Learn how to write SQL Queries(Practice Complex SQL Queries)

Learning SQL syntax is very easy but getting comfortable in writing SQL Queries, especially the complex SQL Queries can be tricky and will need a lot of practice.

In this blog, I have listed below **9 SQL Queries** which should help you to practice intermediate to complex SQL queries.

You will find below SQL Questions along with the data and table structure required to solve each SQL question. The SQL Query to solve these questions will be attached to an .txt file. You can simple download the file for each question to get the solved SQL Queries.

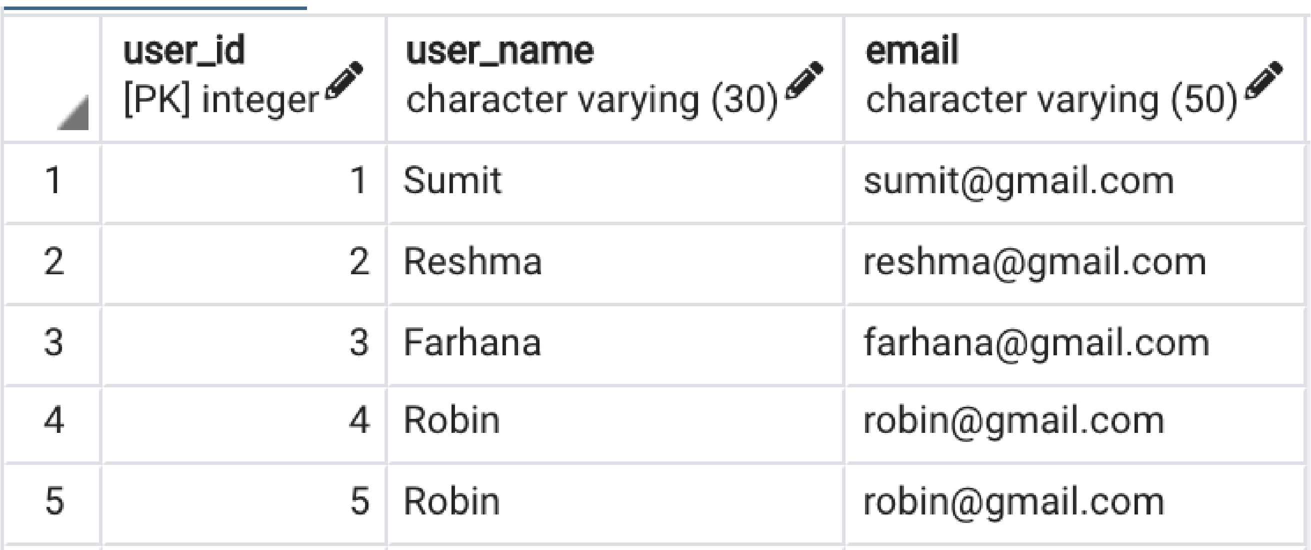
*** *Note: Please note, I have used PostgreSQL database to executed all of these queries. I believe these queries would work just fine with any other major RDBMS such as Oracle, MySQL, Microsoft SQL Server. However, if you find any query not working in your RDBMS, then leave a comment below so I could help.*

1. Write a SQL Query to fetch all the duplicate records in a table.

Table Name: USERS

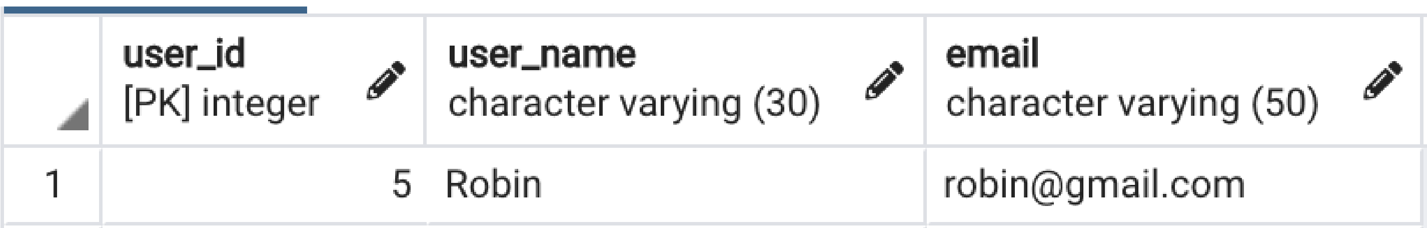
Note: Record is considered duplicate if a user name is present more than once.

Approach: Partition the data based on user name and then give a row number to each of the partitioned user name. If a user name exists more than once then it would have multiple row numbers. Using the row number which is other than 1, we can identify the duplicate records.



	user_id [PK] integer	user_name character varying (30)	email character varying (50)
1	1	Sumit	sumit@gmail.com
2	2	Reshma	reshma@gmail.com
3	3	Farhana	farhana@gmail.com
4	4	Robin	robin@gmail.com
5	5	Robin	robin@gmail.com

USERS



	user_id [PK] integer	user_name character varying (30)	email character varying (50)
1	5	Robin	robin@gmail.com

Expected Output

There are several ways to write this query. Such as either using the CTID field in PostgreSQL or by using the ROWID field in Oracle, MySQL, Microsoft SQL Server etc but a simpler way to write this query would be using window function.

Try to write this query yourself before looking the query I have written to solve it.

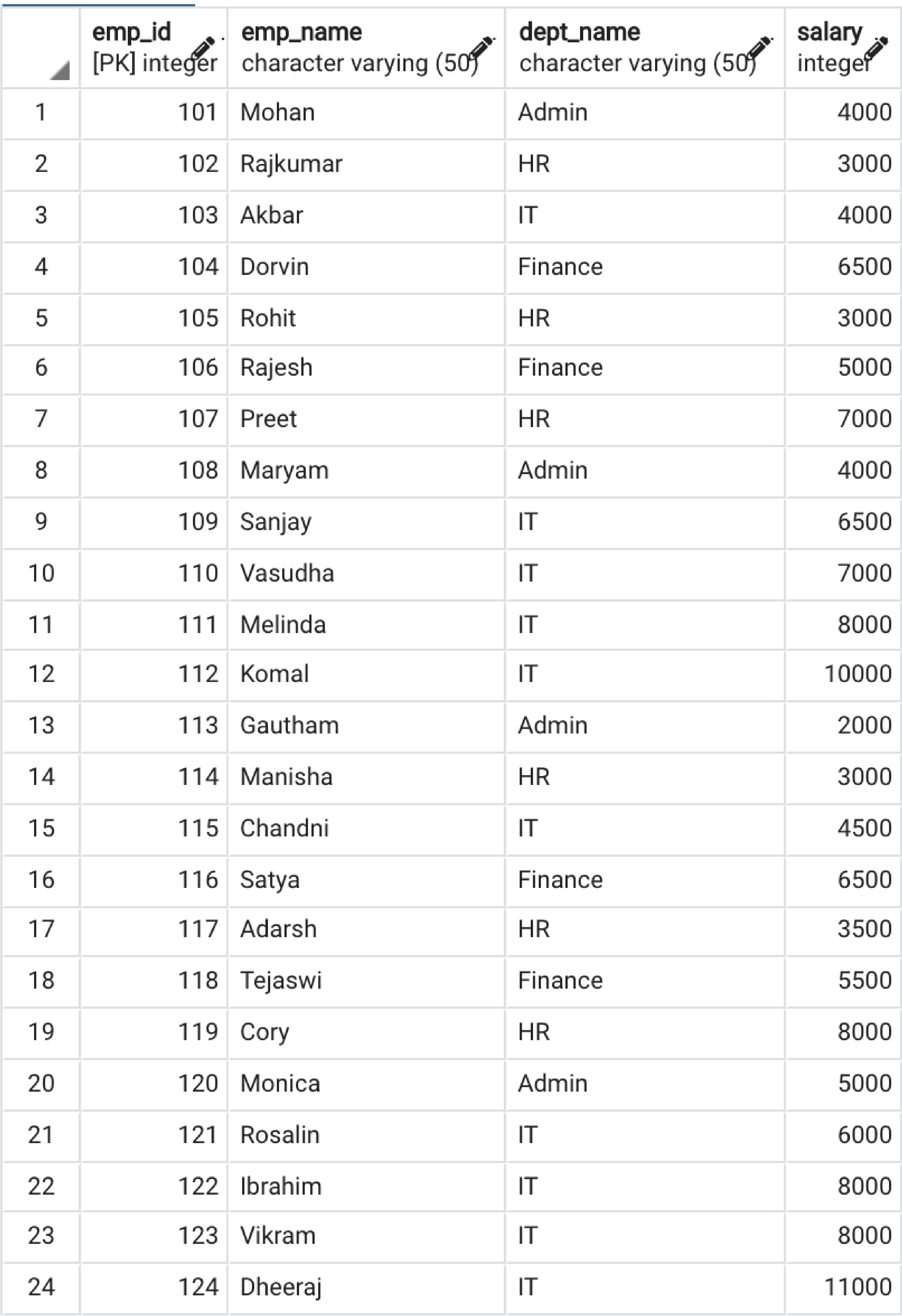
Click on the download button below to download the .txt file which will have the table structure, table data and the solved SQL Query.

Download Script - Query 1

2. Write a SQL query to fetch the second last record from employee table.

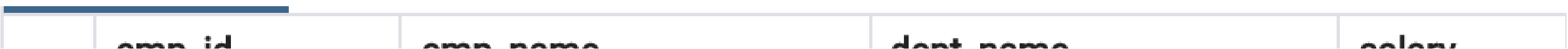
Table Name: EMPLOYEE

Approach: Using window function sort the data in descending order based on employee id. Provide a row number to each of the record and fetch the record having row number as 2.

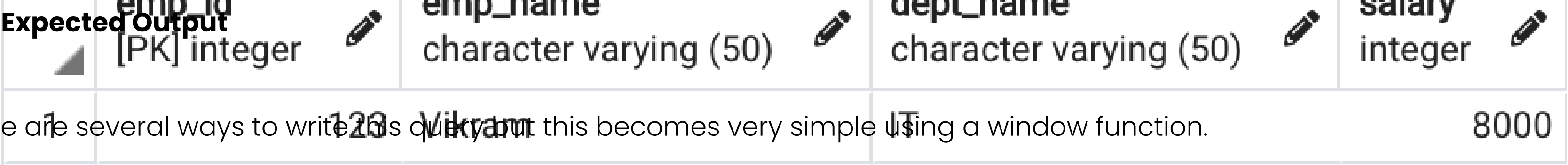


	emp_id [PK] integer	emp_name character varying (50)	dept_name character varying (50)	salary integer
1	101	Mohan	Admin	4000
2	102	Rajkumar	HR	3000
3	103	Akbar	IT	4000
4	104	Dorvin	Finance	6500
5	105	Rohit	HR	3000
6	106	Rajesh	Finance	5000
7	107	Preet	HR	7000
8	108	Maryam	Admin	4000
9	109	Sanjay	IT	6500
10	110	Vasudha	IT	7000
11	111	Melinda	IT	8000
12	112	Komal	IT	10000
13	113	Gautham	Admin	2000
14	114	Manisha	HR	3000
15	115	Chandni	IT	4500
16	116	Satya	Finance	6500
17	117	Adarsh	HR	3500
18	118	Tejaswi	Finance	5500
19	119	Cory	HR	8000
20	120	Monica	Admin	5000
21	121	Rosalin	IT	6000
22	122	Ibrahim	IT	8000
23	123	Vikram	IT	8000
24	124	Dheeraj	IT	11000

EMPLOYEE



Expected Output



	emp_id [PK] integer	emp_name character varying (50)	dept_name character varying (50)	salary integer
1	123	Vikram	IT	8000

Again, there are several ways to write this query but this becomes very simple using a window function.

Try to write this query yourself before looking the query I have written to solve it.

