

MySQL Practice Questions

Data Manipulation (CRUD operations)

Functions used: - **SELECT, FROM, WHERE, LIKE, BETWEEN, AGG FUNCTION:** - MIN, MAX, UPDATE, DELETE

Insurance Dataset Download Link:

1. Show records of 'male' patient from 'southwest' region.

```
5 • SELECT * FROM insurance  
6 WHERE gender = 'male' and region = 'southwest';  
7
```

Result Grid											
Filter Rows: <input type="text"/> Export: Wrap Cell Content:											
index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	
1	2	24	male	30.1	87	No	0	No	southeast	1131.51	
2	3		male	33.3	82	Yes	0	No	southeast	1135.94	
3	4		male	33.7	80	No	0	No	northwest	1136.4	
4	5		male	34.1	100	No	0	No	northwest	1137.01	
5	6		male	34.4	96	Yes	0	No	northwest	1137.47	
6	7		male	37.3	86	Yes	0	No	northwest	1141.45	
7	8	19	male	41.1	100	No	0	No	northwest	1146.8	
8	9	20	male	43	86	No	0	No	northwest	1149.4	
15	16	32	male	30.4	86	Yes	0	No		1256.3	
16	17	35	male	34.1	90	No	0	No	southwest	1261.44	
17	18	41	male	34.4	84	No	0	No	southwest	1261.86	
18	19	49	male	35.4	97	Yes	0	No	southwest	1263.25	
19	20	48	male	33.3	91	Yes	0	No	southeast	1391.53	
21	22	34	male	31.1	96	No	0	No	southwest	1526.31	
22	23	10	male	35.5	100	Yes	0	No	southwest	1533.47	

2. Show all records having BMI in range 30 to 45 both inclusive.

```
8 • SELECT * FROM insurance  
9 WHERE bmi BETWEEN 30 and 45;
```

Result Grid											
Filter Rows: <input type="text"/> Export: Wrap Cell Content:											
index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	
1	2	24	male	30.1	87	No	0	No	southeast	1131.51	
2	3		male	33.3	82	Yes	0	No	southeast	1135.94	
3	4		male	33.7	80	No	0	No	northwest	1136.4	
4	5		male	34.1	100	No	0	No	northwest	1137.01	
5	6		male	34.4	96	Yes	0	No	northwest	1137.47	
6	7		male	37.3	86	Yes	0	No	northwest	1141.45	
7	8	19	male	41.1	100	No	0	No	northwest	1146.8	
8	9	20	male	43	86	No	0	No	northwest	1149.4	
15	16	32	male	30.4	86	Yes	0	No		1256.3	
16	17	35	male	34.1	90	No	0	No	southwest	1261.44	
17	18	41	male	34.4	84	No	0	No	southwest	1261.86	
18	19	49	male	35.4	97	Yes	0	No	southwest	1263.25	
19	20	48	male	33.3	91	Yes	0	No	southeast	1391.53	
21	22	34	male	31.1	96	No	0	No	southwest	1526.31	

3. Show minimum and maximum blood pressure of diabetic patient who smokes. Make column names as Min-BP and Max-BP respectively.

```
11 •  SELECT max(bloodpressure) as maxbp
12      , min(bloodpressure) as minbp FROM insurance
13 WHERE smoker = 'Yes';
```

Result Grid			Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:	
	maxbp	minbp						
	140	80						

4. Find no of unique patients who are not from southwest region.

```
15 •  SELECT COUNT(DISTINCT(PatientID))as count FROM insurance
16 WHERE gender = 'male' and region = 'southwest';
```

Result Grid			Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:	
	count							
	137							

5. Total claim amount from male smoker.

```
18 •  SELECT round(sum(claim)) as smoker_male_claim FROM insurance
19 WHERE gender = 'male' and smoker = 'Yes'
```

Result Grid			Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:	
	smoker_male_claim							
	3243123							

6. Select all records of south region.

```
21 •  SELECT * FROM insurance
22 WHERE region LIKE 'south%';
```

Result Grid			Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:	
	index	PatientID	age	gender	bmi	bloodpressure	diabetic	children
	0	1	39	male	23.2	91	Yes	0
	1	2	24	male	30.1	87	No	0
	2	3		male	33.3	82	Yes	0
	16	17	35	male	34.1	90	No	0
	17	18	41	male	34.4	84	No	0
	18	19	49	male	35.4	97	Yes	0
	19	20	48	male	33.3	91	Yes	0
	20	21	45	male	23.2	85	Yes	0
	21	22	34	male	31.1	96	No	0
	22	23	18	male	35.5	100	Yes	0
	23	24	42	male	36.9	93	No	0
	24	25	50	female	20.8	85	Yes	0
	26	27	36	female	26.7	97	Yes	0
	28	29	58	female	31.1	87	No	0
	29	30	25	Female	21.4	92	No	0

7. No of patient having normal blood pressure. Normal range [90-120]

```
24 •  SELECT count(1) as normal_bp FROM insurance  
25      WHERE bloodpressure BETWEEN 90 AND 120;
```

Result Grid	
	normal_bp
▶	696

8. What is the average claim amount for non-smoking female patients who are diabetic?

```
30 •  SELECT ROUND(AVG(claim)) as avg_claim FROM insurance  
31      WHERE gender = 'female' AND smoker = 'No' AND diabetic= 'Yes';
```

Result Grid	
	avg_claim
▶	8594

9. Write a SQL query to update the claim amount for the patient with PatientID = 1234 to 5000.

```
33 •  SELECT * FROM insurance  
34      WHERE PatientID = 1234;
```

Result Grid											
	index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim
▶	1233	1234	20	male	33.6	103	No	1	Yes	northeast	1000

```
33 •  UPDATE insurance  
34      SET claim = 5000  
35      WHERE PatientID = 1234;  
36  
37 •  SELECT * FROM insurance  
38      WHERE PatientID = 1234;
```

Result Grid											
	index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim
▶	1233	1234	20	male	33.6	103	No	1	Yes	northeast	5000

10. Write a SQL query to delete all records for patients who are smokers and have no children.

```
40 • DELETE FROM insurance
```

```
41 WHERE children = 0 AND smoker = 'Yes';
```

Functions used: - WHERE/BETWEEN/ROUND/GROUP BY/ HAVING/ LIMIT/ AGG FUNCTION

Sleep Efficiency Dataset Download Link

11. Find out the average sleep duration of top 15 male candidates who's sleep duration are equal to 7.5 or greater than 7.5.

```
43 • SELECT * FROM sleep_efficiency
```

```
44 WHERE gender = 'Male' AND `sleep duration` > 7.5
```

```
45 LIMIT 15;
```

