Name: Shrish Tariq
Batch: DA Cohort 06 White
Report Title: Medical Appointment Records & Factors Influencing Patient Attendance
SQL Portfolio Project

Introduction

This report presents data-driven insights derived from real-world medical appointment data collected from the state of West Virginia, USA. The dataset includes demographic and clinical details of patients, enabling a comprehensive analysis of factors that influence patient attendance. The primary goal of this project is to utilize SQL-based analysis to identify key patterns that affect appointment no-shows. By uncovering these patterns, the project aims to support clinics in reducing missed appointments, enhancing communication with patients, and optimizing resource allocation.

Data Description:

I have been provided with a dataset of 10,000 appointments from the US state of West Virginia, along with clinic and patient details. The following are the details of the key variables:

PatientId: Unique numbers assigned to each patient for identification
AppointmentID: Unique numbers for each appointment
Gender: Gender of the patient
ScheduledDay: The time and day when the appointment was scheduled
AppointmentDay: Date when the appointment was to take place
Age: Age of the patient
Neighbourhood: Area where the patient lives
Scholarship: Indicates whether the patient is enrolled in a welfare program
Hypertension, Diabetes, Alcoholism, Handcap: Indicators of health conditions
SMS_received: Whether the patient received an SMS reminder
Showed_up: Whether the patient attended the appointment
Date.diff: Calculated difference in days between the scheduled and appointment date

Methodology and SQL Tasks

Data Cleaning

The first step in our data analysis process involves data cleaning and transformation. To achieve this, the CSV file of our Virginia patients' data was imported into the MySQL Workbench, and a database named “Virginiapatients” was created, containing a table named Virginia_patients. Then, the data was loaded into the result grid using the SELECT SQL command: “SELECT * FROM virginia_patients;”. This provided an overview of our dataset, ensured it was free from duplicates, and designated the “PatientId” as the primary key. After this, the following steps were

followed to clean, filter, and retrieve data along with the application of various SQL functions (such as aggregation, common table expressions, and windows functions).

Basic SQL & Data Retrieval

1. Retrieve all columns from the Virginia_patients table.

For this purpose, the following commands are used:

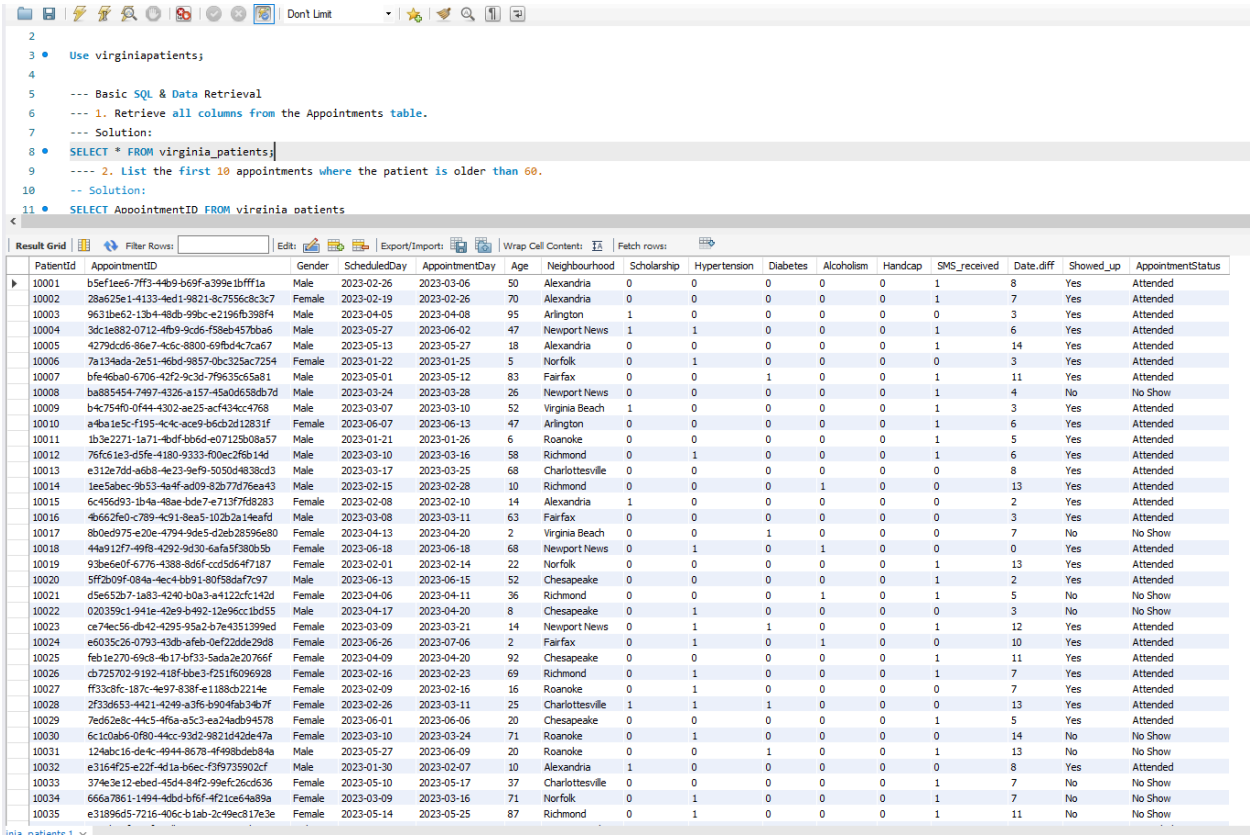
Solution:

CREATE database virginiapatients;

Use virginiapatients;

SELECT * FROM virginia_patients;

After running this statement, all columns from the Virginia_patients table are retrieved.



```
Use virginiapatients;

--- Basic SQL & Data Retrieval
--- 1. Retrieve all columns from the Appointments table.
--- Solution:
SELECT * FROM virginia_patients;
---- 2. List the first 10 appointments where the patient is older than 60.
-- Solution:
SELECT AppointmentID FROM virginia_patients
```