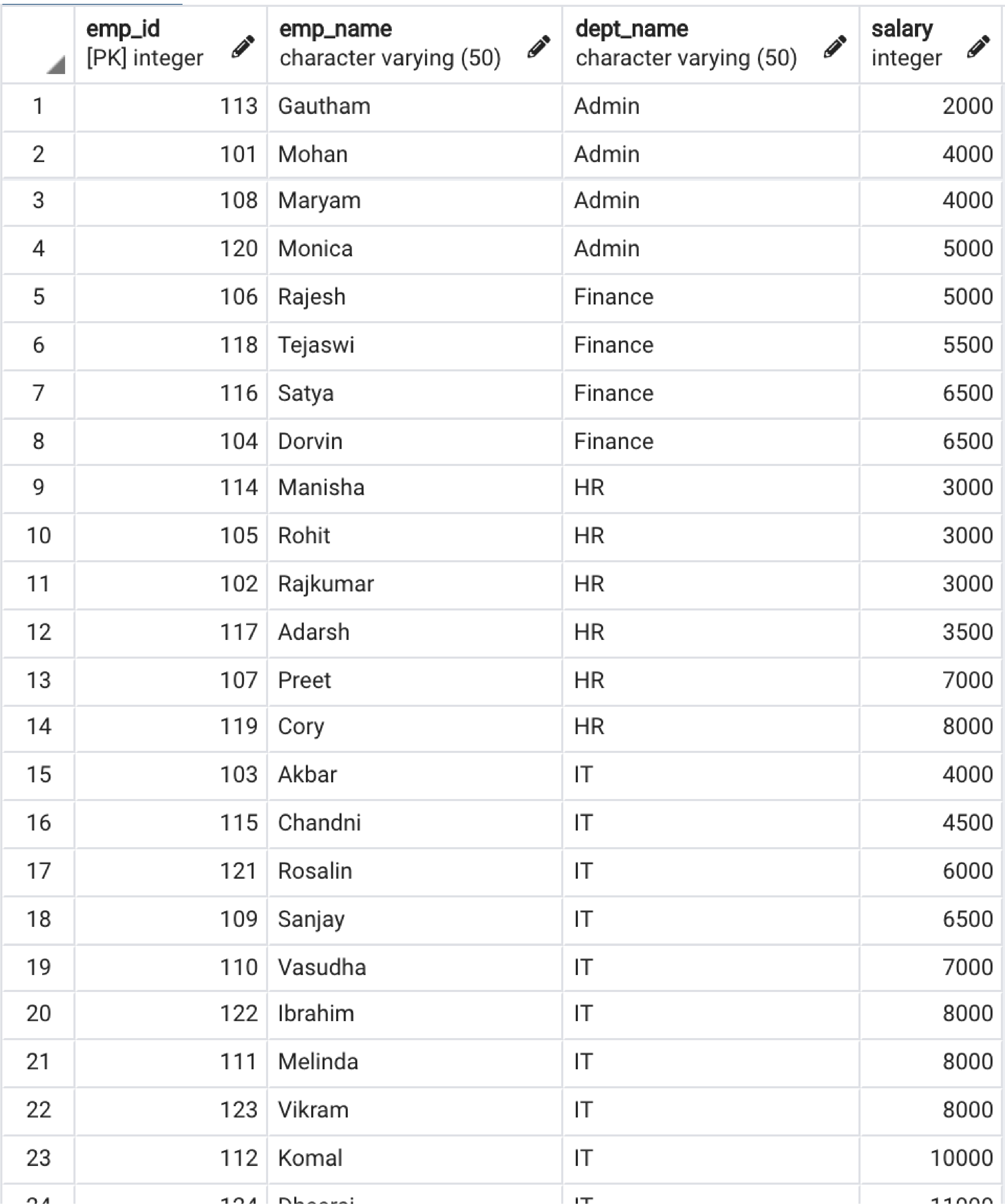
Click on the download button below to download the .txt file which will have the table structure, table data and the solved SQL Query.

Download Script - Query 2

3. Write a SQL query to display only the details of employees who either earn the highest salary or the lowest salary in each department from the employee table.

Table Name: EMPLOYEE

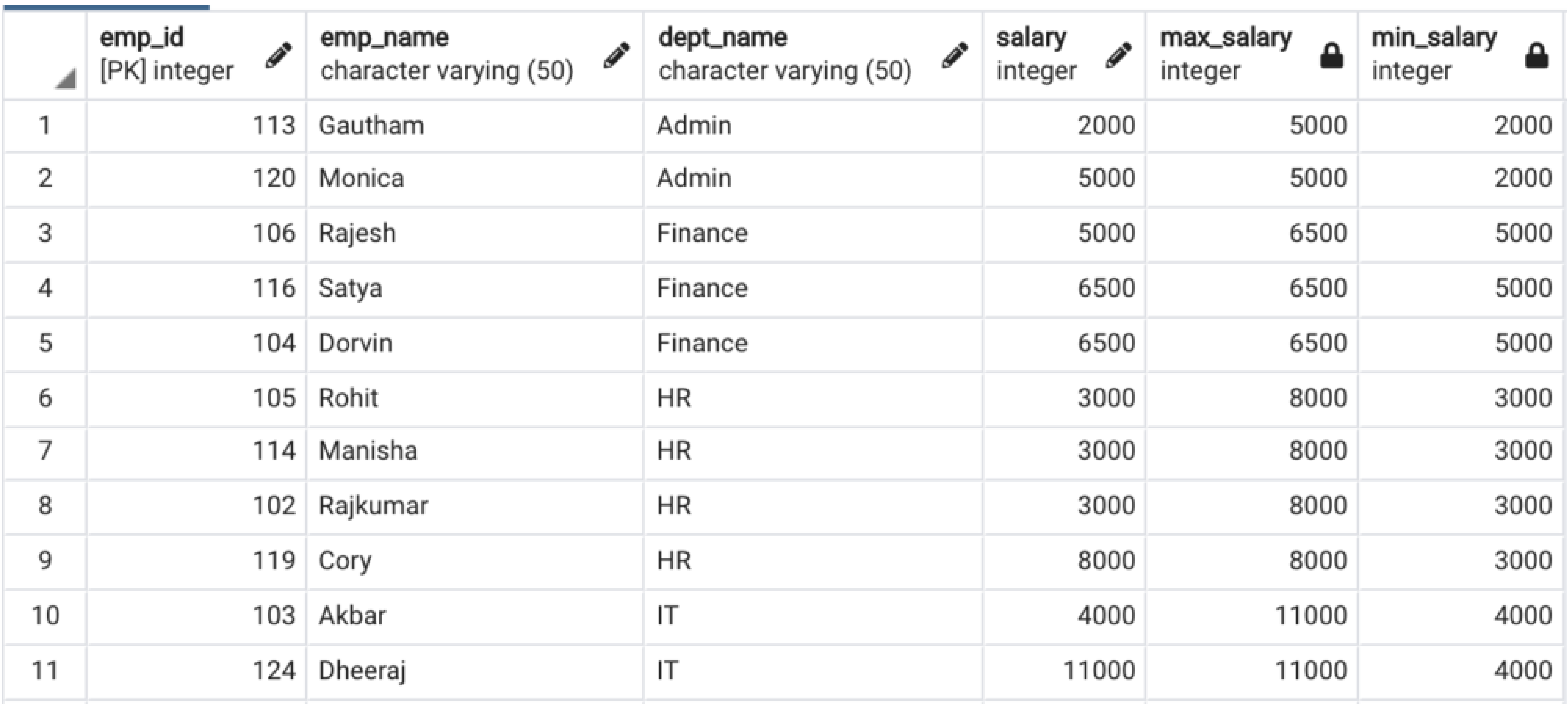
Approach: Write a sub query which will partition the data based on each department and then identify the record with maximum and minimum salary for each of the partitioned department. Finally, from the main query fetch only the data which matches the maximum and minimum salary returned from the sub query.



	emp_id [PK] integer	emp_name character varying (50)	dept_name character varying (50)	salary integer
1	113	Gautham	Admin	2000
2	101	Mohan	Admin	4000
3	108	Maryam	Admin	4000
4	120	Monica	Admin	5000
5	106	Rajesh	Finance	5000
6	118	Tejaswi	Finance	5500
7	116	Satya	Finance	6500
8	104	Dorvin	Finance	6500
9	114	Manisha	HR	3000
10	105	Rohit	HR	3000
11	102	Rajkumar	HR	3000
12	117	Adarsh	HR	3500
13	107	Preet	HR	7000
14	119	Cory	HR	8000
15	103	Akbar	IT	4000
16	115	Chandni	IT	4500
17	121	Rosalin	IT	6000
18	109	Sanjay	IT	6500
19	110	Vasudha	IT	7000
20	122	Ibrahim	IT	8000
21	111	Melinda	IT	8000
22	123	Vikram	IT	8000
23	112	Komal	IT	10000
24	124	Dheeraj	IT	11000

24	124	Dheeraj	IT	11000

EMPLOYEE



	emp_id [PK] integer	emp_name character varying (50)	dept_name character varying (50)	salary integer	max_salary integer	min_salary integer
1	113	Gautham	Admin	2000	5000	2000
2	120	Monica	Admin	5000	5000	2000
3	106	Rajesh	Finance	5000	6500	5000
4	116	Satya	Finance	6500	6500	5000
5	104	Dorvin	Finance	6500	6500	5000
6	105	Rohit	HR	3000	8000	3000
7	114	Manisha	HR	3000	8000	3000
8	102	Rajkumar	HR	3000	8000	3000
9	119	Cory	HR	8000	8000	3000
10	103	Akbar	IT	4000	11000	4000
11	124	Dheeraj	IT	11000	11000	4000

Expected Output

Again there are many way to do this and also we can use a few window functions to achieve the same result. As an added challenge, try out solving this query using a different window function and then comment out your query.

Click on the download button below to download the .txt file which will have the table structure, table data and the solved SQL Query.

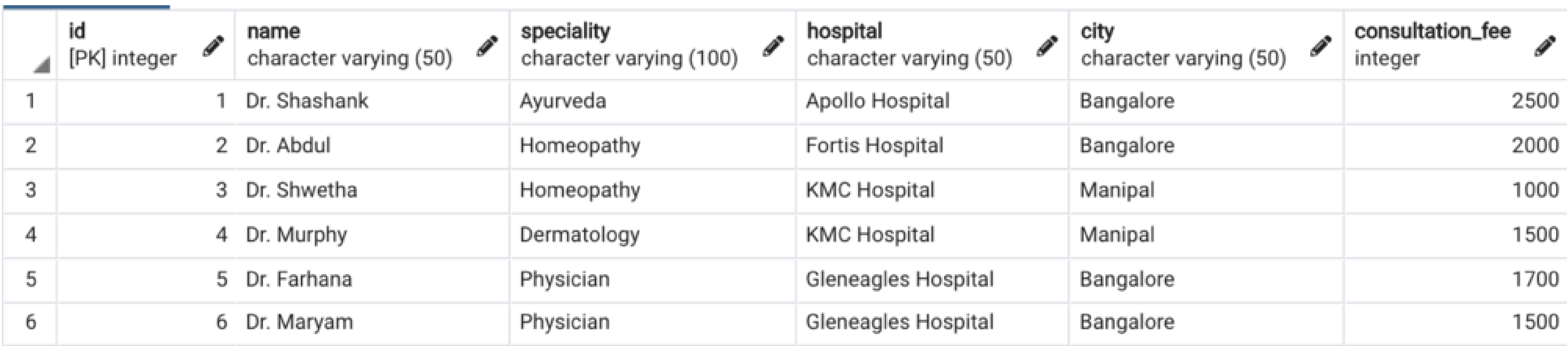
Download Script - Query 3

4. From the doctors table, fetch the details of doctors who work in the same hospital but in different specialty.

Table Name: DOCTORS

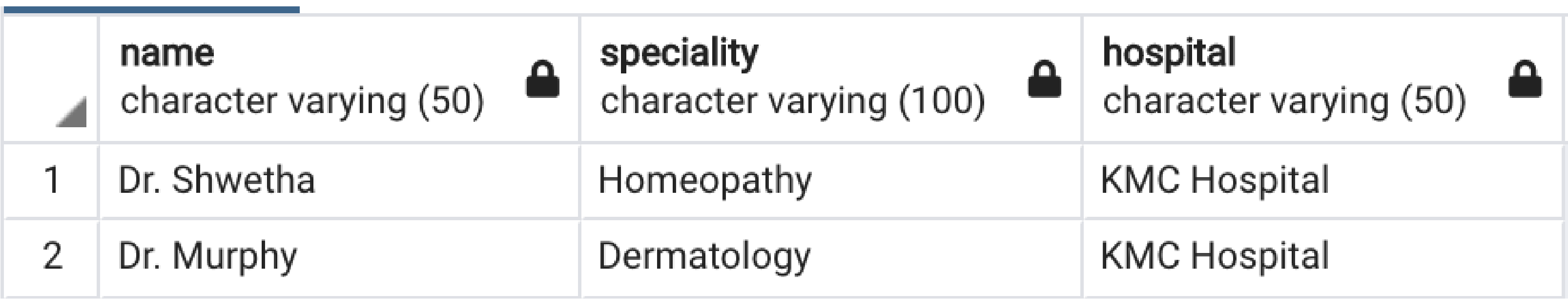
Approach: Use self join to solve this problem. Self join is when you join a table to itself.

Additional Query: Write SQL query to fetch the doctors who work in same hospital irrespective of their specialty.