	ID	Age	Gender	Bedtime	Wakeup time	Sleep duration	Sleep efficiency	REM sleep percentage
▶	5	57	Male	2021-03-13 01:00:00	2021-03-13 09:00:00	8	0.76	27
	8	53	Male	2021-08-16 00:30:00	2021-08-16 10:30:00	10	0.9	28
	11	50	Male	2021-02-28 00:30:00	2021-02-28 08:30:00	8	0.92	23
	14	28	Male	2021-02-25 01:00:00	2021-02-25 09:30:00	8.5	0.64	28
	29	23	Male	2021-11-08 21:30:00	2021-11-08 06:30:00	9	0.65	27
	32	18	Male	2021-05-24 00:30:00	2021-05-24 08:30:00	8	0.68	22
	47	55	Male	2021-04-20 00:00:00	2021-04-20 08:00:00	8	0.91	23
	56	37	Male	2021-12-04 22:00:00	2021-12-04 07:00:00	9	0.84	18
	57	54	Male	2021-09-22 01:00:00	2021-09-22 09:00:00	8	0.99	27
	66	21	Male	2021-02-21 01:00:00	2021-02-21 09:00:00	8	0.77	19
	78	46	Male	2021-01-11 02:00:00	2021-01-11 11:00:00	9	0.7	18
	81	55	Male	2021-07-14 01:30:00	2021-07-14 10:00:00	8.5	0.86	20
	92	57	Male	2021-08-20 21:30:00	2021-08-20 05:30:00	8	0.94	27
	100	65	Male	2021-12-07 01:30:00	2021-12-07 09:30:00	8	0.77	20
	102	53	Male	2021-01-28 01:00:00	2021-01-28 09:30:00	8.5	0.71	20

12. Show average deep sleep time for both Gender. Round result at 2 decimal places. Note: sleep time and deep sleep percentage will give you, deep sleep time.

```
47 •  SELECT Gender
48      , ROUND(AVG(`Deep sleep percentage`),2) as avg_deep_sleep
49  FROM sleep_efficiency
50  GROUP BY Gender;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Gender	avg_deep_sleep			
▶	Female	52.08			
	Male	53.57			

13. Find out the lowest 10th to 30th light sleep percentage records where deep sleep percentage values are between 25 to 45. Display age, light sleep percentage and deep sleep percentage columns only.

```
52 •  SELECT age
53      , `Light sleep percentage`
54      , `Deep sleep percentage`
55  FROM sleep_efficiency
56  WHERE `Deep sleep percentage` BETWEEN 25 AND 45
57  LIMIT 10,30;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	age	Light sleep percentage	Deep sleep percentage		
▶	55	45	35		
	39	47	35		
	44	48	32		
	30	45	35		
	15	45	35		
	56	45	35		
	29	47	25		
	27	52	25		
	52	47	35		
	28	52	25		

14. Group by on exercise frequency and smoking status and show average deep sleep time, average light sleep time and average rem sleep time. Note the differences in deep sleep time for smoking and non-smoking status

```

59 •  SELECT
60      `Exercise frequency`
61      , `Smoking status`
62      , AVG(`Deep sleep percentage`) as avg_deep_sleep
63      , AVG(`Light sleep percentage`) as avg_light_sleep
64      , AVG(`REM sleep percentage`) as avg_rem_sleep
65  FROM sleep_efficiency
66  GROUP BY `Exercise frequency` , `Smoking status`;

```

	Exercise frequency	Smoking status	avg_deep_sleep	avg_light_sleep	avg_rem_sleep
▶	3	Yes	48.0417	29.5833	22.3750
	3	No	57.1231	18.9846	23.8923
	1	Yes	48.3056	27.5278	24.1667
	1	No	57.2143	19.0714	23.7143
	0	No	49.4699	28.6627	21.8675
	0	Yes	44.7407	33.3704	21.8889
	5	No	56.8333	20.3333	22.8333
	2	No	59.1250	20.7083	20.1667
	2	Yes	54.2857	22.3333	23.3810

15. Group By on Awakening and show AVG Caffeine consumption, AVG Deep sleep time and AVG Alcohol consumption only for people who do exercise at least 3 days a week. Show result in descending order awakenings

```

68 •  SELECT ROUND(AVG(`Caffeine consumption`),2) as avg_caffine
69      , AVG(`Deep sleep percentage`) as avg_deep_sleep
70      , ROUND(AVG(`Alcohol consumption`),2) as avg_alcohol
71  FROM sleep_efficiency
72  WHERE `Exercise frequency` >= 3
73  GROUP BY Awakenings;

```

	avg_caffine	avg_deep_sleep	avg_alcohol
▶	27.33	59.9535	0.58
	3.12	45.1250	1.75
	13.81	55.8060	1.34
	7.81	49.9375	1.31
	5.77	49.3846	1.54

[Power Generation Dataset Download Link](#)

16. Display those power stations which have average 'Monitored Cap. (MW)' (display the values) between 1000 and 2000 and the number of occurrences of the power stations (also display these values) are greater than 200. Also sort the result in ascending order

```
78 •  SELECT `Power Station`  
79      , ROUND(AVG(`Monitored Cap.(MW)`),2)as avg_MW  
80      , count(*) as cnt  
81  FROM powergeneration  
82  WHERE `Monitored Cap.(MW)` BETWEEN 1000 AND 2000  
83  GROUP BY `Power Station` HAVING cnt > 200  
84  ORDER BY avg_mw ASC;
```

Result Grid			
	Power Station	avg_MW	cnt
▶	BRBCL	1000	431
	NTPL	1000	932
	HNPC	1040	933
	MPL	1050	933
	GMR ENERG	1050	933
	RWPL (JSW)	1080	487
	JSWBL	1080	446
	CESC	1125	933
	TOR. POW. (SUGEN)	1147.5	934

[Undergrad Collage Dataset Link](#)

17. Display top 10 lowest "value" State names of which the Year either belong to 2013 or 2017 or 2021 and type is 'Public In-State'. Also, the number of occurrences should be between 6 to 10. Display the average value up to 2 decimal places, state names and the occurrences of the states.

```
86 •  SELECT state, ROUND(AVG(VALUE),2) as avg_value, count(state) as count FROM undergrad  
87  WHERE Type = 'Public In-State' AND Year IN (2013 , 2017, 2021)  
88  GROUP BY state HAVING count > 6 AND count < 10  
89  ORDER BY avg_value ASC  
90  LIMIT 10;
```

Result Grid			
	state	avg_value	count
▶	New Mexico	5403.50	8
	Utah	5565.50	8
	Wyoming	5753.25	8
	Idaho	5805.38	8
	Florida	5940.38	8
	Oklahoma	6088.00	8
	North Carolina	6103.00	8
	Montana	6158.13	8
	Mississippi	6200.38	8
	Arkansas	6239.00	8

18. Best state in terms of low education cost (Tuition Fees) in 'Public' type university.

```
92 •   SELECT state, AVG(Value) as avg_value FROM undergrad
93     WHERE Expense = 'Fees/Tuition' AND Type LIKE 'Public%'
94     GROUP BY state
95     ORDER BY avg_value ASC
96     LIMIT 1;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	state	avg_value		
▶	Wyoming	7089.8571		

19. 2nd Costliest state for Private education in year 2021. Consider, Tuition and Room fee both.