Result Grid

PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up	AppointmentStatus
10001	b5ef1ee6-7ff3-44b9-b69f-a399e1bfff1a	Male	2023-02-26	2023-03-06	50	Alexandria	0	0	0	0	0	1	8	Yes	Attended
10002	28a625e1-4133-4ed1-9821-8c7556c8c3c7	Female	2023-02-19	2023-02-26	70	Alexandria	0	0	0	0	0	1	7	Yes	Attended
10003	9631be62-13b4-48db-99bc-e2196fb398f4	Male	2023-04-05	2023-04-08	95	Arlington	1	0	0	0	0	0	3	Yes	Attended
10004	3dc1e882-0712-4fb9-9cd6-f58eb457bba6	Male	2023-05-27	2023-06-02	47	Newport News	1	1	0	0	0	1	6	Yes	Attended
10005	4279dcd6-86e7-4c6c-8800-69fbd4c7ca67	Male	2023-05-13	2023-05-27	18	Alexandria	0	0	0	0	0	1	14	Yes	Attended
10006	7a134ada-2e51-46bd-9857-0bc325ac7254	Female	2023-01-22	2023-01-25	5	Norfolk	0	1	0	0	0	0	3	Yes	Attended
10007	bfe46ba0-6706-42f2-9c3d-7f9635c65a81	Male	2023-05-01	2023-05-12	83	Fairfax	0	0	1	0	0	1	11	Yes	Attended
10008	ba885454-7497-4326-a157-45a0d658db7d	Male	2023-03-24	2023-03-28	26	Newport News	0	0	0	0	0	1	4	No	No Show
10009	b4c754f0-0f44-4302-ae25-acf434cc4768	Male	2023-03-07	2023-03-10	52	Virginia Beach	1	0	0	0	0	1	3	Yes	Attended
10010	a4ba1e5c-f195-4c4c-ace9-b6cb2d12831f	Female	2023-06-07	2023-06-13	47	Arlington	0	0	0	0	0	1	6	Yes	Attended
10011	1b3e2271-1a71-4bdf-bb6d-e07125b08a57	Male	2023-01-21	2023-01-26	6	Roanoke	0	0	0	0	0	1	5	Yes	Attended
10012	76fc61e3-d5fe-4180-9333-f00ec2f6b14d	Male	2023-03-10	2023-03-16	58	Richmond	0	1	0	0	0	1	6	Yes	Attended
10013	e312e7dd-a6b8-4e23-9ef9-5050d4838cd3	Male	2023-03-17	2023-03-25	68	Charlottesville	0	0	0	0	0	0	8	Yes	Attended
10014	1ee5abec-9b53-4a4f-ad09-82b77d76ea43	Male	2023-02-15	2023-02-28	10	Richmond	0	0	0	1	0	0	13	Yes	Attended
10015	6c456d93-1b4a-48ae-bde7-e713f7fd8283	Female	2023-02-08	2023-02-10	14	Alexandria	1	0	0	0	0	0	2	Yes	Attended
10016	4b662fe0-c789-4c91-8ea5-102b2a14eafd	Male	2023-03-08	2023-03-11	63	Fairfax	0	0	0	0	0	0	3	Yes	Attended
10017	8b0ed975-e20e-4794-9de5-d2eb28596e80	Female	2023-04-13	2023-04-20	2	Virginia Beach	0	0	1	0	0	0	7	No	No Show
10018	44a912f7-49f8-4292-9d30-6afa5f380b5b	Female	2023-06-18	2023-06-18	68	Newport News	0	1	0	1	0	0	0	Yes	Attended
10019	93be6e0f-6776-4388-8d6f-ccd5d64f7187	Female	2023-02-01	2023-02-14	22	Norfolk	0	0	0	0	0	1	13	Yes	Attended
10020	5ff2b09f-084a-4ec4-bb91-80f58daf7c97	Male	2023-06-13	2023-06-15	52	Chesapeake	0	0	0	0	0	1	2	Yes	Attended
10021	d5e652b7-1a83-4240-b0a3-a4122cfc142d	Female	2023-04-06	2023-04-11	36	Richmond	0	0	0	1	0	1	5	No	No Show
10022	020359c1-941e-42e9-b492-12e96cc1bd55	Male	2023-04-17	2023-04-20	8	Chesapeake	0	1	0	0	0	0	3	No	No Show
10023	ce74ec56-db42-4295-95a2-b7e4351399ed	Female	2023-03-09	2023-03-21	14	Newport News	0	1	1	0	0	1	12	Yes	Attended
10024	e6035c26-0793-43db-afeb-0ef22dde29d8	Female	2023-06-26	2023-07-06	2	Fairfax	0	1	0	1	0	0	10	Yes	Attended
10025	feb1e270-69c8-4b17-bf33-5ada2e20766f	Female	2023-04-09	2023-04-20	92	Chesapeake	0	0	0	0	0	1	11	Yes	Attended
10026	cb725702-9192-418f-bbe3-f251f6096928	Female	2023-02-16	2023-02-23	69	Richmond	0	1	0	0	0	1	7	Yes	Attended
10027	ff33c8fc-187c-4e97-838f-e1188cb2214e	Female	2023-02-09	2023-02-16	16	Roanoke	0	1	0	0	0	0	7	Yes	Attended
10028	2f33d653-4421-4249-a3f6-b904fab34b7f	Female	2023-02-26	2023-03-11	25	Charlottesville	1	1	1	0	0	0	13	Yes	Attended
10029	7ed62e8c-44c5-4f6a-a5c3-ea24adb94578	Female	2023-06-01	2023-06-06	20	Chesapeake	0	0	0	0	0	1	5	Yes	Attended
10030	6c1c0ab6-0f80-44cc-93d2-9821d42de47a	Female	2023-03-10	2023-03-24	71	Roanoke	0	1	0	0	0	0	14	No	No Show
10031	124abc16-de4c-4944-8678-4f498bdeb84a	Male	2023-05-27	2023-06-09	20	Roanoke	0	0	1	0	0	1	13	No	No Show
10032	e3164f25-e22f-4d1a-b6ec-f3f9735902cf	Male	2023-01-30	2023-02-07	10	Alexandria	1	0	0	0	0	0	8	Yes	Attended
10033	374e3e12-ebed-45d4-84f2-99efc26cd636	Female	2023-05-10	2023-05-17	37	Charlottesville	0	0	0	0	0	1	7	No	No Show
10034	666a7861-1494-4dbd-bf6f-4f21ce64a89a	Female	2023-03-09	2023-03-16	71	Norfolk	0	1	0	0	0	1	7	No	No Show
10035	e31896d5-7216-406c-b1ab-2c49ec817e3e	Female	2023-05-14	2023-05-25	87	Richmond	0	1	0	0	0	1	11	No	No Show