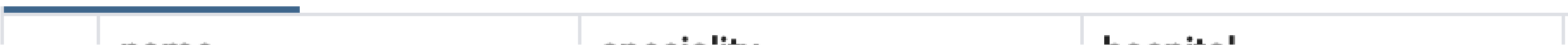
	id [PK] integer	name character varying (50)	speciality character varying (100)	hospital character varying (50)	city character varying (50)	consultation_fee integer
1	1	Dr. Shashank	Ayurveda	Apollo Hospital	Bangalore	2500
2	2	Dr. Abdul	Homeopathy	Fortis Hospital	Bangalore	2000
3	3	Dr. Shwetha	Homeopathy	KMC Hospital	Manipal	1000
4	4	Dr. Murphy	Dermatology	KMC Hospital	Manipal	1500
5	5	Dr. Farhana	Physician	Gleneagles Hospital	Bangalore	1700
6	6	Dr. Maryam	Physician	Gleneagles Hospital	Bangalore	1500

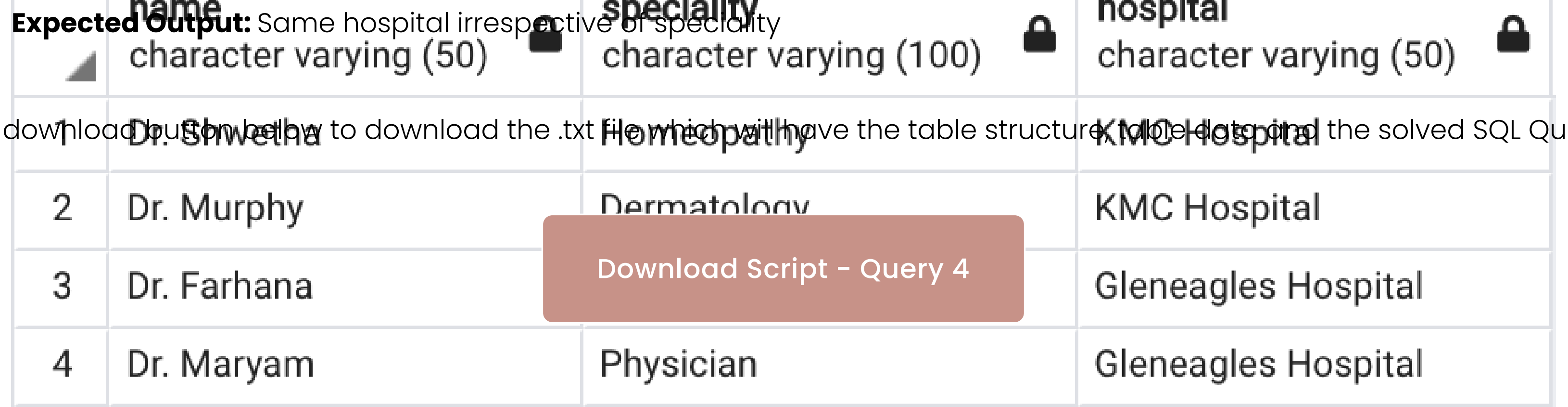
DOCTORS



	name character varying (50)	speciality character varying (100)	hospital character varying (50)
1	Dr. Shwetha	Homeopathy	KMC Hospital
2	Dr. Murphy	Dermatology	KMC Hospital

Expected Output: Same hospital different speciality





	name character varying (50)	speciality character varying (100)	hospital character varying (50)
1	Dr. Shwetha	Homeopathy	KMC Hospital
2	Dr. Murphy	Dermatology	KMC Hospital
3	Dr. Farhana	[illegible]	Gleneagles Hospital
4	Dr. Maryam	Physician	Gleneagles Hospital

Expected Output: Same hospital irrespective of speciality

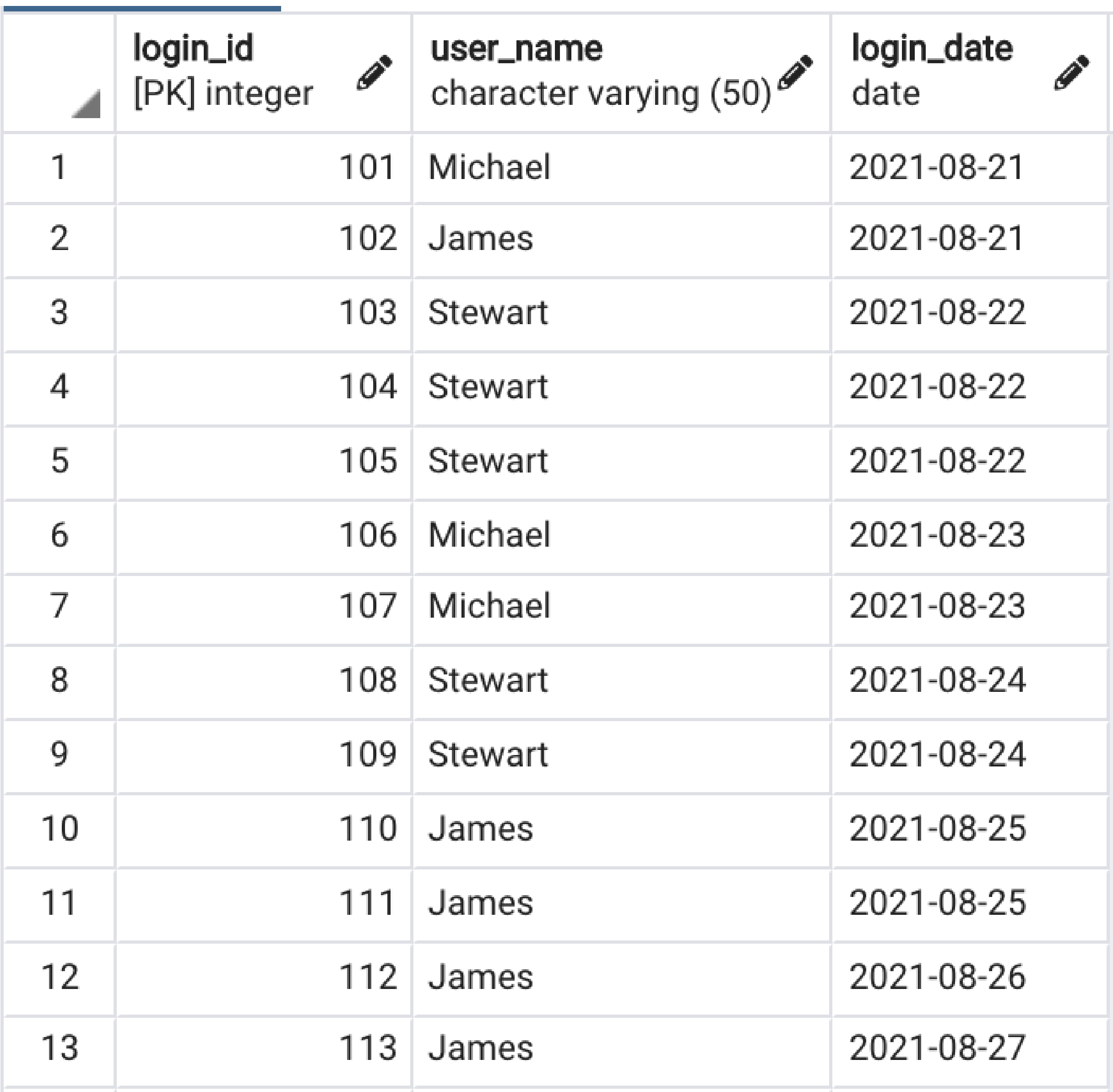
Click on the download button below to download the .txt file which will have the table structure, table data and the solved SQL Query.

Download Script - Query 4

5. From the login_details table, fetch the users who logged in consecutively 3 or more times.

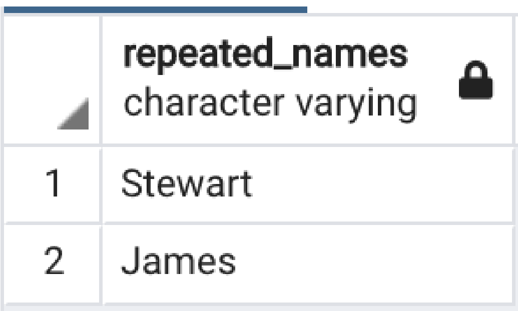
Table Name: LOGIN_DETAILS

Approach: We need to fetch users who have appeared 3 or more times consecutively in login details table. There is a window function which can be used to fetch data from the following record. Use that window function to compare the user name in current row with user name in the next row and in the row following the next row. If it matches then fetch those records.



	login_id [PK] integer	user_name character varying (50)	login_date date
1	101	Michael	2021-08-21
2	102	James	2021-08-21
3	103	Stewart	2021-08-22
4	104	Stewart	2021-08-22
5	105	Stewart	2021-08-22
6	106	Michael	2021-08-23
7	107	Michael	2021-08-23
8	108	Stewart	2021-08-24
9	109	Stewart	2021-08-24
10	110	James	2021-08-25
11	111	James	2021-08-25
12	112	James	2021-08-26
13	113	James	2021-08-27

LOGIN_DETAILS



	repeated_names character varying
1	Stewart
2	James

Expected Output

Click on the download button below to download the .txt file which will have the table structure, table data and the solved SQL Query.

Download Script - Query 5

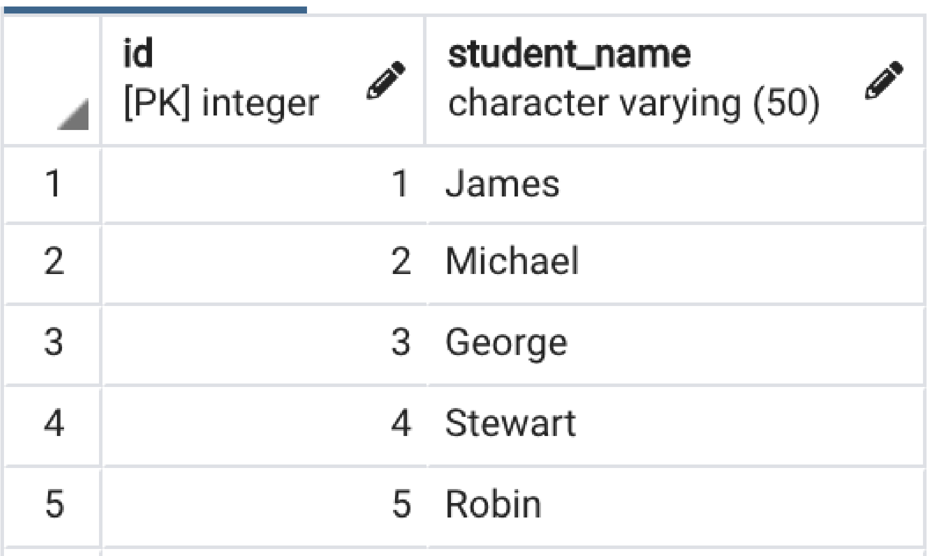
6. From the students table, write a SQL query to interchange the adjacent student names.

Note: If there are no adjacent student then the student name should stay the same.

Table Name: STUDENTS

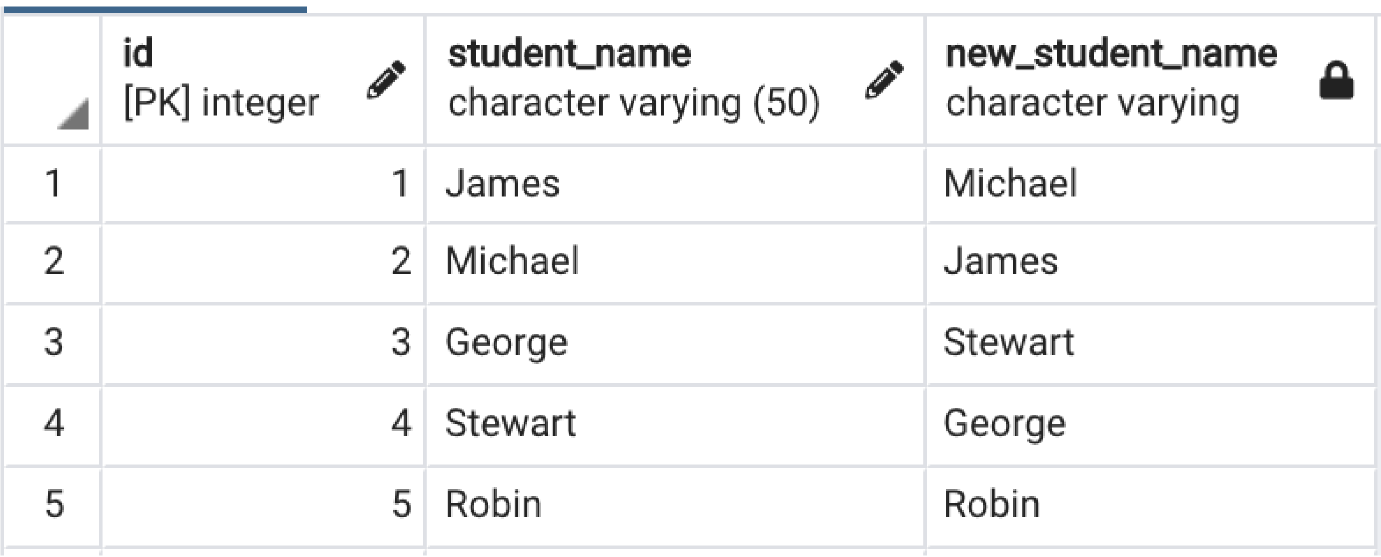
Approach: Assuming id will be a sequential number always. If id is an odd number then fetch the student name from the following record. If id is an even number then fetch the student name from the preceding record. Try to figure out the window function which can be used to fetch the preceding the following record data.