```
98 •   SELECT state, AVG(Value) as avg_value FROM undergrad
99     WHERE Year = 2021 AND Type LIKE '%Private%'
100    GROUP BY state
101    ORDER BY avg_value DESC
102    LIMIT 1,1;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	state	avg_value		
▶	Vermont	31685.5000		

[Shipping Ecommerce Dataset Link](#)

20. Display total and average values of Discount offered for all the combinations of 'Mode_of_Shipment' (display this feature) and 'Warehouse block' (display this feature also) for all male ('M') and 'High' Product importance. Also sort the values in descending order of Mode_of_Shipment and ascending order of Warehouse block.

```
104 •   SELECT Warehouse_block  
105      , Mode_of_Shipment  
106      , SUM(Discount_offered) as total_discount  
107      , AVG(Discount_offered) as avg_discount  
108  FROM shipping_ecommerce  
109 WHERE Gender = 'M' AND Product_importance = 'High'  
110 GROUP BY Warehouse_block, Mode_of_Shipment  
111 ORDER BY Mode_of_Shipment DESC, Warehouse_block ASC;
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:				
	Warehouse_block	Mode_of_Shipment	total_discount	avg_discount
▶	A	Ship	689	16.0233
	B	Ship	785	18.2558
	C	Ship	556	11.3469
	D	Ship	753	12.3443
	F	Ship	1505	15.0500
	A	Road	279	18.6000
	B	Road	142	11.8333
	C	Road	138	9.8571
	D	Road	338	18.7778
	F	Road	350	13.4615
	A	Flight	274	17.1250
	B	Flight	222	18.5000
	C	Flight	190	10.0000

[Freedom Ranking Dataset Link](#)

Data description

Feature	Description
A	Electoral Process
B	Political Pluralism and Participation
C	Functioning of Government
D	Freedom of Expression and Belief
E	Associational and Organizational Rights
F	Rule of Law
G	Personal Autonomy and Individual Rights
CL	Civil Liberties Scores
Status	F=Free, PF=Partly Free, NF=Not Free

21. Find out top 10 countries' which have maximum A and D values.

```
113 •  SELECT c1.country, c1.edition, A, D
114    FROM country_ab as c1
115    JOIN country_cd as c2 on c1.country = c2.country
116      AND c1.edition = c2.Edition
117    ORDER BY A DESC, D DESC
118    LIMIT 10;
```

The screenshot shows a database result grid with the following columns: country, edition, A, and D. The data is sorted by A DESC and D DESC, with a limit of 10 rows. The results are as follows:

	country	edition	A	D
▶	Netherlands	2020	12	16
	Sweden	2021	12	16
	Luxembourg	2019	12	16
	Iceland	2013	12	16
	Portugal	2021	12	16
	Netherlands	2022	12	16
	New Zealand	2022	12	16
	Barbados	2021	12	16
	Palau	2020	12	16
	Slovakia	2014	12	16

22. Find out highest CL value for 2020 for every region. Also sort the result in descending order. Also display the CL values in descending order.

```
120 •  SELECT * FROM country_cl
121      WHERE Edition = 2020
122      ORDER BY CL DESC;
```

Result Grid				
	Country	Edition	CL	Status
▶	Finland	2020	60	F
	Luxembourg	2020	60	F
	Norway	2020	60	F
	Sweden	2020	60	F
	Netherlands	2020	59	F
	Canada	2020	58	F
	Ireland	2020	58	F
	Uruguay	2020	58	F
	Australia	2020	57	F
	Barbados	2020	57	F
	Belgium	2020	57	F
	Denmark	2020	57	F
	Iceland	2020	57	F

Functions used: - JOINS/CONCAT/SUBQUERY/ COMMON TABLE EXPRESSION

Sales Dataset Link

This link contains 4 Tables

- Customer
- Employees
- Product
- Sales

23. Q-3 Find top-5 most sold products.

```
134 •  SELECT p.name, sum(s.Quantity) as quantity_sold FROM sales1 as s
135     JOIN products as p on s.ProductID = p.ProductID
136     GROUP BY p.name
137     ORDER BY quantity_sold DESC
138     LIMIT 5;
```

Result Grid		
	name	quantity_sold
▶	Touring-3000 Blue, 54	38680
	Rear Derailleur	34314
	ML Mountain Frame - Black, 44	31080
	Mountain-100 Silver, 38	28080
	LL Road Seat/Saddle	27600

24. Find sales man who sold most no of products.

```
140 •  SELECT CONCAT(FirstName, ' ', MiddleInitial, ' ', LastName) as employee_name  
141      , Sum(s.Quantity) as quantity_sold  
142  FROM sales1 as s  
143  JOIN products as p on s.ProductID = p.ProductID  
144  JOIN employees as e on s.salespersonid = e.employeeid  
145  GROUP BY employee_name  
146  ORDER BY quantity_sold DESC  
147  LIMIT 1;
```

Result Grid	
<input type="button" value="Filter Rows:"/>	<input type="button" value="Export:"/> <input type="button" value="Wrap Cell Content:"/>
employee_name	quantity_sold
▶ Abraham e Bennet	115410

25. Sales man name who has most no of unique customer.

```
154 •  SELECT CONCAT(FirstName, ' ', LastName) as emp_name  
155      , COUNT(DISTINCT(CustomerID)) as unique_customer  
156  FROM sales1 as s  
157  JOIN employees as e ON s.SalesPersonID = e.EmployeeID  
158  GROUP BY emp_name  
159  ORDER BY unique_customer DESC  
160  LIMIT 1;
```

Result Grid	
<input type="button" value="Filter Rows:"/>	<input type="button" value="Export:"/> <input type="button" value="Wrap Cell Content:"/>
emp_name	unique_customer
▶ Abraham Bennet	8

26. Sales man who has generated most revenue. Show top 5.

```
162 • Ⓜ WITH CTE AS (  
163     SELECT CONCAT(FirstName, ' ', MiddleInitial, ' ', LastName) as emp_name  
164     , s.ProductID , s.Quantity  
165     , p.name, p.price  
166     , round(s.Quantity * p.price)  as revenue  
167   FROM sales1 as s  
168   JOIN products as p ON s.ProductID = p.ProductID  
169   JOIN employees as e ON s.SalesPersonID = e.EmployeeID  
170   SELECT emp_name, SUM(revenue) as total_revenue FROM CTE  
171   GROUP BY emp_name  
172   ORDER BY total_revenue DESC
```

Result Grid	
<input type="button" value="Filter Rows:"/>	<input type="button" value="Export:"/> <input type="button" value="Wrap Cell Content:"/>
emp_name	total_revenue
▶ Dirk t Stringer	114339262
Michel e DeFrance	67844269
Albert i Ringer	49155984
Abraham e Bennet	39995314
Heather c McBaden	35970708

27. List all customers who have made more than 10 purchases.