2. List the first 10 appointments where the patient is older than 60.

Solution:

SELECT AppointmentID FROM virginia_patients

WHERE Age > 60 limit 10;

```
11 •  SELECT AppointmentID FROM virginia_patients
12    WHERE Age > 60 limit 10;
```

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

AppointmentID
28a625e1-4133-4ed1-9821-8c7556c8c3c7
9631be62-13b4-48db-99bc-e2196fb398f4
bfe46ba0-6706-42f2-9c3d-7f9635c65a81
e312e7dd-a6b8-4e23-9ef9-5050d4838cd3
4b662fe0-c789-4c91-8ea5-102b2a14eafd
44a912f7-49f8-4292-9d30-6afa5f380b5b
feb1e270-69c8-4b17-bf33-5ada2e20766f
cb725702-9192-418f-bbe3-f251f6096928
6c1c0ab6-0f80-44cc-93d2-9821d42de47a
666a7861-1494-4dbd-bf6f-4f21ce64a89a

To retrieve the information of the first ten patients from the Virginia_patients table whose age is above sixty, the SELECT command with the appointmentID column and a limit clause was applied, and it gave the results of the appointments.

3. Show the unique neighborhoods from which patients came.

Solution:

SELECT Distinct Neighbourhood From virginia_patients;

```
14    -- Solution:
15 •  SELECT Distinct Neighbourhood From virginia_patients;
16 •  SELECT * FROM virginia_patients;
```

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

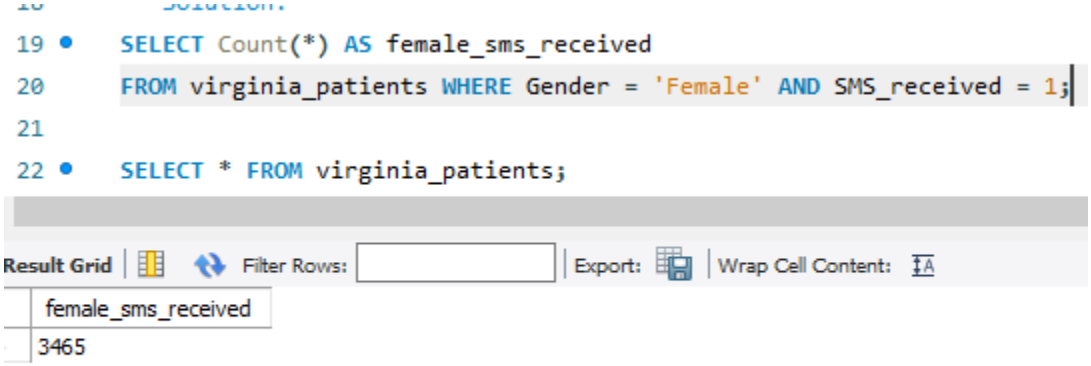
Neighbourhood
Alexandria
Arlington
Newport News
Norfolk
Fairfax
Virginia Beach
Roanoke
Richmond
Charlottesville
Chesapeake

By running the DISTINCT() function in MySQL, I was able to filter and check the data set for duplicate values, ensuring it contains no identical values. In addition, it showed me a distinct neighbourhood from the Virginia_patients table for my analysis.

4. Find all female patients who received an SMS reminder. Give a count of them

Solution:

SELECT Count(*) AS female_sms_received
FROM virginia_patients WHERE Gender = 'Female' AND SMS_received = 1;
SELECT * FROM virginia_patients;



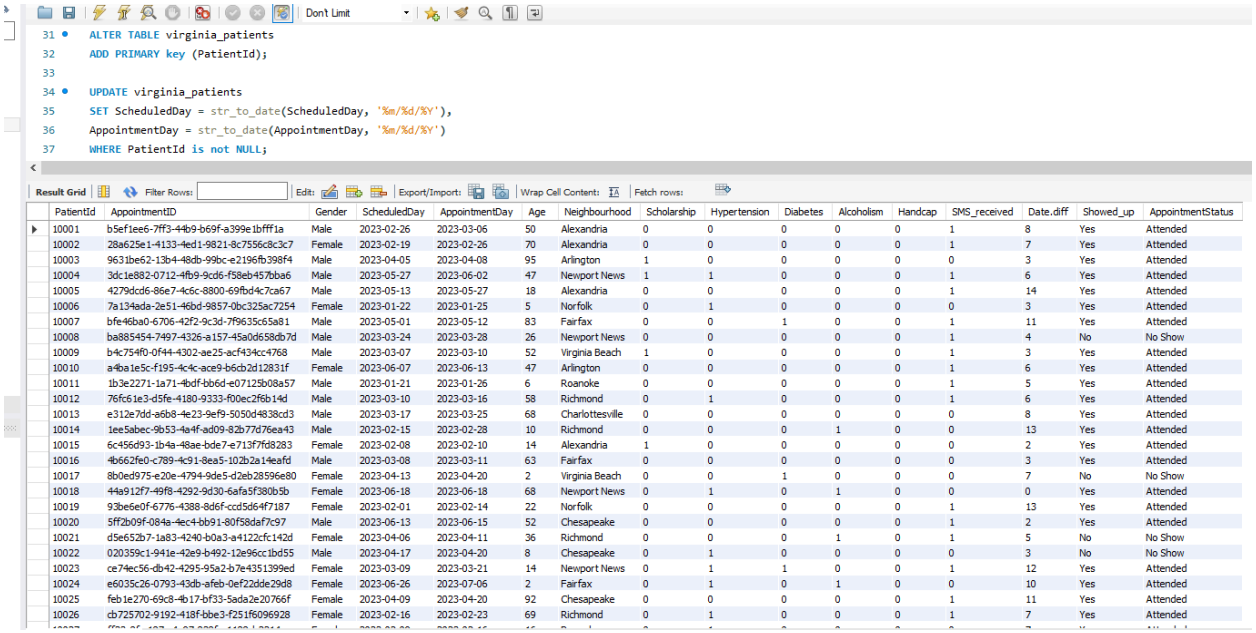
To find the number of female patients who received SMS reminders for their appointments, the COUNT() function was used. It returned a count of **3465** female patients who received an SMS. This query helps us derive insights into whether SMS reminders were fairly distributed across genders and how these reminders may have influenced **appointment attendance,** especially when analyzed alongside the no_show status.