If the last record is an odd number then it wont have any adjacent even number hence figure out a way to not interchange the last record data.



	id [PK] integer	student_name character varying (50)
1	1	James
2	2	Michael
3	3	George
4	4	Stewart
5	5	Robin

STUDENTS



	id [PK] integer	student_name character varying (50)	new_student_name character varying
1	1	James	Michael
2	2	Michael	James
3	3	George	Stewart
4	4	Stewart	George
5	5	Robin	Robin

Expected Output

Click on the download button below to download the .txt file which will have the table structure, table data and the solved SQL Query.

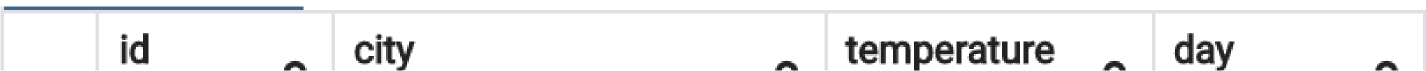
Download Script - Query 6

7. From the weather table, fetch all the records when London had extremely cold temperature for 3 consecutive days or more.

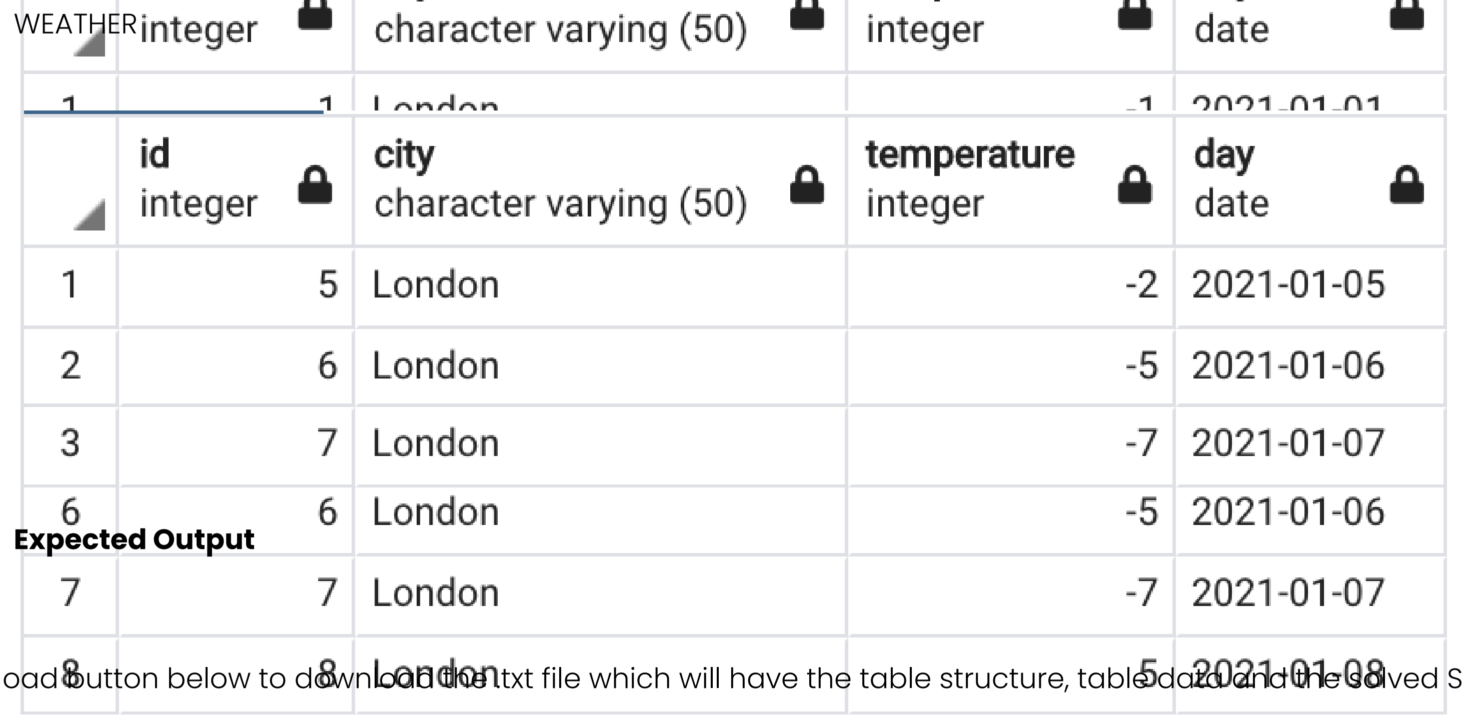
Note: Weather is considered to be extremely cold when its temperature is less than zero.

Table Name: WEATHER

Approach: First using a sub query identify all the records where the temperature was very cold and then use a main query to fetch only the records returned as very cold from the sub query. You will not only need to compare the records following the current row but also need to compare the records preceding the current row. And may also need to compare rows preceding and following the current row. Identify a window function which can do this comparison pretty easily.



	id	city	temperature	day



	integer	character varying (50)	integer	date
1	1	London	-1	2021-01-01
6	6	London	-5	2021-01-06
7	7	London	-7	2021-01-07
8	8	London	-5	2021-01-08

WEATHER

Expected Output

	id integer	city character varying (50)	temperature integer	day date
1	5	London	-2	2021-01-05
2	6	London	-5	2021-01-06
3	7	London	-7	2021-01-07

Click on the download button below to download the txt file which will have the table structure, table data and the solved SQL Query.

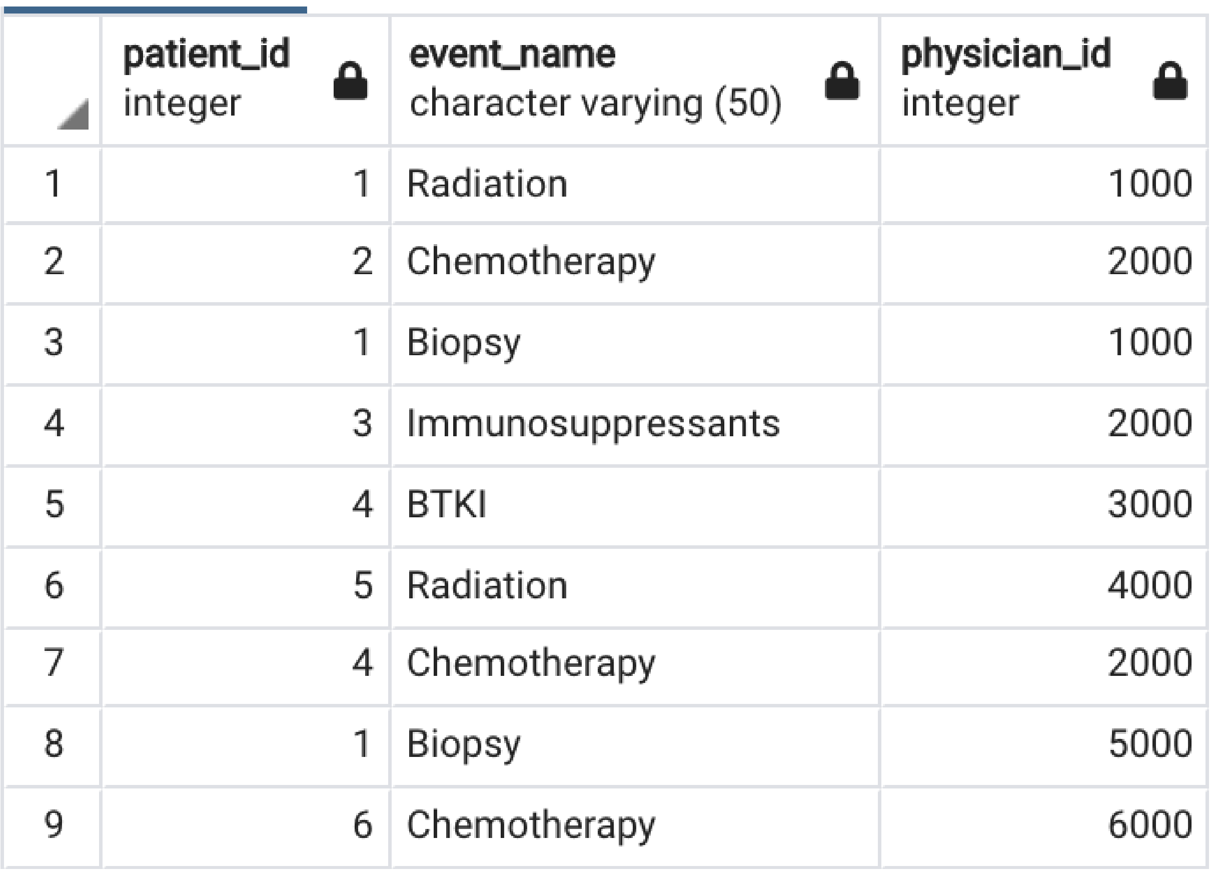
Download Script - Query 7

8. From the following 3 tables (event_category, physician_speciality, patient_treatment), write a SQL query to get the histogram of specialties of the unique physicians who have done the procedures but never did prescribe anything.

Table Name: EVENT_CATEGORY, PHYSICIAN_SPECIALITY, PATIENT_TREATMENT

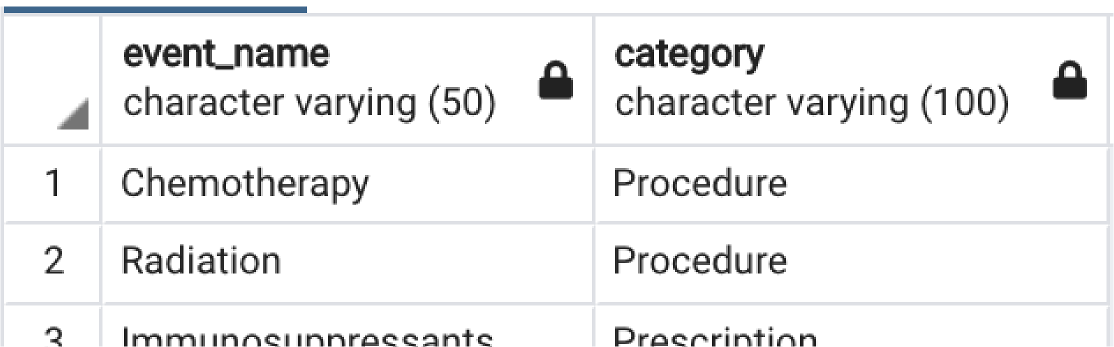
Approach: Using the patient treatment and event category table, identify all the physicians who have done "Prescription". Have this recorded in a sub query.

Then in the main query join the patient treatment, event category and physician speciality table to identify all the physician who have done "Procedure". From these physicians, remove those physicians you got from sub query to return the physicians who have done Procedure but never did Prescription.

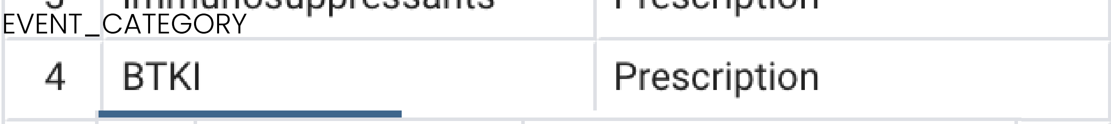


	patient_id integer	event_name character varying (50)	physician_id integer
1	1	Radiation	1000
2	2	Chemotherapy	2000
3	1	Biopsy	1000
4	3	Immunosuppressants	2000
5	4	BTKI	3000
6	5	Radiation	4000
7	4	Chemotherapy	2000
8	1	Biopsy	5000
9	6	Chemotherapy	6000

PATIENT_TREATMENT

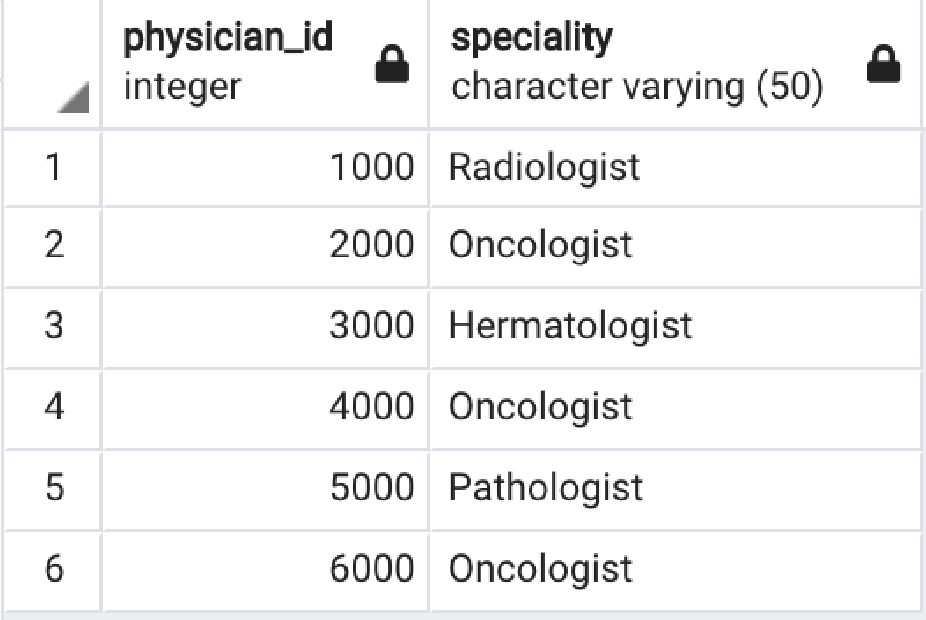


	event_name character varying (50)	category character varying (100)
1	Chemotherapy	Procedure
2	Radiation	Procedure
3	Immunosuppressants	Prescription