```
175 •  SELECT CONCAT(FirstName, ' ', MiddleInitial, ' ', LastName) as cust_name  
176      , count(*) as order_count FROM sales1 as s  
177  JOIN customers as c ON s.CustomerID = c.CustomerID  
178  GROUP BY cust_name HAVING order_count > 10;
```

Result Grid	
Filter Rows:	
cust_name	order_count
NULL	129
Alexandria F Peterson	25
Alicia A Chapman	38
Andrew R Hill	28
Austin M Jai	31

28. List all salespeople who have made sales to more than 5 customers.

```
181 •  SELECT CONCAT(FirstName, ' ', LastName) as emp_name  
182      , COUNT(DISTINCT(CustomerID)) as customer_count  
183  FROM sales1 as s  
184  JOIN employees as e ON s.SalesPersonID = e.EmployeeID  
185  GROUP BY emp_name HAVING customer_count > 5  
186  ORDER BY customer_count DESC;
```

Result Grid	
Filter Rows:	
emp_name	customer_count
Abraham Bennet	8

[Olympics Dataset Link](#)

29. Display the names of athletes who won a gold medal in the 2008 Olympics and whose height is greater than the average height of all athletes in the 2008 Olympics.

```
188 • ⏷ SELECT Name FROM (
189   ⏷   SELECT *, AVG(Height) OVER() as avg_height FROM olympic
190   ⏷   WHERE Year = 2008 AND Medal = 'Gold') as t
191   ⏷   WHERE Height > avg_height;
192
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Name			
▶	Jin Ziwei			
	Liu Zige			
	Meng Guanliang			
	Tang Bin			
	Tong Wen			
	Wang Liqin			
	Xi Aihua			
	Zhang Xiaoping			
	Zhang Yangyang			

30. Display the names of athletes who won a medal in the sport of basketball in the 2016 Olympics and whose weight is less than the average weight of all athletes who won a medal in the 2016 Olympics.

```
193 • ⏷ SELECT Name FROM(
194   ⏷   SELECT *, AVG(Weight) OVER() as avg_weight FROM olympic
195   ⏷   WHERE Sport = 'Basketball' AND Year = 2016 AND Medal IS NOT NULL) as t
196   ⏷   WHERE Weight < avg_weight;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Name			
▶	Anna Cruz Lebrato			
	Sylvia Domnguez Fernndez			
	Rodolfo "Rudy" Fernndez Farrs			
	Laura Gil Collado			
	Juan Carlos Navarro Feijoo			
	Astou Barro Ndour Gueye			
	Laia Palau Alts			
	Laura Quevedo Caizares			
	Leonor Rodrguez Manso			
	Sergio Rodrguez Gmez			

31. Display the names of all athletes who have won a medal in the sport of swimming in both the 2008 and 2016 Olympics.

```
198 • SELECT Name FROM olympic  
199 WHERE Medal IS NOT NULL AND Sport = 'Swimming' AND Year IN (2008,2016)
```

Name
Fu Yuanhui
Jiao Liuyang
Liu Zige
Pang Jiaying
Pang Jiaying
Pang Jiaying
Shi Jinglin
Sun Yang
Sun Yang
Sun Ye

32. Display the names of all countries that have won more than 50 medals in a single year.

```
202 • SELECT Year, Country, count(*) as medal_cnt FROM olympic  
203 WHERE Medal IS NOT NULL  
204 GROUP BY Year, Country HAVING medal_cnt > 50;
```

Year	Country	medal_cnt
1996	China	107
1984	China	74
2008	China	184
2000	China	79
2012	China	126
2016	China	113
1988	China	52
2004	China	96
1992	China	82
1920	Denmark	71

33. Display the names of all athletes who have won medals in more than one sport in the same year.

```
211 • SELECT name, group_concat(Sport) as sprot_name FROM olympic
212 WHERE medal IS NOT NULL
213 GROUP BY name, year HAVING count(distinct sport) > 1
```

Result Grid	
	name
▶	Aileen Muriel Riggin (-Soule)
	Swimming,Diving
	Alexander Viggo Jensen
	Shooting,Weightlifting,Weightlifting
	Amedee Valle Reyburn, Jr.
	Swimming,Water Polo
	Carl Schuhmann
	Gymnastics,Gymnastics,Wrestling,Gymnastics
	Charles Marius Duma Adolphe Gondouin
	Tug-Of-War,Rugby
	David Thurwell Hammond
	Swimming,Water Polo
	Dsir Alfred Mrchez
	Swimming,Water Polo
	Edward "Ned" Barrett
	Tug-Of-War,Wrestling
	Edwin Harold "Teddy" Flack
	Tennis,Athletics,Athletics
	Frank D. Kehoe
	Diving,Water Polo

Insurance Dataset Download Link:

34. How many patients have claimed more than the average claim amount for patients who are smokers and have at least one child, and belong to the southeast region?

```
216 • SELECT Count(1) as count FROM insurance
217 WHERE
218   claim > (SELECT AVG(claim) FROM insurance
219     WHERE Smoker = 'Yes' AND Region = 'southeast' AND children >= 1);
```

Result Grid	
	count
▶	88

35. How many patients have claimed more than the average claim amount for patients who are not smokers and have a BMI greater than the average BMI for patients who have at least one child?

```
221 • SELECT count(*) FROM insurance
222 WHERE claim > (SELECT AVG(claim) FROM insurance
223   WHERE smoker = 'No' AND bmi > (SELECT AVG(bmi) FROM insurance
224     WHERE children >= 1));
```

Result Grid	
	count(*)
▶	582

36. How many patients have claimed more than the average claim amount for patients who have a BMI greater than the average BMI for patients who are diabetic, have at least one child, and are from the southwest region?

```
226 •  SELECT count(1)as count FROM insurance
227   WHERE claim > (SELECT AVG(claim) FROM insurance
228     WHERE bmi > (SELECT AVG(bmi) FROM insurance
229       WHERE diabetic = 'yes' AND region = 'southwest' AND children >=1))
```

Result Grid	
count	
▶ 297	

37. What is the difference in the average claim amount between patients who are smokers and patients who are non-smokers, and have the same BMI and number of children?

```
231 •  SELECT AVG(t1.claim-t2.claim) as avg_claim FROM insurance t1
232   JOIN insurance t2 on t1.bmi = t2.bmi
233     AND t1.children = t2.children
234     AND t1.smoker <> t2.smoker;
235
```

Result Grid	
avg_claim	
▶ 0.00000000002091293206471883	

38. What are the top 5 patients who claimed the highest insurance amounts?