Data Modification & Filtering

5. Update the dates in the ScheduledDay and AppointmentDay columns to the correct date format (hint: str_to_date() is a very helpful function to sort out dates)

Solution:

SELECT * FROM virginia_patients;
SET sql_safe_updates = 0;
ALTER TABLE virginia_patients
ADD PRIMARY key (PatientId);
UPDATE virginia_patients
SET ScheduledDay = str_to_date(ScheduledDay, '%m/%d/%Y'),
AppointmentDay = str_to_date(AppointmentDay, '%m/%d/%Y')
WHERE PatientId is not NULL;



```
ALTER TABLE virginia_patients
ADD PRIMARY key (PatientId);

UPDATE virginia_patients
SET ScheduledDay = str_to_date(ScheduledDay, '%m/%d/%Y'),
AppointmentDay = str_to_date(AppointmentDay, '%m/%d/%Y')
WHERE PatientId is not NULL;
```

Result Grid

PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up	AppointmentStatus
10001	b5ef1ee6-7ff3-44b9-b69f-a399e1bfff1a	Male	2023-02-26	2023-03-06	50	Alexandria	0	0	0	0	0	1	8	Yes	Attended
10002	28a625e1-4133-4ed1-9821-8c7556c8c3c7	Female	2023-02-19	2023-02-26	70	Alexandria	0	0	0	0	0	1	7	Yes	Attended
10003	9631be62-13b4-48db-99bc-e2196fb398f4	Male	2023-04-05	2023-04-08	95	Arlington	1	0	0	0	0	0	3	Yes	Attended
10004	3dc1e882-0712-4fb9-9cd6-f58eb457bba6	Male	2023-05-27	2023-06-02	47	Newport News	1	1	0	0	0	1	6	Yes	Attended
10005	4279dcd6-86e7-4c6c-8800-69fbd4c7ca67	Male	2023-05-13	2023-05-27	18	Alexandria	0	0	0	0	0	1	14	Yes	Attended
10006	7a134ada-2e51-46bd-9857-0bc325ac7254	Female	2023-01-22	2023-01-25	5	Norfolk	0	1	0	0	0	0	3	Yes	Attended
10007	bfe46ba0-6706-42f2-9c3d-7f9635c65a81	Male	2023-05-01	2023-05-12	83	Fairfax	0	0	1	0	0	1	11	Yes	Attended
10008	ba885454-7497-4326-a157-45a0d658db7d	Male	2023-03-24	2023-03-28	26	Newport News	0	0	0	0	0	1	4	No	No Show
10009	b4c754f0-0f44-4302-ae25-acf434cc4768	Male	2023-03-07	2023-03-10	52	Virginia Beach	1	0	0	0	0	1	3	Yes	Attended
10010	a4ba1e5c-f195-4c4c-ace9-b6cb2d12831f	Female	2023-06-07	2023-06-13	47	Arlington	0	0	0	0	0	1	6	Yes	Attended
10011	1b3e2271-1a71-4bdf-bb6d-e07125b08a57	Male	2023-01-21	2023-01-26	6	Roanoke	0	0	0	0	0	1	5	Yes	Attended
10012	76fc61e3-d5fe-4180-9333-f00ec2f6b14d	Male	2023-03-10	2023-03-16	58	Richmond	0	1	0	0	0	1	6	Yes	Attended
10013	e312e7dd-a6b8-4e23-9ef9-5050d4838cd3	Male	2023-03-17	2023-03-25	68	Charlottesville	0	0	0	0	0	0	8	Yes	Attended
10014	1ee5abec-9b53-4a4f-ad09-82b77d76ea43	Male	2023-02-15	2023-02-28	10	Richmond	0	0	0	1	0	0	13	Yes	Attended
10015	6c456d93-1b4a-48ae-bde7-e713f7fd8283	Female	2023-02-08	2023-02-10	14	Alexandria	1	0	0	0	0	0	2	Yes	Attended
10016	4b662fe0-c789-4c91-8ea5-102b2a14eafd	Male	2023-03-08	2023-03-11	63	Fairfax	0	0	0	0	0	0	3	Yes	Attended
10017	8b0ed975-e20e-4794-9de5-d2eb28596e80	Female	2023-04-13	2023-04-20	2	Virginia Beach	0	0	1	0	0	0	7	No	No Show
10018	44a912f7-49f8-4292-9d30-6afa5f380b5b	Female	2023-06-18	2023-06-18	68	Newport News	0	1	0	1	0	0	0	Yes	Attended
10019	93be6e0f-6776-4388-8d6f-ccd5d64f7187	Female	2023-02-01	2023-02-14	22	Norfolk	0	0	0	0	0	1	13	Yes	Attended
10020	5ff2b09f-084a-4ec4-bb91-80f58daf7c97	Male	2023-06-13	2023-06-15	52	Chesapeake	0	0	0	0	0	1	2	Yes	Attended
10021	d5e652b7-1a83-4240-b0a3-a4122cfc142d	Female	2023-04-06	2023-04-11	36	Richmond	0	0	0	1	0	1	5	No	No Show
10022	020359c1-941e-42e9-b492-12e96cc1bd55	Male	2023-04-17	2023-04-20	8	Chesapeake	0	1	0	0	0	0	3	No	No Show
10023	ce74ec56-db42-4295-95a2-b7e4351399ed	Female	2023-03-09	2023-03-21	14	Newport News	0	1	1	0	0	1	12	Yes	Attended
10024	e6035c26-0793-43db-afeb-0ef22dde29d8	Female	2023-06-26	2023-07-06	2	Fairfax	0	1	0	1	0	0	10	Yes	Attended
10025	feb1e270-69c8-4b17-bf33-5ada2e20766f	Female	2023-04-09	2023-04-20	92	Chesapeake	0	0	0	0	0	1	11	Yes	Attended
10026	cb725702-9192-418f-bbe3-f251f6096928	Female	2023-02-16	2023-02-23	69	Richmond	0	1	0	0	0	1	7	Yes	Attended

