IN EACH CATEGORY.:

```
SELECT CATEGORY,GENDER,COUNT(TRANSACTIONS_ID)  
AS TOTAL_NUMBER_OF_TRANSACTIONS FROM SALES  
GROUP BY CATEGORY,GENDER  
ORDER BY CATEGORY
```

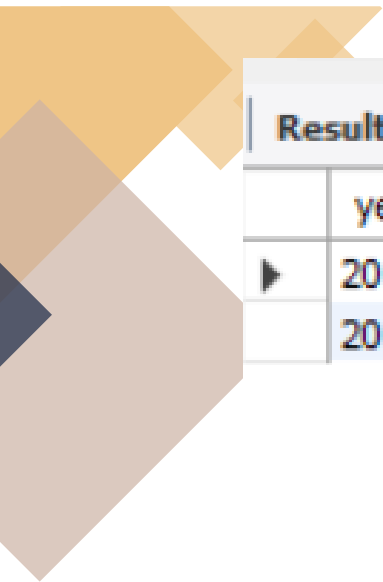


The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with 4 columns: an empty column, 'category', 'gender', and 'total_number_of_transactions'. There are 6 rows of data. The first row is expanded, showing a right-pointing triangle icon. The data is as follows:

	category	gender	total_number_of_transactions
▶	Beauty	Female	330
	Beauty	Male	281
	Clothing	Female	347
	Clothing	Male	351
	Electronics	Female	335
	Electronics	Male	343

7-WRITE A SQL QUERY TO CALCULATE THE AVERAGE SALE FOR EACH MONTH. FIND OUT BEST SELLING MONTH IN EACH YEAR:

```
WITH QUESTION AS (  
  SELECT YEAR(SALE_DATE) AS YEARLY,  
         MONTH(SALE_DATE) AS MONTHLY,  
         AVG(TOTAL_SALE) AS AVERAGE_SALE,  
         RANK() OVER (PARTITION BY YEAR(SALE_DATE) ORDER  
                     BY AVG(TOTAL_SALE) DESC) AS RANKING  
  FROM SALES  
  GROUP BY MONTH(SALE_DATE), YEAR(SALE_DATE)  
  ORDER BY YEARLY, AVERAGE_SALE DESC  
)  
SELECT YEARLY, MONTHLY, AVERAGE_SALE  
FROM QUESTION WHERE RANKING=1
```





Result Grid			
Filter Rows:			
	yearly	monthly	average_sale
▶	2022	7	541.3415
	2023	2	535.5319

SALE

8-WRITE A SQL QUERY TO FIND THE TOP 5 CUSTOMERS BASED ON THE HIGHEST TOTAL SALES **:

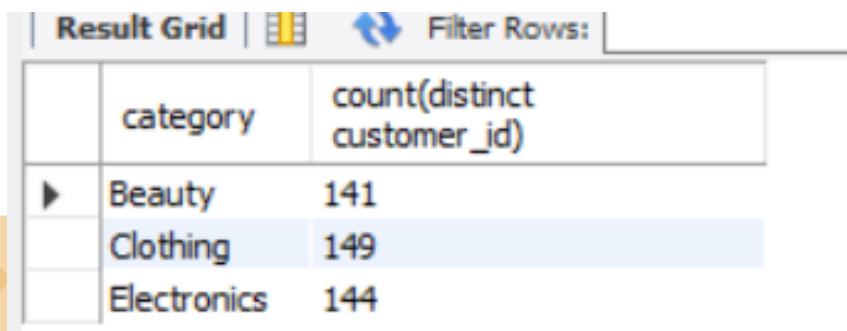
```
SELECT CUSTOMER_ID,SUM(TOTAL_SALE)
FROM SALES
GROUP BY CUSTOMER_ID
ORDER BY SUM(TOTAL_SALE) DESC
LIMIT 5
```

Result Grid   Filter Rows: <input type="text"/>	
customer_id	sum(total_sale)
3	38440
1	30750
5	30405
2	25295
4	23580

SALE

9-WRITE A SQL QUERY TO FIND THE NUMBER OF UNIQUE CUSTOMERS WHO PURCHASED ITEMS FROM EACH CATEGORY.:

```
SELECT CATEGORY,COUNT(DISTINCT CUSTOMER_ID)
FROM SALES
GROUP BY CATEGORY
```




The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'category' and 'count(distinct customer_id)'. The table has three rows: 'Beauty' with a count of 141, 'Clothing' with a count of 149, and 'Electronics' with a count of 144. The 'Clothing' row is highlighted in blue. Above the table, there are icons for a grid, a refresh button, and a 'Filter Rows:' input field.

	category	count(distinct customer_id)
▶	Beauty	141
	Clothing	149
	Electronics	144

SALE

10-WRITE A SQL QUERY TO CREATE EACH SHIFT AND NUMBER OF ORDERS (EXAMPLE MORNING <12, AFTERNOON BETWEEN 12 & 17, EVENING >17):

```
WITH NUMBER_TOTAL AS (  
  SELECT *,  
  CASE  
    WHEN HOUR(SALE_TIME) <12 THEN 'MORNING'  
    WHEN HOUR(SALE_TIME) BETWEEN 12 AND 17 THEN  
    'AFTERNOON'  
    ELSE 'EVENING'  
  END  
  AS SHIFT FROM RETAIL.SALES  
)  
SELECT SHIFT,COUNT(TRANSACTIONS_ID) FROM  
NUMBER_TOTAL  
GROUP BY SHIFT  
ORDER BY COUNT(TRANSACTIONS_ID) DESC
```

Result Grid  Filter Rows: <input type="text"/>		
	shift	count(Transactions_id)
+	Evening	1062
	Morning	548
	Afternoon	377