```
236 •  SELECT * FROM insurance
237   ORDER BY claim DESC
238   LIMIT 5;
```

Result Grid											
	index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim
▶	1337	1338	30	male	34.5	91	Yes	3	Yes	northwest	60021.4
	1336	1337	59	female	38.1	120	No	1	Yes	northeast	58571.07
	1333	1334	44	male	36.4	127	No	1	Yes	southwest	51194.56
	1332	1333	26	male	37	120	No	2	Yes	southeast	49577.66
	1331	1332	18	male	41.1	104	No	1	Yes	southeast	48970.25

39. What is the average insurance claimed by patients based on the number of children they have?

```
240 •  SELECT children, round(AVG(claim)) as avg_claim  
241      FROM insurance  
242      GROUP BY Children;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	children	avg_claim		
▶	0	7585		
	1	12631		
	2	15074		
	3	15355		
	4	13851		
	5	8786		

40. What is the highest and lowest claimed amount by patients in each region?

```
245 •  SELECT region, lowest_claim, highest_claim  
246   ⏺ FROM(SELECT *,  
247     FIRST_VALUE(claim)  
248       OVER(PARTITION BY region ORDER BY claim) as 'lowest_claim'  
249     , LAST_VALUE(claim)  
250       OVER(PARTITION BY region ORDER BY claim  
251         ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) as 'highest_claim'  
252     , ROW_NUMBER() OVER(PARTITION BY REGION) as 'row_num'  
253   FROM insurance) as t  
254   WHERE row_num = 1;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	
	region	lowest_claim	highest_claim		
▶		1252.41	1256.3		
	northeast	1694.8	58571.07		
	northwest	1136.4	60021.4		
	southeast	1121.87	49577.66		
	southwest	1261.44	51194.56		

41. What is the percentage of smokers in each age group?

```
256 •  SELECT *,  
257      (count/count(*) OVER(partition by smoker))*100 as 'smoker%' FROM  
258      (SELECT *, COUNT(*) OVER(partition by age) as 'count'  
259          FROM insurance) t  
260 WHERE smoker= 'Yes' ;
```

	Result Grid	Filter Rows:	Export:	Wrap Cell Content:									
	index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	count	smoker%
▶	1331	1332	18	male	41.1	104	No	1	Yes	southeast	48970.25	16	10.0629
	1140	1141	19	male	29.8	89	No	3	Yes	southwest	25309.49	27	16.9811
	1045	1046	19	male	25.3	95	No	2	Yes	southeast	18972.5	27	16.9811
	1021	1022	19	male	24	80	No	3	Yes	southeast	17663.14	27	16.9811
	1320	1321	19	male	42.9	104	Yes	2	Yes	southeast	47462.89	27	16.9811
	1016	1017	19	male	23	99	Yes	2	Yes	northwest	17361.77	27	16.9811
	1308	1309	20	male	42.4	104	No	3	Yes	southeast	46151.12	23	14.4654
	1138	1139	20	male	28.2	107	Yes	3	Yes	northwest	24915.22	23	14.4654
	1249	1250	20	male	32.8	99	Yes	1	Yes	northeast	39125.33	23	14.4654
	1233	1234	20	male	33.6	103	No	1	Yes	northeast	5000	23	14.4654
	1276	1277	20	male	38.4	124	Yes	3	Yes	southeast	41949.24	23	14.4654
	1038	1039	21	male	20.1	108	No	2	Yes	southeast	18767.74	17	10.6918
	1251	1252	21	male	31.4	89	Yes	1	Yes	northeast	39556.49	17	10.6918

42. For each patient, calculate the difference between their claimed amount and the average claimed amount of patients with the same number of children.

```
262 •  SELECT *,  
263      ROUND(claim - AVG(claim) OVER(PARTITION BY Children)) as claim_diff  
264      FROM insurance;
```

	Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:							
	index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	claim_diff
	29	30	35	female	31.4	93	No	0	No	southeast	1622.19	-5963
	30	31	29	male	20.4	80	Yes	0	No	northwest	1625.43	-5960
	31	32	49	male	21.8	80	No	0	No	northwest	1627.28	-5958
	32	33	21	male	22.6	96	Yes	0	No	northwest	1628.47	-5956
	33	34	52	female	36.9	81	No	0	No	southeast	1629.83	-5955
	34	35	43	female	38.2	86	Yes	0	No	southeast	1631.67	-5953
	35	36	47	female	38.3	93	No	0	No	southeast	1631.82	-5953
	36	37	48	male	25.2	98	No	0	No	northwest	1632.04	-5953
	37	38	34	male	25.6	83	No	0	No	northwest	1632.56	-5952
	38	39	28	female	39.2	87	No	0	No	southeast	1633.04	-5952

43. Show the patient with the highest BMI in each region and their respective rank.

```
262 • SELECT *,  
263     ROUND(claim - AVG(claim) OVER(PARTITION BY Children)) as claim_diff  
264     FROM insurance;  
265  
266 • SELECT * FROM (  
267     SELECT *  
268     , RANK() OVER(ORDER BY bmi DESC) as rnk  
269     , RANK() OVER(PARTITION BY region ORDER BY BMI DESC) as grp_rnk
```

	index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	rnk	grp_rnk
▶	9	10	30	male	53.1	97	No	0	No	northwest	1163.46	1	1
	1299	1300	50	male	52.6	110	No	1	Yes	southeast	44501.4	2	1
	675	676	49	female	48.1	81	Yes	2	No	northeast	9432.93	5	1
	1306	1307	43	female	47.6	112	Yes	2	Yes	southwest	46113.51	7	1
	15	16	32	male	30.4	86	Yes	0	No		1256.3	612	1

44. Calculate the difference between the claimed amount of each patient and the claimed amount of the patient who has the highest BMI in their region.

```
275 • SELECT *  
276     , ROUND(claim - FIRST_VALUE(claim)  
277             OVER(PARTITION BY REGION ORDER BY claim DESC)) as claim_diff  
278     FROM insurance;
```

	index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	claim_diff
▶	15	16	32	male	30.4	86	Yes	0	No		1256.3	0
	14	15	40	male	28.7	81	Yes	0	No		1253.94	-2
	13	14	32	male	27.6	100	No	0	No		1252.41	-4
	1336	1337	59	female	38.1	120	No	1	Yes	northeast	58571.07	0
	1326	1327	26	male	40.6	113	Yes	3	Yes	northeast	48549.18	-10022
	1325	1326	52	female	36.4	133	Yes	1	Yes	northeast	48517.56	-10054
	1322	1323	33	female	36.8	117	Yes	1	Yes	northeast	47896.79	-10674
	1310	1311	26	female	37.1	95	No	3	Yes	northeast	46255.11	-12316
	1301	1302	60	female	35	92	Yes	2	Yes	northeast	44641.2	-13930
	1290	1291	47	male	41.9	140	Yes	3	Yes	northeast	43753.34	-14818

45. For each patient, find the maximum BMI value among their next three records (ordered by age).