To update the date format of both the ScheduledDay and AppointmentDay columns, the str_to_date function was used. It provided us with a valid SQL date format by converting it from a string to a date. This date updation will help in date-based calculations and filter our rows more reliably.

6. Modify the datatypes of the ScheduledDay and AppointmentDay columns to DATE

Solution:

```
ALTER TABLE virginia_patients
modify column ScheduledDay DATE,
modify column AppointmentDay DATE;
SELECT * FROM virginia_patients;
```

Table: virginia_patients

Columns:

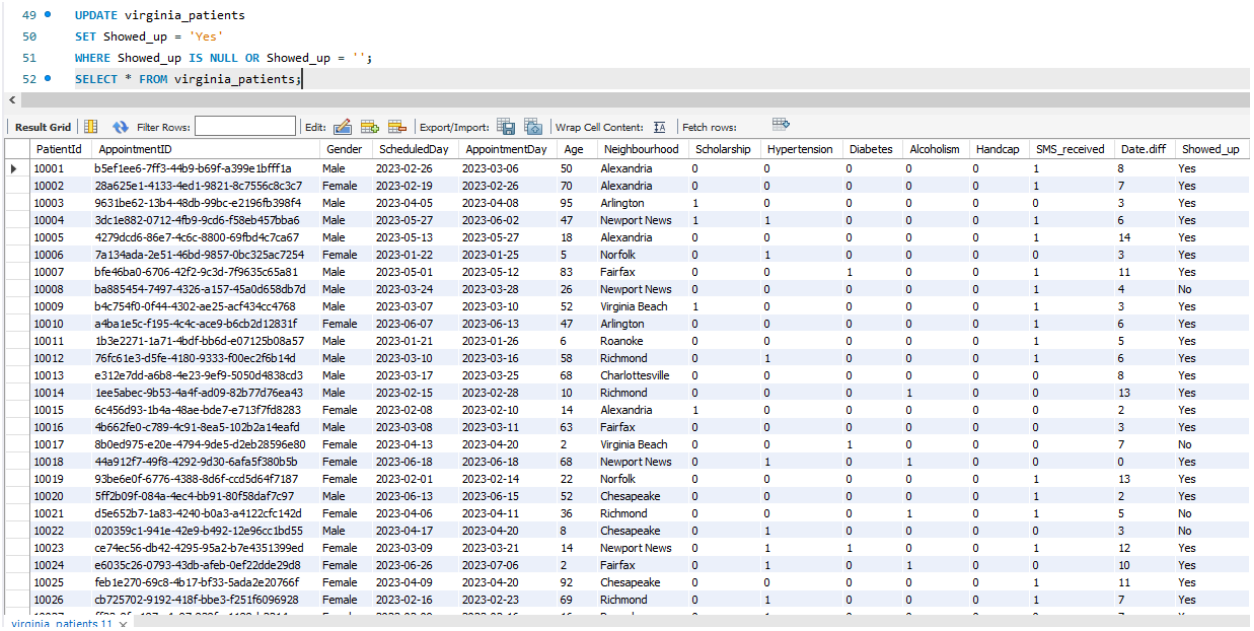
PatientId	int PK
AppointmentID	text
Gender	text
ScheduledDay	dat
AppointmentDay	dat
Age	int
Neighbourhood	text
Scholarship	int
Hypertension	int
Diabetes	int
Alcoholism	int
Handcap	int
SMS_received	int
Date.diff	int
Showed_up	text
AppointmentStatus	text

In this query, the datatypes of the ScheduledDay and appointmentDay columns were changed to a date datatype by using the alter table, modify column command.

7. Update the 'Showed_up' status to 'Yes' where it is null or empty

Solution:

```
UPDATE virginia_patients
SET Showed_up = 'Yes'
WHERE Showed_up IS NULL OR Showed_up = ' ';
SELECT * FROM virginia_patients;
```



PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up
10001	b5ef1ee6-7ff3-44b9-b69f-a399e1bfff1a	Male	2023-02-26	2023-03-06	50	Alexandria	0	0	0	0	0	1	8	Yes
10002	28a625e1-4133-4ed1-9821-8c7556c8c3c7	Female	2023-02-19	2023-02-26	70	Alexandria	0	0	0	0	0	1	7	Yes
10003	9631be62-13b4-48db-99bc-e2196fb398f4	Male	2023-04-05	2023-04-08	95	Arlington	1	0	0	0	0	0	3	Yes
10004	3dc1e882-0712-4fb9-9cd6-f58eb457bba6	Male	2023-05-27	2023-06-02	47	Newport News	1	1	0	0	0	1	6	Yes
10005	4279dcd6-86e7-4c6c-8800-69fbd4c7ca67	Male	2023-05-13	2023-05-27	18	Alexandria	0	0	0	0	0	1	14	Yes
10006	7a134ada-2e51-46bd-9857-0bc325ac7254	Female	2023-01-22	2023-01-25	5	Norfolk	0	1	0	0	0	0	3	Yes
10007	bfe46ba0-6706-42f2-9c3d-7f9635c65a81	Male	2023-05-01	2023-05-12	83	Fairfax	0	0	1	0	0	1	11	Yes
10008	ba885454-7497-4326-a157-45a0d658db7d	Male	2023-03-24	2023-03-28	26	Newport News	0	0	0	0	0	1	4	No
10009	b4c754f0-0f44-4302-ae25-acf434cc4768	Male	2023-03-07	2023-03-10	52	Virginia Beach	1	0	0	0	0	1	3	Yes
10010	a4ba1e5c-f195-4c4c-ace9-b6cb2d12831f	Female	2023-06-07	2023-06-13	47	Arlington	0	0	0	0	0	1	6	Yes
10011	1b3e2271-1a71-4bdf-bb6d-e07125b08a57	Male	2023-01-21	2023-01-26	6	Roanoke	0	0	0	0	0	1	5	Yes
10012	76fc61e3-d5fe-4180-9333-f00ec2f6b14d	Male	2023-03-10	2023-03-16	58	Richmond	0	1	0	0	0	1	6	Yes
10013	e312e7dd-a6b8-4e23-9ef9-5050d4838cd3	Male	2023-03-17	2023-03-25	68	Charlottesville	0	0	0	0	0	0	8	Yes
10014	1ee5abec-9b53-4a4f-ad09-82b77d76ea43	Male	2023-02-15	2023-02-28	10	Richmond	0	0	0	1	0	0	13	Yes
10015	6c456d93-1b4a-48ae-bde7-e713f7fd8283	Female	2023-02-08	2023-02-10	14	Alexandria	1	0	0	0	0	0	2	Yes
10016	4b662fe0-c789-4c91-8ea5-102b2a14eafd	Male	2023-03-08	2023-03-11	63	Fairfax	0	0	0	0	0	0	3	Yes
10017	8b0ed975-e20e-4794-9de5-d2eb28596e80	Female	2023-04-13	2023-04-20	2	Virginia Beach	0	0	1	0	0	0	7	No
10018	44a912f7-49f8-4292-9d30-6afa5f380b5b	Female	2023-06-18	2023-06-18	68	Newport News	0	1	0	1	0	0	0	Yes
10019	93be6e0f-6776-4388-8d6f-ccd5d64f7187	Female	2023-02-01	2023-02-14	22	Norfolk	0	0	0	0	0	1	13	Yes
10020	5ff2b09f-084a-4ec4-bb91-80f58daf7c97	Male	2023-06-13	2023-06-15	52	Chesapeake	0	0	0	0	0	1	2	Yes
10021	d5e652b7-1a83-4240-b0a3-a4122cfc142d	Female	2023-04-06	2023-04-11	36	Richmond	0	0	0	1	0	1	5	No
10022	020359c1-941e-42e9-b492-12e96cc1bd55	Male	2023-04-17	2023-04-20	8	Chesapeake	0	1	0	0	0	0	3	No
10023	ce74ec56-db42-4295-95a2-b7e4351399ed	Female	2023-03-09	2023-03-21	14	Newport News	0	1	1	0	0	1	12	Yes
10024	e6035c26-0793-43db-afeb-0ef22dde29d8	Female	2023-06-26	2023-07-06	2	Fairfax	0	1	0	1	0	0	10	Yes
10025	feb1e270-69c8-4b17-bf33-5ada2e20766f	Female	2023-04-09	2023-04-20	92	Chesapeake	0	0	0	0	0	1	11	Yes
10026	cb725702-9192-418f-bbe3-f251f6096928	Female	2023-02-16	2023-02-23	69	Richmond	0	1	0	0	0	1	7	Yes

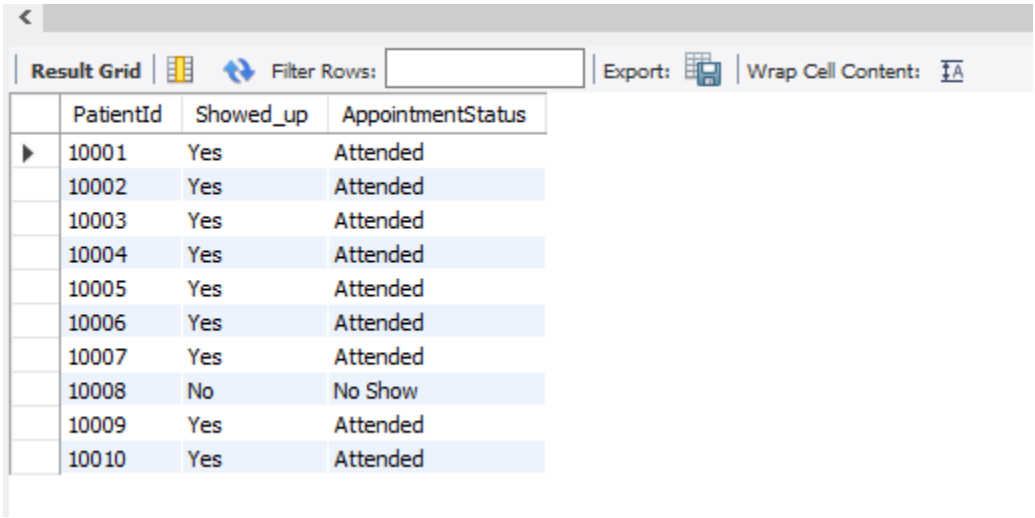
In this query, updates the shoed_up column in the Virginia_patients table and either null or an empty string with yes. This is the most important step in data analysis, as it finds all missing or

blank values in rows and standardizes them to yes. Also, it cleans the dataset and allows us to run filter and sort queries more reliably.