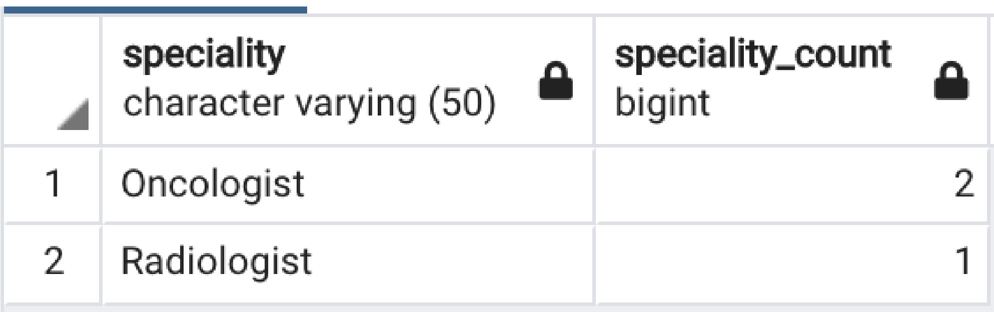
3	Immunosuppressants	Prescription
4	BTKI	Prescription
5		

EVENT_CATEGORY



	physician_id integer	speciality character varying (50)
1	1000	Radiologist
2	2000	Oncologist
3	3000	Hermatologist
4	4000	Oncologist
5	5000	Pathologist
6	6000	Oncologist

PHYSICIAN_SPECIALITY



	speciality character varying (50)	speciality_count bigint
1	Oncologist	2
2	Radiologist	1

Expected Output

Click on the download button below to download the .txt file which will have the table structure, table data and the solved SQL Query.

Download Script - Query 8

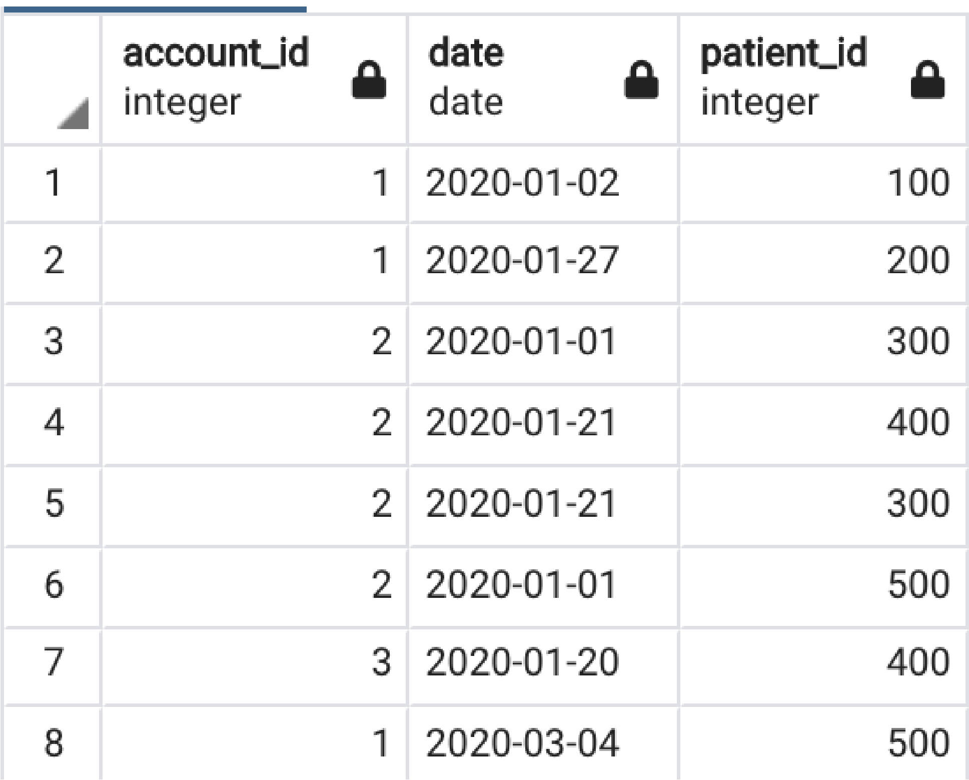
9. Find the top 2 accounts with the maximum number of unique patients on a monthly basis.

Note: Prefer the account id with the least value in case of same number of unique patients

Table Name: PATIENT_LOGS

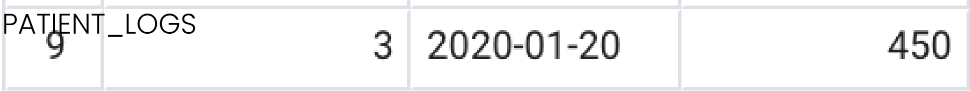
Approach: First convert the date to month format since we need the output specific to each month. Then group together all data based on each month and account id so you get the total no of patients belonging to each account per month basis.

Then rank this data as per no of patients in descending order and account id in ascending order so in case there are same no of patients present under multiple account if then the ranking will prefer the account if with lower value. Finally, choose upto 2 records only per month to arrive at the final output.

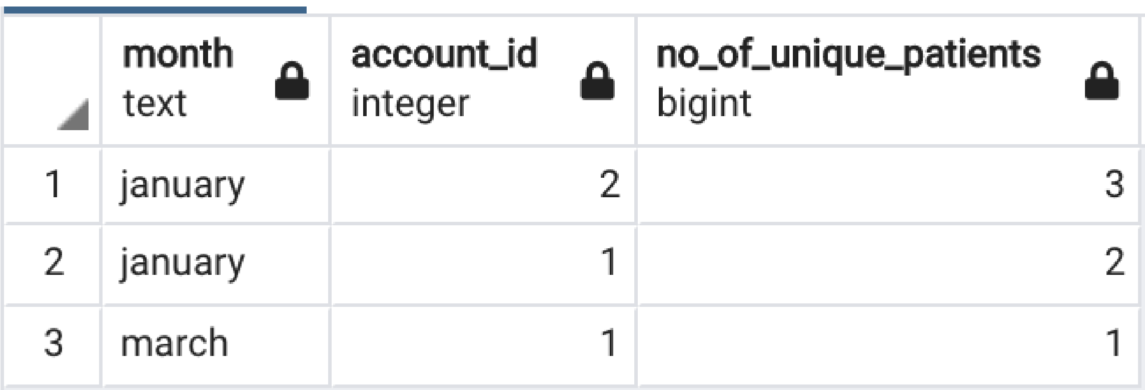


	account_id integer	date date	patient_id integer
1	1	2020-01-02	100
2	1	2020-01-27	200
3	2	2020-01-01	300
4	2	2020-01-21	400
5	2	2020-01-21	300
6	2	2020-01-01	500
7	3	2020-01-20	400
8	1	2020-03-04	500

PATIENT_LOGS



9	3	2020-01-20	450



	month text	account_id integer	no_of_unique_patients bigint
1	january	2	3
2	january	1	2
3	march	1	1

Expected Output

Click on the download button below to download the .txt file which will have the table structure, table data and the solved SQL Query.

Download Script - Query 9

10. SQL Query to fetch "N" consecutive records from a table based on a certain condition

Note: Write separate queries to satisfy following scenarios:

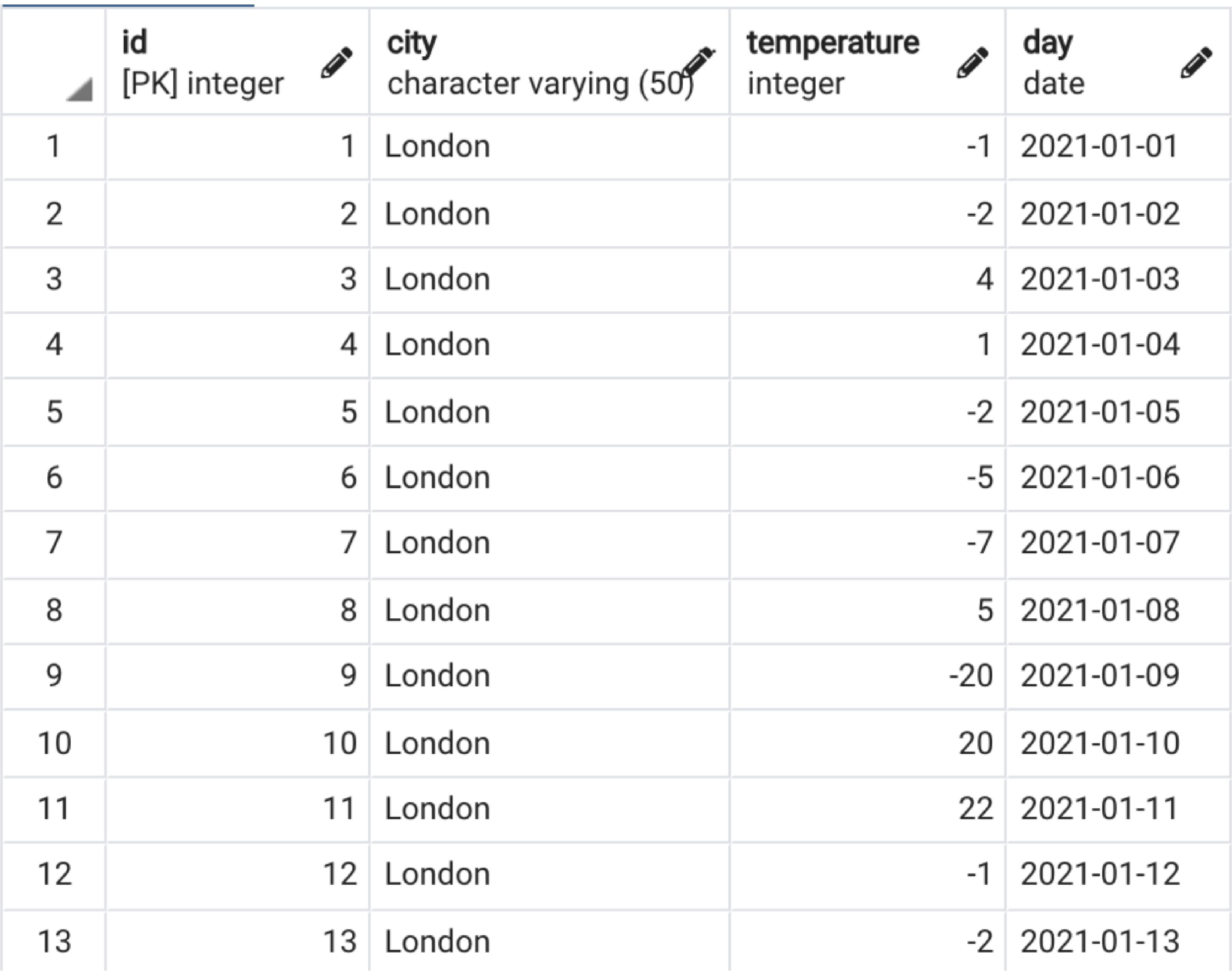
10a. when the table has a primary key

10b. When table does not have a primary key

10c. Query logic based on data field

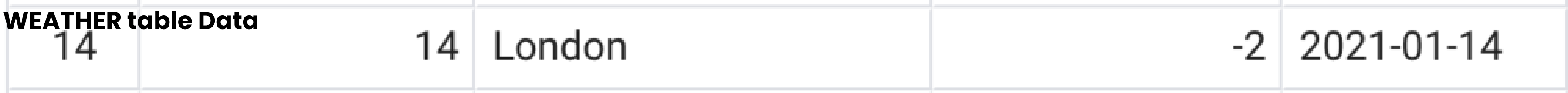
10a. when the table has a primary key

Table Name: WEATHER

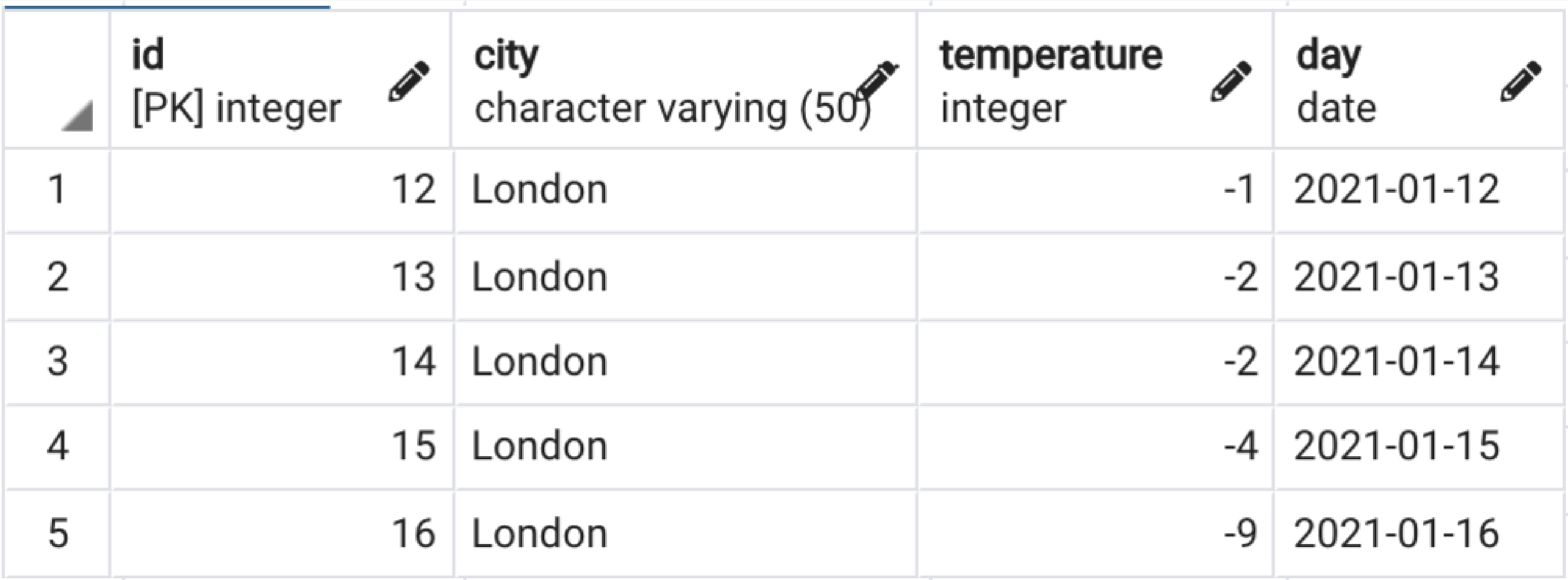


	id [PK] integer	city character varying (50)	temperature integer	day date
1	1	London	-1	2021-01-01
2	2	London	-2	2021-01-02
3	3	London	4	2021-01-03
4	4	London	1	2021-01-04
5	5	London	-2	2021-01-05
6	6	London	-5	2021-01-06
7	7	London	-7	2021-01-07
8	8	London	5	2021-01-08
9	9	London	-20	2021-01-09
10	10	London	20	2021-01-10
11	11	London	22	2021-01-11
12	12	London	-1	2021-01-12
13	13	London	-2	2021-01-13

WEATHER table Data



14	14	London	-2	2021-01-14

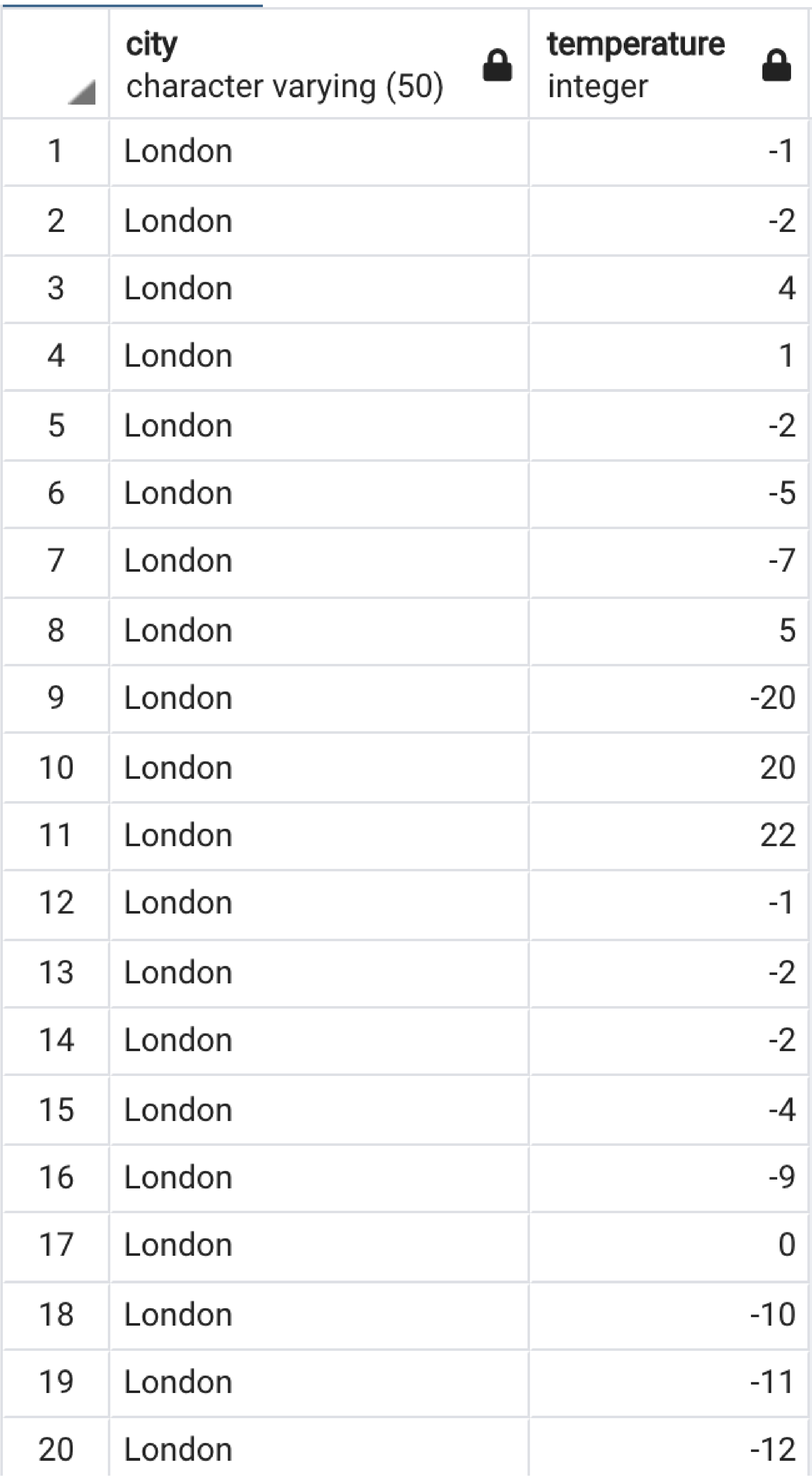


	id [PK] integer	city character varying (50)	temperature integer	day date
1	12	London	-1	2021-01-12
2	13	London	-2	2021-01-13
3	14	London	-2	2021-01-14
4	15	London	-4	2021-01-15
5	16	London	-9	2021-01-16

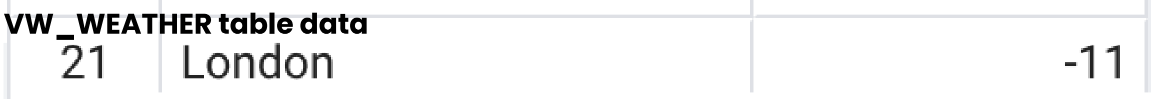
10a: Expected Output

10b. When table does not have a primary key

Table Name: VW_WEATHER

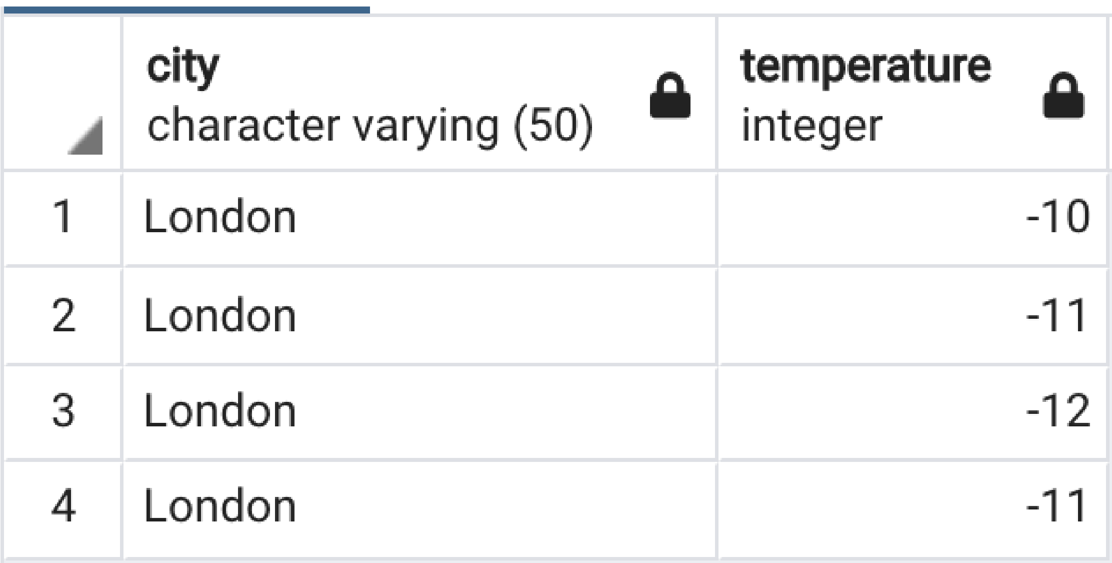


	city character varying (50)	temperature integer
1	London	-1
2	London	-2
3	London	4
4	London	1
5	London	-2
6	London	-5
7	London	-7
8	London	5
9	London	-20
10	London	20
11	London	22
12	London	-1
13	London	-2
14	London	-2
15	London	-4
16	London	-9
17	London	0
18	London	-10
19	London	-11
20	London	-12



21	London	-11

VW_WEATHER table data

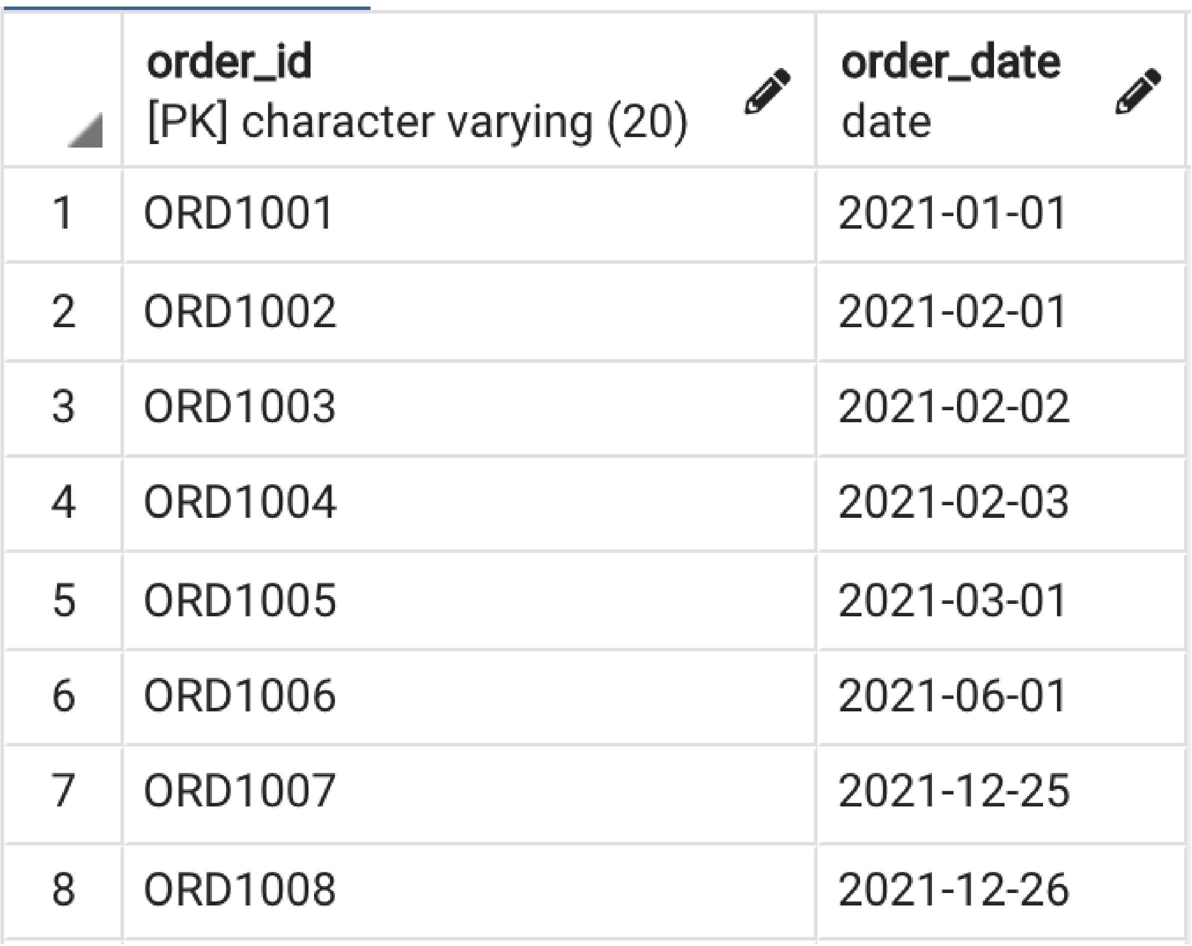


	city character varying (50)	temperature integer
1	London	-10
2	London	-11
3	London	-12
4	London	-11