```
280 •  SELECT *
281      , MAX(bmi) OVER(ORDER BY age ROWS BETWEEN 1 FOLLOWING AND 3 FOLLOWING) as highest_bmi
282  FROM insurance
283  ORDER BY age;
```

index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	highest_bmi
162	163	47	male	17.7	92	Yes	0	No	northwest	2680.95	26.7
163	164	28	male	38.1	81	Yes	0	No	southeast	2689.5	41.7
164	165	52	female	28.1	90	Yes	0	No	northwest	2690.11	33.1
165	166	30	male	31.1	91	No	0	No	northwest	2699.57	34.5
166	167	51	female	24.5	80	Yes	1	No	northwest	2709.11	34.6
167	168	58	female	24.6	87	Yes	1	No	northwest	2709.24	31.1
168	169	26	female	25.7	88	No	1	No	northwest	2710.83	35.6
169	170	41	female	31.8	88	Yes	1	No	northwest	2719.28	35.9
170	171	46	male	26.2	84	No	0	No	southeast	2721.32	42.1
171	172	23	male	30.6	96	No	0	No	southeast	2727.4	43

46. For each patient, find the rolling average of the last 2 claims.

```
285 •  SELECT *
286      , ROUND(AVG(claim) OVER(ROWS BETWEEN 2 PRECEDING AND 1 PRECEDING)) as rolling_avg
287  FROM insurance;
```

index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	rolling_avg
0	1	39	male	23.2	91	Yes	0	No	southeast	1121.87	NULL
1	2	24	male	30.1	87	No	0	No	southeast	1131.51	1122
2	3		male	33.3	82	Yes	0	No	southeast	1135.94	1127
3	4		male	33.7	80	No	0	No	northwest	1136.4	1134
4	5		male	34.1	100	No	0	No	northwest	1137.01	1136
5	6		male	34.4	96	Yes	0	No	northwest	1137.47	1137
6	7		male	37.3	86	Yes	0	No	northwest	1141.45	1137
7	8	19	male	41.1	100	No	0	No	northwest	1146.8	1139
8	9	20	male	43	86	No	0	No	northwest	1149.4	1144
9	10	30	male	53.1	97	No	0	No	northwest	1163.46	1148

47. Find the first claimed insurance value for male and female patients, within each region order the data by patient age in ascending order, and only include patients who are non-diabetic and have a BMI value between 25 and 30.

```
289 •  WITH CTE AS (
290   SELECT * FROM insurance
291   WHERE diabetic = 'No' AND bmi BETWEEN 25 AND 30)
292  SELECT * FROM (
293    SELECT *
294      , FIRST_VALUE(claim) OVER(PARTITION BY region, gender ORDER BY age) as first_claim
295      , ROW_NUMBER() OVER(partition by region, gender order by age) as 'row_num'
296  FROM CTE) as t
297  WHERE row_num = 1;
```

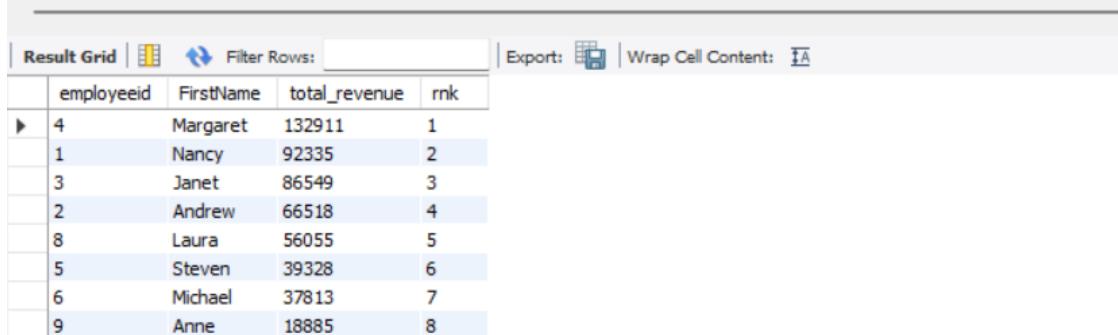
index	PatientID	age	gender	bmi	bloodpressure	diabetic	children	smoker	region	claim	first_claim	row_num
13	14	32	male	27.6	100	No	0	No		1252.41	1252.41	1
41	42	18	male	27.8	93	No	0	No	northwest	1635.73	1635.73	1
284	285	25	female	26.8	98	No	2	No	northwest	4189.11	4189.11	1
462	463	18	male	29.8	97	No	2	No	southeast	6406.41	6406.41	1
586	587	26	female	29.8	84	No	2	No	southeast	8219.2	8219.2	1
702	703	26	female	25.8	110	No	1	No	southwest	9861.03	9861.03	1
719	720	27	female	27.1	103	No	1	No	northeast	10106.13	10106.13	1
1140	1141	19	male	29.8	89	No	3	Yes	southwest	25309.49	25309.49	1
1185	1186	18	male	29.7	103	No	2	No	northeast	32108.66	32108.66	1

Functions used: - **WINDOW_FUNCTIONS:** RANK, ROW_NUMBER, CUMSUM, CUMAVG,
PERCENT_RANK/JOINS /SUBQUERY/ COMMON TABLE EXPRESSION

North Wind Dataset Link

48. Rank Employee in terms of revenue generation. Show employee id, first name, revenue, and rank

```
301 • SELECT *, RANK() OVER(ORDER BY total_revenue DESC) as rnk FROM(
302     SELECT e.employeeid, FirstName, SUM(od.revenue) as total_revenue
303     FROM employees AS e
304     JOIN orders AS o ON e.EmployeeID = o.EmployeeID
305     JOIN order_details AS od ON od.orderid = o.orderID
306     GROUP BY employeeid, FirstName) as t
307     ORDER BY total_revenue DESC;
```



The screenshot shows a database query results grid. At the top, there are navigation buttons for 'Result Grid' (highlighted), 'Export', and 'Wrap Cell Content'. Below the header row, there are eight data rows. The columns are labeled 'employeeid', 'FirstName', 'total_revenue', and 'rnk'. The data is as follows:

	employeeid	FirstName	total_revenue	rnk
▶	4	Margaret	132911	1
	1	Nancy	92335	2
	3	Janet	86549	3
	2	Andrew	66518	4
	8	Laura	56055	5
	5	Steven	39328	6
	6	Michael	37813	7
	9	Anne	18885	8

49. Show All products cumulative sum of units sold each month.