8. Add a new column AppointmentStatus using a CASE statement: ○ 'No Show' if Showed_up = 'No' ○ 'Attended' otherwise

Solution:

```
ALTER TABLE virginia_patients
ADD Column AppointmentStatus TEXT;
SELECT * FROM virginia_patients;
UPDATE virginia_patients
SET AppointmentStatus = Case
WHEN Showed_up = 'No' THEN 'No Show' ELSE 'Attended'
END;
SELECT * FROM virginia_patients;
SELECT PatientId, Showed_up,
CASE WHEN Showed_up = 'No' THEN 'No Show' ELSE 'Attended' END AS
AppointmentStatus
FROM virginia_patients LIMIT 10;
```



PatientId	Showed_up	AppointmentStatus
10001	Yes	Attended
10002	Yes	Attended
10003	Yes	Attended
10004	Yes	Attended
10005	Yes	Attended
10006	Yes	Attended
10007	Yes	Attended
10008	No	No Show
10009	Yes	Attended
10010	Yes	Attended

In this query, a new column “AppointmentStatus in the Virginia_patients table was added to store the information on whether the patient attended or missed their appointment.

Also, a CASE expression was used to create information for the new column AppointmentStatus as:

- If Showed_up = ‘No’ then AppointmentStatus = ‘No Show’
- Otherwise, if ‘yes’ then sets it to ‘Attended’

And it created a cleaner and more readable version of the AppointmentStatus column, allowing us to generate more calculations, reporting, and visualizations.

9. Filter appointments for diabetic patients with hypertension.

Solution:

```
SELECT * FROM virginia_patients;
```

SELECT * FROM virginia_patients
WHERE Diabetes = 1 AND Hypertension = 1;

```
71  --- 9. Filter appointments for diabetic patients with hypertension.
72  -- Solution:
73  select * FROM virginia_patients;
74  SELECT * FROM virginia_patients
75  WHERE Diabetes = 1 AND Hypertension = 1;
76
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up	AppointmentStatus
10023	ce74ec56-db42-4295-95a2-b7e4351399ed	Female	2023-03-09	2023-03-21	14	Newport News	0	1	1	0	0	1	12	Yes	Attended
10028	2f33d653-4421-4249-a3f6-b904fab34b7f	Female	2023-02-26	2023-03-11	25	Charlottesville	1	1	1	0	0	0	13	Yes	Attended
10045	8a38ef1d-015d-441f-8e22-0c2dea771919	Female	2023-05-19	2023-05-26	16	Newport News	0	1	1	1	0	1	7	No	No Show
10047	edfc1ab2-c198-4cfb-a5cf-9236bdce8c96	Female	2023-03-29	2023-04-05	66	Norfolk	0	1	1	0	0	1	7	Yes	Attended
10128	cf03c9ad-72d5-48d4-9ed6-d4366cfc3638	Male	2023-06-01	2023-06-01	91	Chesapeake	0	1	1	0	0	1	0	Yes	Attended
10168	2c01b5a6-c9dd-40a8-a78f-5465228dc422	Male	2023-02-05	2023-02-05	86	Alexandria	0	1	1	1	1	1	0	Yes	Attended
10184	d9e07cbd-1734-4957-8341-1e59a5680f62	Male	2023-06-25	2023-07-04	81	Alexandria	0	1	1	0	0	0	9	Yes	Attended
10216	6c74eed4-4578-44e6-b32b-ad052c682589	Male	2023-05-12	2023-05-26	75	Alexandria	1	1	1	0	0	1	14	No	No Show
10234	f9848030-9fa9-45b1-ae18-76034ad551ad	Female	2023-04-26	2023-05-04	51	Virginia Beach	0	1	1	0	0	0	8	Yes	Attended
10268	a5c5677f-d8af-426c-bd1d-53ca4e027022	Female	2023-02-04	2023-02-10	52	Virginia Beach	1	1	1	0	0	1	6	Yes	Attended
10314	fcc6febe-7b48-4fd1-8185-361480f377c2	Male	2023-02-13	2023-02-14	79	Roanoke	0	1	1	0	0	0	1	No	No Show
10378	0c44c0fd-e13c-464d-b23e-860e0ae35494	Male	2023-06-13	2023-06-15	87	Fairfax	0	1	1	0	0	0	2	Yes	Attended
10390	f38f922c-56c0-472f-b747-e5cd94571290	Male	2023-04-06	2023-04-13	29	Newport News	1	1	1	0	0	1	7	Yes	Attended
10409	14cb07a9-0180-4ceb-a665-c3027c0ba9d2	Female	2023-06-21	2023-07-05	22	Arlington	0	1	1	0	0	1	14	No	No Show
10411	ee31bc55-a973-48f7-8f7d-59dc4b9486e6	Male	2023-04-12	2023-04-13	15	Norfolk	0	1	1	0	0	0	1	Yes	Attended
10440	ce8a4293-673a-48f7-a193-ab0809d73a75	Male	2023-05-17	2023-05-27	40	Richmond	0	1	1	0	0	1	10	Yes	Attended
10514	4eaeba0b-f07a-4ab7-8b9f-82a53f76f3b0	Female	2023-03-29	2023-04-10	66	Arlington	0	1	1	1	0	1	12	Yes	Attended
10515	b5f861b0-1eda-428c-81f3-e01c7da1f511	Female	2023-06-12	2023-06-22	24	Richmond	1	1	1	0	0	1	10	Yes	Attended
10550	cb771603-8eef-4874-92c8-acc2d05e9278	Male	2023-05-21	2023-05-26	38	Chesapeake	1	1	1	0	0	1	5	Yes	Attended
10560	967c2be7-02e5-4343-88f9-75a10cf2b534	Female	2023-02-01	2023-02-14	68	Chesapeake	0	1	1	0	0	1	13	Yes	Attended
10564	fe94ff67-04b4-4910-bc16-16b146bb08ea	Male	2023-04-16	2023-04-18	87	Newport News	0	1	1	0	0	1	2	Yes	Attended
10594	e2beeee6-a113-4905-b158-6f2cd283dfa3	Female	2023-03-21	2023-04-02	85	Norfolk	0	1	1	0	0	1	12	Yes	Attended
10614	719c6ba4-1476-4da4-b25a-08ffe538ef54	Female	2023-01-21	2023-02-01	17	Alexandria	0	1	1	0	0	1	11	Yes	Attended
10615	63f15a67-4a0d-44cc-9043-1241c02edc3d	Female	2023-03-23	2023-03-27	38	Virginia Beach	1	1	1	0	0	1	4	No	No Show
10633	21178e9f-0b44-4f4d-9ef7-88de02241f4c	Male	2023-02-14	2023-02-17	34	Richmond	1	1	1	0	1	1	3	Yes	Attended
10634	a41d3b6e-e318-42ba-87e9-2ea5bb1e7e3a	Male	2023-06-11	2023-06-16	81	Richmond	0	1	1	0	0	0	5	Yes	Attended
10655	0b3968c3-2316-4221-b3aa-6da6972d5b66	Female	2023-03-10	2023-03-24	67	Richmond	0	1	1	1	0	0	14	Yes	Attended
10663	d6c69578-e2e5-4a89-888c-7a64361bc766	Male	2023-03-18	2023-03-24	11	Chesapeake	0	1	1	0	0	1	6	Yes	Attended
10673	02af2537-7298-4c1d-87cd-c1e36c4d79e0	Female	2023-01-06	2023-01-10	16	Newport News	0	1	1	0	0	1	4	Yes	Attended
10721	3d756daf-5279-4436-a224-a0d1ebeb50d9	Female	2023-04-12	2023-04-21	3	Newport News	0	1	1	0	0	0	9	Yes	Attended
10744	91824513-6fb9-4ec6-92d7-f029714bfa37	Male	2023-02-17	2023-02-20	98	Norfolk	1	1	1	0	0	1	3	No	No Show
10753	778446d0-77aa-4c1a-b30b-e62f7e25b982	Female	2023-05-18	2023-05-19	89	Roanoke	0	1	1	0	0	1	1	Yes	Attended
10784	b2f089b9-964a-452c-bb1e-e55a37b3ef4d	Female	2023-01-09	2023-01-23	9	Charlottesville	1	1	1	0	0	1	14	No	No Show
10795	4bede704-6fbf-4eed-9f55-2ac3c66b7b69	Male	2023-02-21	2023-02-24	31	Charlottesville	1	1	1	1	0	0	3	No	No Show
10796	becc1545-21bc-4cff-a0a7-b4a8376b1e42	Male	2023-05-04	2023-05-08	19	Arlington	0	1	1	0	0	1	4	Yes	Attended

virginia_patients 5

This query filters the rows from the Virginia_patients table for the patients who have both diabetes and hypertension. By running this query, we are able to identify the patients who are medically at high risk, and tracking their appointment attendance along with SMS reminders is significantly important for public health insights.

10. Order the records by Age in descending order and show only the top 5 oldest patients.

Solution:

SELECT * FROM virginia_patients
ORDER BY Age DESC LIMIT 5;
SELECT * FROM virginia_patients;