10b: Expected Output

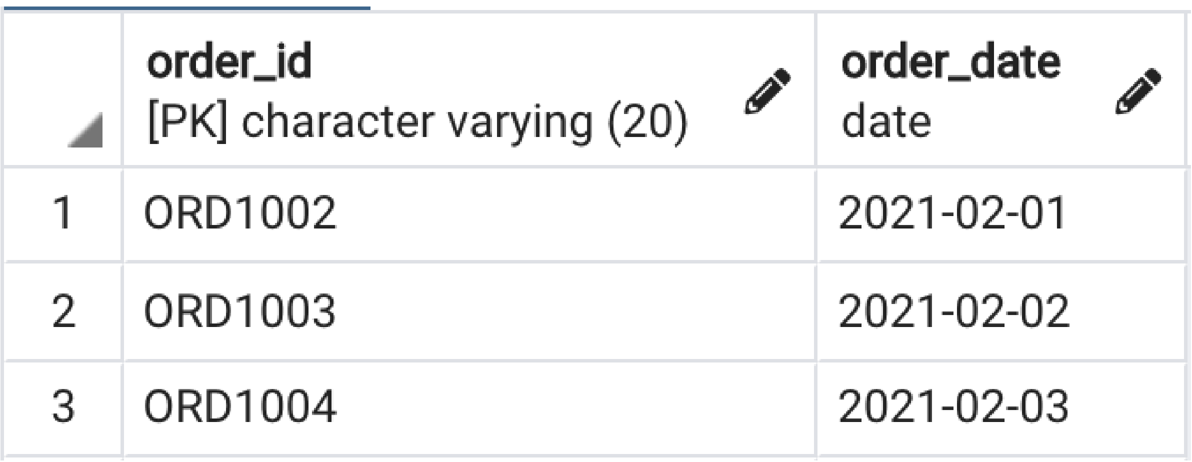
10c. Query logic based on data field

Table Name: ORDERS



	order_id [PK] character varying (20)	order_date date
1	ORD1001	2021-01-01
2	ORD1002	2021-02-01
3	ORD1003	2021-02-02
4	ORD1004	2021-02-03
5	ORD1005	2021-03-01
6	ORD1006	2021-06-01
7	ORD1007	2021-12-25
8	ORD1008	2021-12-26

ORDERS table data



	order_id [PK] character varying (20)	order_date date
1	ORD1002	2021-02-01
2	ORD1003	2021-02-02
3	ORD1004	2021-02-03

10c: Expected Output

Download Scripts

how to write complex sql queries • learn how to write sql queries • practice complex sql queries • Learn how to write SQL Queries(Practice Complex SQL Queries) • Learn how to write SQL Queries (Practice Complex SQL Queries) • sql queries • practice sql queries • how to write sql queries • sql query • sql queries using window functions • sql queries using window function • window function sql queries • sql query using self join • complex sql queries • sql • query • writing sql query • practicing sql query • sql query writing • practice sql • learn sql • learn sql query

Comments (104)

Newest First

Preview

Post Comment…

coli A day ago · 0 Likes

Excellent - thank you !

A question - is it possible to modify below's code to only add new names (to avoid duplicates) ?

```
INSERT INTO myexisitingtable (mycolumnname)
SELECT MSysObjects.name
FROM MSysObjects
WHERE MSysObjects.type In (1,4,6)
and MSysObjects.name not like '~*'
and MSysObjects.name not like 'MSys*'
order by MSysObjects.name;
```

Anandh Kumar S 3 weeks ago · 0 Likes

where I can get this dataset ?

LUIS 3 weeks ago · 0 Likes

EXCELENTE

Anandh Kumar S 3 weeks ago · 0 Likes

could you please help me to where i can get the dataset

Jaime 2 months ago · 0 Likes

Thank you so much for these lessons and guidance, it worth every second of it and going throughout these exercises enlightened me a lot. Really appreciate it!!💯👌🚀

Hanisha Dua 2 months ago · 0 Likes

Hi Thoufiq,

I created below solution for Question 3. Would it be possible for you to check and share if this is a good one to go for. I got the table required from this. Post that I tried grouping the query by group by dept_name i.e. adding group by in both the queries of Union and then this could not be executed.

```
select *
from (select * ,
rank() over(partition by DEPT_NAME order by salary) as rn from employee) x
where x.rn = 1
union
select *
from (select * ,
rank() over(partition by DEPT_NAME order by salary desc) as rn from employee) x
where x.rn = 1
```

vc 2 months ago · 0 Likes

bcv

shama 3 months ago · 0 Likes

Thank you Sir
so useful queries

TAN 4 months ago · 0 Likes

Question 8 solutions returns no result.

Rutu 4 months ago · 0 Likes

hii i am getting this error while executing the first question
"The ORDER BY clause is invalid in views, inline functions, derived tables, subqueries, and common table expressions, unless TOP, OFFSET or FOR XML is also specified."
my query was: select * from

```
(
select *,ROW_NUMBER() over(partition BY name ORDER BY user_id) as rn
from student order by user_id) x
where x.rn > 1;
```

Please help

Guest 2 months ago · 0 Likes

Try this:

```
select * from
(
select *,ROW_NUMBER() over(partition BY user_name ORDER BY user_id) as rn
from users) x
where x.rn > 1;
```

Kit 4 months ago · 0 Likes

I tried Q.8: Please send feed back if other work around can be done thanks

```
select PS.speciality, count(speciality) Occurrence
from event_category EC
join patient_treatment PT on PT.event_name=EC.event_name
join physician_speciality PS on PS.physician_id=PT.physician_id
where ec.category='procedure' and pt.physician_id !='2000'
group by speciality;
```

deepika 5 months ago · 0 Likes

your videos are very helpful. You teach every concept so clearly.

Zubair 5 months ago · 0 Likes

Shukran my brother, ive just started learning sql now at age 39. Im hoping i can master it inshaAllah.
Keep us in your dua.
Zubair from cape town
+27716084294

sumit 5 months ago · 0 Likes

query 8 -- Please redefine what sql query to write in an easy language?

Pri 5 months ago · 0 Likes

Love your blog and YouTube videos!

Question for you - for query 5 when I paste the script into (DB Browser), the date and month does not show from row 3 onwards (shows only the year). Do you know why this is the case?

ken 5 months ago · 0 Likes

Q6

please let me know if I got it right

```
select *,
case when id = 1 then lead(student_name) over()
when id = 2 then lag(student_name) over()
when id = 3 then lead(student_name) over()
when id = 4 then lag(student_name) over()
else student_name
end as new
from test6;
```

kiran 3 months ago · 0 Likes

what if you have more than 100+ rows in your table?

Dennis A month ago · 0 Likes

This will not be easy if you have more than 1000+ rows to work with

sagar 5 months ago · 0 Likes

From where to get database?

wasim saifi 5 months ago · 0 Likes

DOWNLOAD .txt file and create tabel and insert value
you can copy it from file download
and just copy and paste

josh 5 months ago · 0 Likes

Why i cant get table in workbench my sql
Server option> data import> import completed with 1 error why?

sandeep 5 months ago · 0 Likes

```
Query 3--
Select * from (
Select max(salary) over(partition by DEPT_NAME) Max_Salary,
min(salary) over(partition by DEPT_NAME) min_Salary,
* from new_employee ) a
where ((a.SALARY=a.Max_Salary) or (a.SALARY=a.min_Salary))
```

sandeep 5 months ago · 0 Likes

Query 1

```
;with cte as
(Select ROW_NUMBER() over(partition by user_name order by user_id) rn,* from workers)
Select * from CTE where rn=2
```

```
;with cte as
(Select dense_rank() over(partition by user_name order by user_id) rn,* from workers)
Select * from CTE where rn=2
```

```
Select * from (
Select USER_NAME,count(*) cnt from workers
group by USER_NAME having count(*)>1
) a , workers w
where a.user_name=w.user_name
```

sandeep 5 months ago · 0 Likes

Query 2

```
select * from new_employee
order by emp_ID desc
OFFSET 1 ROWS
FETCH NEXT 1 ROWS ONLY;
```

```
with cte as (
select ROW_NUMBER() over(order by emp_id desc) rn
,* from new_employee
)
select * from cte where rn=2
```

```
Select top 1 * from (
select top 2 * from new_employee order by emp_ID desc) a
order by emp_ID
```

tejaswini 6 months ago · 0 Likes

question 6:

```
select *,isnull(case when id%2=0 then
lag(student_name) over(order by id)
when id%2=1 then
lead(student_name) over(order by id)
else
student_name
end,student_name) as new_student
from students
```

Terry H 7 months ago · 0 Likes

Add just 1 more record and both proposed solution fall apart. Only works with 2 doctors per hospital.

Add just #7 or all 3 values to see a better range of failures

```
insert into doctors values
(7, 'Dr. Bell', 'Urology', 'KMC Hospital', 'Lonetree', 5555)
(8, 'Dr. Gilmer', 'Internal Medicine', 'KMC Hospital', 'Aurora', 3333)
(9, 'Dr. Lumley', 'Internal Medicine', 'Gleneagles Hospital', 'Lonetree', 2222)
```

Pikachu 7 months ago · 0 Likes

1. From the login_details table, fetch the users who logged in consecutively 3 or more times.

ans:

```
select count(login_id) as count,user_name
from login_details
group by user_name
having count > 3 ;
```

praksh 7 months ago · 0 Likes

Hi sir,

Can you explain how do we identify and delete duplicate records in the table based on combination of 2 fields.

Sanjeeva 7 months ago · 0 Likes

Hi Sir,
Q1- Solution 1 is unable to understand, kindly explore more.

```
-- Solution 1:

-- Replace ctid with rowid for Oracle, MySQL and Microsoft SQLServer
select *
from users u
where u.ctid not in (
select min(ctid) as ctid
from users
group by user_name
order by ctid);
```

Santanu 8 months ago · 0 Likes

Q5. 3 consecutive day login

```
select distinct x.user_name from(
select *,
case when(user_name=lag(user_name) over() and user_name=lag(user_name, 2) over()) then 1 end as flg
from login_details) x where x.flg=1;
```

Santanu 8 months ago · 0 Likes

```
select distinct x.user_name from(
select *,
case when(user_name=lag(user_name) over() and user_name=lag(user_name, 2) over()) then 1 end as flg
from login_details) x where x.flg=1;
```

Kannadhasan 8 months ago · 0 Likes

For Question NO : 3 ,

alternate query :

```
select * from
(
select *,
max(salary) over (partition by dept_name) as max_salary ,
min(salary) over (partition by dept_name) as min_salary from employee
) x
where salary = x.max_salary or salary = x.min_salary;
```

Deepak Ahirwar 9 months ago · 0 Likes

Hello Sir,

I am trying 2nd Practical of finding 2nd highest salary instead of 2nd last record . Where the 2nd Last Salary as per data is = 10000.

Please could you help me what it must be,

I came across with the Query but seems not to be enough and is hardcoded.

```
with CTE(emp_ID,emp_NAME,DEPT_NAME,SALARY,Ranks)
AS(select * ,
rank() over(order by salary) as Ranks
from employee)
--order by salary desc)
```

```
select * from CTE
where CTE.Ranks = 23
order by Ranks desc;
```

Deepak Ahirwar 9 months ago · 0 Likes

Hello Sir,

The Query for 2 is giving wrong results as '8000' . Where 2nd Last Salary as per data is = 10000.

The query seems to be wrong. Please help

```
select emp_id, emp_name, dept_name, salary
from (
select *,
row_number() over (order by emp_id desc) as rn
from employee e) x
where x.rn = 2;
```

ssw 7 months ago · 0 Likes

i agree with you, the problem or solution is wrong

Sai Venkatesh 10 months ago · 0 Likes

This is quite useful, i had replicated the queries/questions with some of the employee tables that i already had in My Sql

Shankar 10 months ago · 0 Likes

Q2

```
SELECT * FROM EMPLOYEE
ORDER BY EMP_ID DESC
LIMIT 2,1
```

abhishek 10 months ago · 0 Likes

Q4. I updated the table with few more data and the query doesn't work. can you please provide a general query which can work on any number of data in the table .
waiting for your response.