```
309 • SELECT *
310   , SUM(quantity) OVER(PARTITION BY ProductID
311     ORDER BY month ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) as cumsum
312   FROM(
313     SELECT ProductID, MONTH(OrderDate)as month
314       , SUM(Quantity) as quantity
315     FROM orders as o
316   JOIN order_details as od ON o.OrderID = od.OrderID
317   GROUP BY ProductID, MONTH(OrderDate)
318   ORDER BY ProductID) t;
```

Result Grid			
ProductID	month	quantity	cumsum
1	1	34	34
1	3	15	49
1	4	40	89
1	5	8	97
1	6	10	107
1	7	29	136
1	8	63	199
1	9	20	219
1	11	27	246
1	12	15	261
2	1	60	60
2	2	55	115
2	3	60	175

50. Show Percentage of total revenue by each supplier

```
321 • SELECT *, (revenue/SUM(revenue) OVER()) * 100 as '%total_revenue' FROM (
322   SELECT s.SupplierID, sum(Revenue) as revenue FROM Suppliers as s
323   JOIN products as p ON s.SupplierID = p.SupplierID
324   JOIN order_details as od ON od.ProductID = p.ProductID
325   GROUP BY s.SupplierID
326   ORDER BY s.SupplierID) as t
327   GROUP BY SupplierID;
```

Result Grid		
SupplierID	revenue	%total_revenue
1	12505	2.1197
2	17817	3.0202
3	13464	2.2823
4	11360	1.9256
5	9357	1.5861
6	8600	1.4578
7	52197	8.8479
8	22744	3.8553
9	5074	0.8601
10	1831	0.3104

51. Show Percentage of total orders by each supplier

```
344 • SELECT *, (sum_order/SUM(sum_order) OVER()) * 100 as '%total_revenue' FROM (
345     SELECT s.SupplierID, sum(OrderID) as sum_order FROM Suppliers as s
346     JOIN products as p ON s.SupplierID = p.SupplierID
347     JOIN order_details as od ON od.ProductID = p.ProductID
348     GROUP BY s.SupplierID
349     ORDER BY s.SupplierID) as t
350     GROUP BY SupplierID;
```

Result Grid		
SupplierID	sum_order	%total_revenue
1	375464	3.5979
2	374851	3.5920
3	156806	1.5026
4	229432	2.1985
5	250993	2.4051
6	344860	3.3046
7	729603	6.9914
8	657540	6.3008

[Drug Dataset Link](#)

52. For each condition, what is the average satisfaction level of drugs that are "On Label" vs "Off Label"?

```
353 • SELECT `Condition`, Indication
354     , ROUND(AVG(Satisfaction)) as avg_satisfaction FROM drug_clean
355     GROUP BY `Condition`, Indication
```

Result Grid			
Condition	Indication	avg_satisfaction	
Acute Bacterial Sinusitis	On Label	3	
Acute Bacterial Sinusitis	Off Label	3	
Atopic Dermatitis	On Label	4	
Atopic Dermatitis	Off Label	3	
Atopic Dermatitis		4	
Bacterial Conjunctivitis	On Label	3	
Bacterial Conjunctivitis	Off Label	2	
Bacterial Urinary Tract Infection	On Label	3	
Bacterial Urinary Tract Infection	Off Label	4	
Infantile Autism	On Label	2	

53. For each drug type (RX, OTC, RX/OTC), what is the average ease of use and satisfaction level of drugs with a price above the average for their type?

```
360 • SELECT Type
361      , ROUND(AVG(EaseOfUse),2) as avg_ease
362      , ROUND(AVG(Satisfaction),2) as avg_satisfaction
363 FROM (
364     SELECT Type, Price
365           , EaseOfUse, Satisfaction
366           , ROUND(AVG(Price) OVER(PARTITION BY Type),2) as avg_price
367     FROM drug_clean) as t
368 WHERE price > avg_price
369 GROUP BY Type;
```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	Type	avg_ease	avg_satisfaction			
▶	OTC	3.82	3.37			
▶	RX	3.82	3.15			
	RX/OTC	3.51	2.9			

54. What is the cumulative distribution of EaseOfUse ratings for each drug type (RX, OTC, RX/OTC)? Show the results in descending order by drug type and cumulative distribution. (Use the built-in method and the manual method by calculating on your own. For the manual method, use the "ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW" and see if you get the same results as the built-in method.)

```
371 • SELECT Type, EaseOfUse,  
372      CUME_DIST() OVER(PARTITION BY Type ORDER BY EaseofUse) as cumulative_dist,  
373      count(1) OVER(PARTITION BY Type ORDER BY EaseofUse  
374      ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) * 0.1/
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Type	EaseOfUse	cumulative_dist	percentile_score	
OTC	5	1	0.08720	
OTC	5	1	0.08780	
OTC	5	1	0.08841	
OTC	5	1	0.08902	
OTC	5	1	0.08963	
OTC	5	1	0.09024	
OTC	5	1	0.09085	
OTC	5	1	0.09146	
OTC	5	1	0.09207	
OTC	5	1	0.09268	
OTC	5	1	0.07866	
OTC	5	1	0.10000	
OTC	5	1	0.09939	

55. What is the median satisfaction level for each medical condition? Show the results in descending order by median satisfaction level. (Don't repeat the same rows of your result.)

```

379 •   SELECT `Condition`, AVG(Satisfaction) as satisfaction
380   FROM(
381     SELECT *, row_number() OVER(PARTITION BY `Condition` ORDER by `Satisfaction`)as rn,
382           count(1) OVER(PARTITION BY `Condition`) as total_count
383   FROM drug_clean) as t
384   WHERE rn BETWEEN total_count*1.0/2 AND total_count*1.0/2 + 1
385   GROUP BY `Condition`
386   ORDER BY satisfaction DESC;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Condition	satisfaction		
▶	flatulence	4.165		
	biliary calculus	4.1525		
	meniere's disease	4.067		
	back pain	3.8841304350000003		
	chickenpox	3.88		
	Sleepiness Due To Obstructive Sleep Apnea	3.785		
	furunculosis	3.77		
	Atopic Dermatitis	3.69		
	herpes zoster	3.5425		
	hemorrhoids	3.5		
	gout	3.5		
	fever	3.5		
	vertigo	3.5		

56. What is the running average of the price of drugs for each medical condition? Show the results in ascending order by medical condition and drug name.

```

389 •   SELECT `Condition`, Drug, Price
390   , ROUND(AVG(Price) OVER(partition by `condition`  

391           rows between unbounded preceding and current row)) as running_avg
392   FROM drug_clean
393   ORDER BY `Condition` ASC, Drug ASC;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Condition	Drug	Price	running_avg	
▶ Acute Bacterial Sinusitis	Amoxicillin	12.59	13	
Acute Bacterial Sinusitis	Amoxicillin-Pot Clavulanate	287.37	150	
Acute Bacterial Sinusitis	Amoxicillin-Pot Clavulanate	70.60857143	124	
Acute Bacterial Sinusitis	Ampicillin	12.59	96	
Acute Bacterial Sinusitis	Ampicillin	125.24	102	
Acute Bacterial Sinusitis	Ampicillin Sodium	143.215	109	
Acute Bacterial Sinusitis	Azithromycin	32.1	98	
Acute Bacterial Sinusitis	Clarithromycin	99.49	98	
Acute Bacterial Sinusitis	Erythromycin	125.24	101	
Acute Bacterial Sinusitis	Erythromycin Ethylsuccinate	332.63	124	
Acute Bacterial Sinusitis	Erythromycin Stearate	161.19	127	
Acute Bacterial Sinusitis	Levofloxacin	172.99	131	
Acute Bacterial Sinusitis	Moxifloxacin	74.99	127	

57. What is the percentage change in the number of reviews for each drug between the previous row and the current row? Show the results in descending order by percentage change.

```

396 •  SELECT `Condition`, Drug, Reviews
397   , (Reviews - LAG(Reviews) OVER(PARTITION BY `Condition`, Drug
398     ORDER BY Reviews DESC)*100)/(LAG(Reviews) OVER(PARTITION BY `Condition`, Drug
399     ORDER BY Reviews DESC)) as '%change'
400   FROM drug_clean
401   ORDER BY '%change' DESC;

```

Condition	Drug	Reviews	%change
Acute Bacterial Sinusitis	Amoxicillin	86.29411765	NULL
Acute Bacterial Sinusitis	Amoxicillin-Pot Clavulanate	267.2857143	NULL
Acute Bacterial Sinusitis	Amoxicillin-Pot Clavulanate	43	-99.83912346339716
Acute Bacterial Sinusitis	Ampicillin	15	NULL
Acute Bacterial Sinusitis	Ampicillin	1	-99.93333333333334
Acute Bacterial Sinusitis	Ampicillin Sodium	1	NULL
Acute Bacterial Sinusitis	Azithromycin	490	NULL
Acute Bacterial Sinusitis	Clarithromycin	141.6	NULL
Acute Bacterial Sinusitis	Erythromycin	41	NULL
Acute Bacterial Sinusitis	Erythromycin Ethylsuccinate	3.666666667	NULL
Acute Bacterial Sinusitis	Erythromycin Stearate	5.666666667	NULL
Acute Bacterial Sinusitis	Levofloxacin	546.6	NULL
Acute Bacterial Sinusitis	Moxifloxacin	259.3333333	NULL

58. What is the percentage of total satisfaction level for each drug type (RX, OTC, RX/OTC)? Show the results in descending order by drug type and percentage of total satisfaction.

```

403 •  WITH temp_df as(
404   SELECT Type
405     , ((SUM(Satisfaction) OVER(PARTITION BY TYPE))/(
406       SUM(Satisfaction) OVER()) * 100 as pct_total_satisfaction
407   FROM drug_clean
408   ORDER BY Type ASC, pct_total_satisfaction DESC)
409   SELECT Type, ROUND(AVG(pct_total_satisfaction)) as avg_satisfaction
410   FROM temp_df
411   GROUP BY Type;

```

Type	avg_satisfaction
OTC	27
RX	69
RX/OTC	4

59. What is the cumulative sum of effective ratings for each medical condition and drug form combination? Show the results in ascending order by medical condition, drug form and the name of the drug.

```
416 • SELECT `Condition`, drug, Form
417   , ROUND(SUM(Effective) OVER(PARTITION BY `Condition`, drug ORDER BY drug
418     ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW),2) as cum_sum_effective
419   FROM drug_clean
420   ORDER BY `Condition` ASC, Form ASC, Drug ASC;
```

Condition	drug	Form	cum_sum_effective
Acute Bacterial Sinusitis	Amoxicillin	Capsule	3.66
Acute Bacterial Sinusitis	Ampicillin	Capsule	3
Acute Bacterial Sinusitis	Amoxicillin-Pot Clavulanate	Liquid (Drink)	3.29
Acute Bacterial Sinusitis	Erythromycin Ethylsuccinate	Liquid (Drink)	3.28
Acute Bacterial Sinusitis	Moxifloxacin	Liquid (Drink)	2.87
Acute Bacterial Sinusitis	Moxifloxacin-Sod.Chloride(Iso)	Liquid (Drink)	2.33
Acute Bacterial Sinusitis	Amoxicillin-Pot Clavulanate	Tablet	6.25
Acute Bacterial Sinusitis	Ampicillin	Tablet	6
Acute Bacterial Sinusitis	Ampicillin Sodium	Tablet	3
Acute Bacterial Sinusitis	Azithromycin	Tablet	3.34

60. What is the rank of the average ease of use for each drug type (RX, OTC, RX/OTC)? Show the results in descending order by rank and drug type.

```
424 • SELECT Type
425   ,AVG(EaseOfUse) as average_ease_of_use
426   ,RANK() OVER(ORDER BY AVG(EaseOfUse) DESC) as rnk
427   FROM drug_clean
428   GROUP BY Type
```

Type	average_ease_of_use	rnk
OTC	4.138997630469513	1
RX/OTC	3.9018981481481485	2
RX	3.840152400675674	3

61. For each condition, what is the average effectiveness of the top 3 most reviewed drugs?

```
430 •  SELECT `Condition`, drug, avg_effectiveness, `rank`  
431   FROM ( SELECT  
432     `Condition`, Drug, AVG(Effective) avg_effectiveness,  
433     ROW_NUMBER() OVER (PARTITION BY `Condition` ORDER BY SUM(Reviews) DESC) AS 'rank'  
434   FROM drug_clean  
435   GROUP BY `Condition`, Drug ) as ranked_drugs  
436 WHERE `rank` <= 3  
437 order by `Condition`, `rank`;
```

Result Grid				
	Condition	drug	avg_effectiveness	rank
▶	Acute Bacterial Sinusitis	Levofloxacin	2.642	1
	Acute Bacterial Sinusitis	Azithromycin	3.341666667	2
	Acute Bacterial Sinusitis	Amoxicillin-Pot Clavulanate	3.1264285715	3
	adenocarcinoma of pancreas	Capecitabine	3.75	1
	adenocarcinoma of pancreas	Gemcitabine	3.345	2
	adenocarcinoma of pancreas	Cisplatin	3.635	3
	Atopic Dermatitis	Prednisone	3.8287500000000003	1
	Atopic Dermatitis	Cetirizine	3.6841666665000004	2
	Atopic Dermatitis	Methylprednisolone	3.201666667	3
	back pain	Capsaicin-Menthol, Capsaicin	2.47	1
	back pain	Capsaicin-Methyl Sal-Menthol	4.96	2
	back pain	Capsaicin	3.332222222	3
	Bacterial Conjunctivitis	Moxifloxacin	3.47	1