Lubna 11 months ago · 0 Likes

Thanks for the scenarios to SQL Queries practice.

Teja 11 months ago · 0 Likes

Thanks for your great effort and Time. All these questions are a lot to us.

pradnya rane 11 months ago · 0 Likes

Respected sir,

I have one query which I am not able to solve please guide me to solving this query.
Q: show records from the database ie how many rows are in databases.

Thank you.

Walfrido Mora 11 months ago · 0 Likes

Great explanation

Hitesh Prajapati 11 months ago · 0 Likes

Growth Accounting Models

Background

Growth accounting for a company, breaks down overall growth in some activity across specific customer segments such as revenue or engagement. A company's revenue can be bucketed into the following categories under current time period (and compared to prior time periods):

New: Gained from customers were first active in the present time period.

Churned: Lost when a customer who was active in the previous time period has no revenue in the present one.

Resurrected: Gained from customers who had churned at some point in the past (and thus generated no revenue in the previous time period) but resumed in the present.

Expansion: Gained from customers increasing revenue relative to the previous time period.

Contraction: Lost from customers decreasing (but not to zero, otherwise they would be churned) revenue relative to the previous time period.

Retained: Carried over by customers from the previous time period to the present one.

For example, a customer who spent $10 last month and $12 this month would have $2 in expansion revenue and $10 in retained revenue. However, if this customer instead spent $8 this month (while still spending $10 last month), then $8 would be counted as retained and $2 as contracted.

There are three important identities that express the definitions above:

All revenue in the present time period is equal to the sum of the gains in revenue (new, resurrected, and expansion) and retained revenue.

Revenue(t) = retained(t) + new(t) + resurrected(t) + expansion(t)

All revenue from the previous time period must either churn, contract, or be retained in the present time period.

Revenue(t-1) = retained(t) + churned(t) + contraction(t)

All change in revenue across a time period is equal to the sum of the gains (new, expansion, and resurrected) minus the sum of the losses (churned and contraction).

Revenue(t) – Revenue(t-1) = new(t) + expansion(t) + resurrected (t) – churned(t) – contraction(r).

Task:

Given the following transactions table which contains information about transactions & subscriptions, you will create a new dbt project that will have a growth accounting model that generates some key growth accounting metrics.

Input:

Postgresql Connection Details:

host: db-qa.cxquuupnwygi.us-west-2.rds.amazonaws.com
port: 5432
database: take_home
user: take_home_user

password: p@ssword01
schema: public

table=transactions
table transactions
(
period date,
customer_id varchar,
amount numeric,
subscription_start date,
subscription_end date,
first_transaction_date date,
);
Expected Output (dbt model):

table growth_accounting
(
-- REQUIRED
period date,
active numeric, -- count of active users
retained numeric, -- count of retained/returning users
new numeric, -- count of new users
-- BONUS/EXTRA CREDITS
resurrected numeric, -- count of resurrected/reactivated users
churned numeric -- count of churned users
);
Please provide me about above given problem... Thanks

Guest 11 months ago · 0 Likes

Alternative Solution Question 7:

WITH tb1 AS (SELECT *, CASE WHEN LEAD(day) OVER() = day + 1 AND LAG(day) OVER() = day - 1 THEN 'Yes'
ELSE 'No' END AS checker
FROM weather
WHERE temperature < 0),

```
tb2 AS (SELECT id, CASE WHEN LEAD(checker) OVER() = 'Yes' THEN 'Ok'
   WHEN LAG(checker) OVER() = 'Yes' THEN 'Ok' ELSE 'No' END AS x FROM tb1)
```

SELECT id, city, temperature, day FROM tb1
INNER JOIN tb2 USING(id)
WHERE checker = 'Yes' OR x = 'Ok';

dc 6 months ago · 0 Likes

Another way:
SELECT * FROM
(SELECT *, COUNT (CASE WHEN weather.temperature < 0 THEN 1 ELSE NULL END) OVER (ORDER BY weather.DAY ROWS BETWEEN 2 PRECEDING AND 2 FOLLOWING) AS is_flag FROM weather) AS x
WHERE x.is_flag >= 3 AND x.temperature < 0

Shubham A year ago · 0 Likes

Que 3.
-- Solution:

select * from (
select *,
max(salary) over (partition by dept_name) as max_salary,
min(salary) over (partition by dept_name) as min_salary
from employee) x
Where salary = max_salary or salary = min_salary

Tariq Mahmood A year ago · 0 Likes

Thanks

Rushitha A year ago · 0 Likes

Hi TFQ,

For Query 5 : From the login_details table, fetch the users who logged in consecutively 3 or more times.
Instead of lead function we can use Rank as well ryt. This also gives correct output.

Below is my Query and please let me know if this is the right way to use Rank function or not.

```
select distinct user_name from (select *,rank() over(partition by user_name order by login_date ) as rnk
from login_details ) where rnk >=3
;
```

Prahlad 11 months ago · 0 Likes

Hi Rushitha ,

your solution has two mistakes
1). lets say that the user does the login 3 times consecutively on a single date and those are the only records for that particular user. now your rank function will give 1 three time in 'rnk' column rather than giving you a three for any row for that particular user partition. so your query will miss that out.
2). when there are three logins for that user but are not consecutive and are made on three different dates and since they are not consecutive that should not come into your solution however it does because for that particular user partition you will get a rank of three.

however in case you have trouble with window function try with this logic:

```
select distinct user_name from login_details d where (select user_name from login_details
where login_id=d.login_id)=(select user_name from login_details where login_id=d.login_id-2)
```

in case of n consecutive records just replace the '2' with n-1 and this will work .let me know if you have trouble understanding the logic

Prajwal 6 months ago · 0 Likes

Hey prahlad!, Really liked your solution and the way you simulated lead and lag ..want to connect with you via Gmail / telegram .. please share

Neda A year ago · 0 Likes

Question 9 in T-SQL:

```
with monthly_count as
(
select account_id, datename (month,[date]) as [month] , count(distinct(patient_id)) as patient_count
from patient_logs
group by account_id,datename (month,[date])
),
ranking as
(
select * , dense_rank() over (partition by [month] order by patient_count desc,account_id) as rnk
from monthly_count
)
select [month] , account_id, patient_count from ranking where rnk <=2
```

Neda A year ago · 0 Likes

Hi,
for Question 7 we can do:

```
with groups as
(
select *,
DENSE_RANK() over(partition by city order by day) as rn,
dateadd(day,-DENSE_RANK() over(partition by city order by day) ,day) as date_group
from weather
where temperature < 0
),
cnt as(
select ,count() over(partition by date_group) as day_count
from groups
)
select id,city,temperature,day
from cnt where day_count >=3
```

BYREDDY S M A year ago · 0 Likes

Q1) Select duplicates

```
Select count(*), user_name
From users
Group by user_name
Having count(*) > 1 ;
```

Theisha Marie A year ago · 0 Likes

Hi, I see the scripts but where can I get the dataset?

Avinash A year ago · 0 Likes

Q9) Can we use this query instead ?
Solution

```
select account_id, count (distinct patient_id) , month(date) as month from patient_logs

group by month,account_id

order by count(distinct id) desc , account id asc.

limit 2;
```

Avinash A year ago · 0 Likes

In Q6 : The answered query in the query doesn't take into account for the last student when the total count of student is in odd number. The solution exchanged the position of last student as well.

Adarsh A year ago · 0 Likes

Hi,

I'm using oracle 11g & for the first problem statement, I used following query;

```
SELECT * FROM SAMPLEUSERS SU WHERE SU.ROWID NOT IN (SELECT MIN(ROWID) AS ROWID FROM SAMPLEUSERS GROUP BY USERNAME ORDER BY ROWID);
```

but I'm getting error " ORA-00923: FROM keyword not found where expected ". I'm a new leaner for DB & I'm not able to understand this error. please suggest.

Aya A year ago · 0 Likes

Good day,

First of all Thank you so much for the informative and beneficial content that you share here and on your YouTube channel.

I have a question regarding the following;

1- Write a SQL Query to fetch all the duplicate records in a table.
When I run the following query, I get an error stating that "The ORDER BY clause is invalid in views, inline functions, derived tables, subqueries, and common table expressions, unless TOP, OFFSET or FOR XML is also specified."

However, when I remove the [ORDER BY user_id], the query works fine. please advise.

```
SELECT user_id, user_name, email
FROM
(
SELECT *, ROW_NUMBER() OVER (PARTITION BY user_name ORDER BY user_id) AS RN
FROM users
ORDER BY user_id
) DUP
WHERE RN>1;
```

ben A year ago · 0 Likes

HEY
IN QUERY NUMBER 3 I WOULD LIKE TO KNOW
why did u use the join table?

I think it is enough to write like that :

```
SELECT *
FROM(
select *, max(SALARY) OVER(PARTITION BY DEPT_NAME) MAXI,
MIN(SALARY) OVER(PARTITION BY DEPT_NAME) MINI
from employee
) C
WHERE C.SALARY=C.MINI OR C.SALARY=C.MAXI
```

thanks for helping

Manikanta 11 months ago · 0 Likes

there will problem when same salary was repeated in other department

khyati A year ago · 0 Likes

Hi, thanks for the alternative solution, but why did you mention where clause at the end?

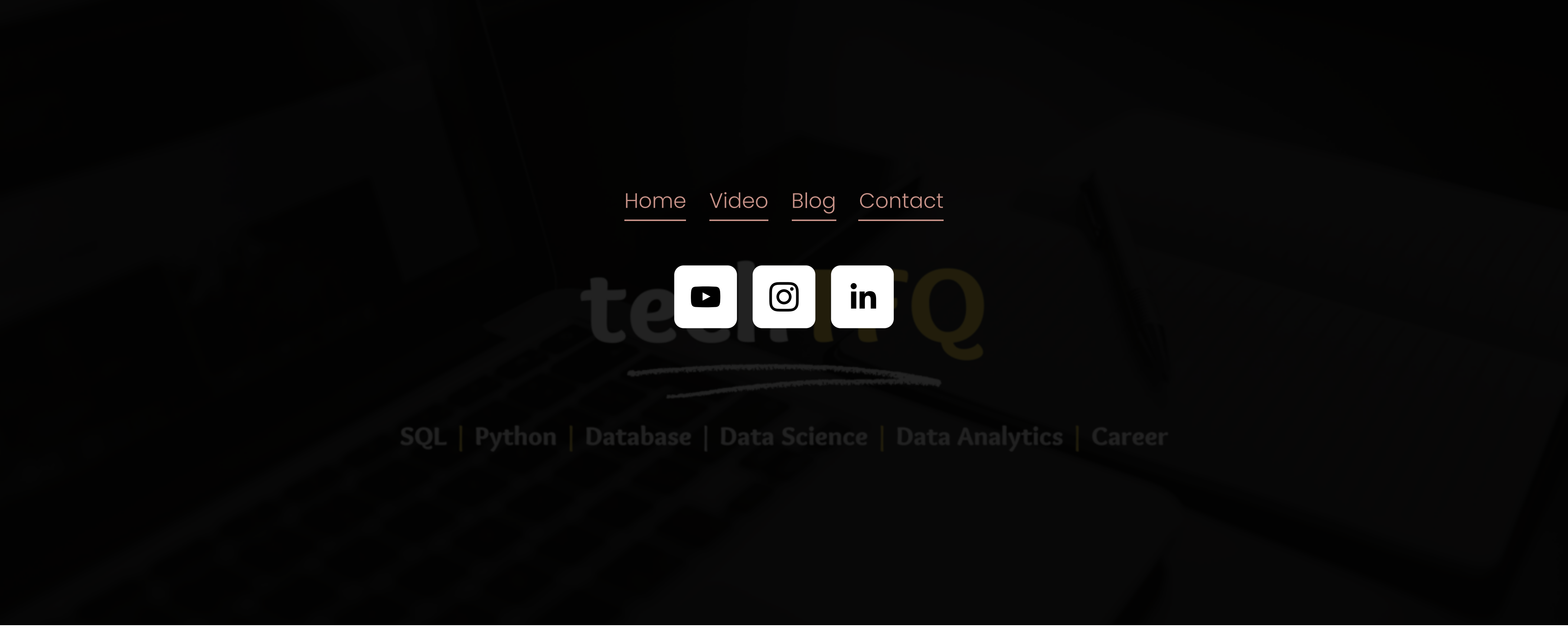
Satya A year ago · 0 Likes

You can use with clause as well if you want to, Using join is just one another approach.
1. The inner query is giving you min and max salaries including all the rows.
2. the actual salaries are being retrieved by the outer query and being compared with min or max and returns the matching records.

Load More Comments

‹ How to learn Python (Step by Step guide) | Python CONCEPTS to learn

Best Resume Format | Tips for writing an AWESOME Resume | ATS Resume Format ›



SQL | Python | Database | Data Science | Data Analytics | Career