```
78  SELECT * FROM virginia_patients
79  ORDER BY Age DESC LIMIT 5;
80  SELECT * FROM virginia_patients;
81
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

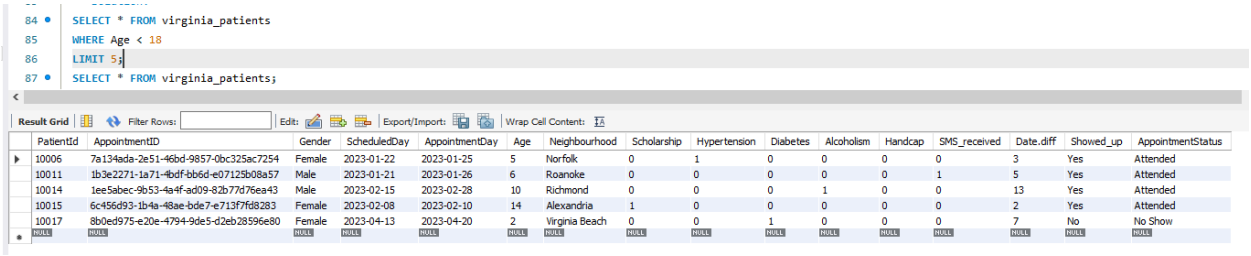
PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up	AppointmentStatus
19876	9e8f862b-537e-474e-9b5c-aace44fcc9f4	Female	2023-05-11	2023-05-25	99	Arlington	0	0	0	1	0	0	14	Yes	Attended
16161	d9738a63-c262-4e44-afdd-42c0484c6265	Female	2023-03-01	2023-03-04	99	Virginia Beach	0	1	0	0	0	1	3	Yes	Attended
16719	5f441ba8-c027-4798-8a58-1d55303808c8	Female	2023-06-09	2023-06-13	99	Alexandria	0	0	0	0	0	0	4	Yes	Attended
17398	95010ff8-191f-4163-9499-81fc739de36c	Male	2023-03-18	2023-04-01	99	Chesapeake	0	0	0	0	0	1	14	Yes	Attended
16231	eb8f6af2-d601-436b-9784-f6e61d112177	Male	2023-03-11	2023-03-23	99	Chesapeake	0	0	0	0	0	1	12	Yes	Attended
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

This query sorts all patients' data from the Virginia_patients table in descending order and displays only the five oldest patients. These results gave us a snapshot of patients who have a higher risk of chronic illness and need more medical monitoring.

11. Limit results to the first 5 appointments for patients under age 18.

Solution:

```
SELECT * FROM virginia_patients
WHERE Age < 18
LIMIT 5;
SELECT * FROM virginia_patients;
```



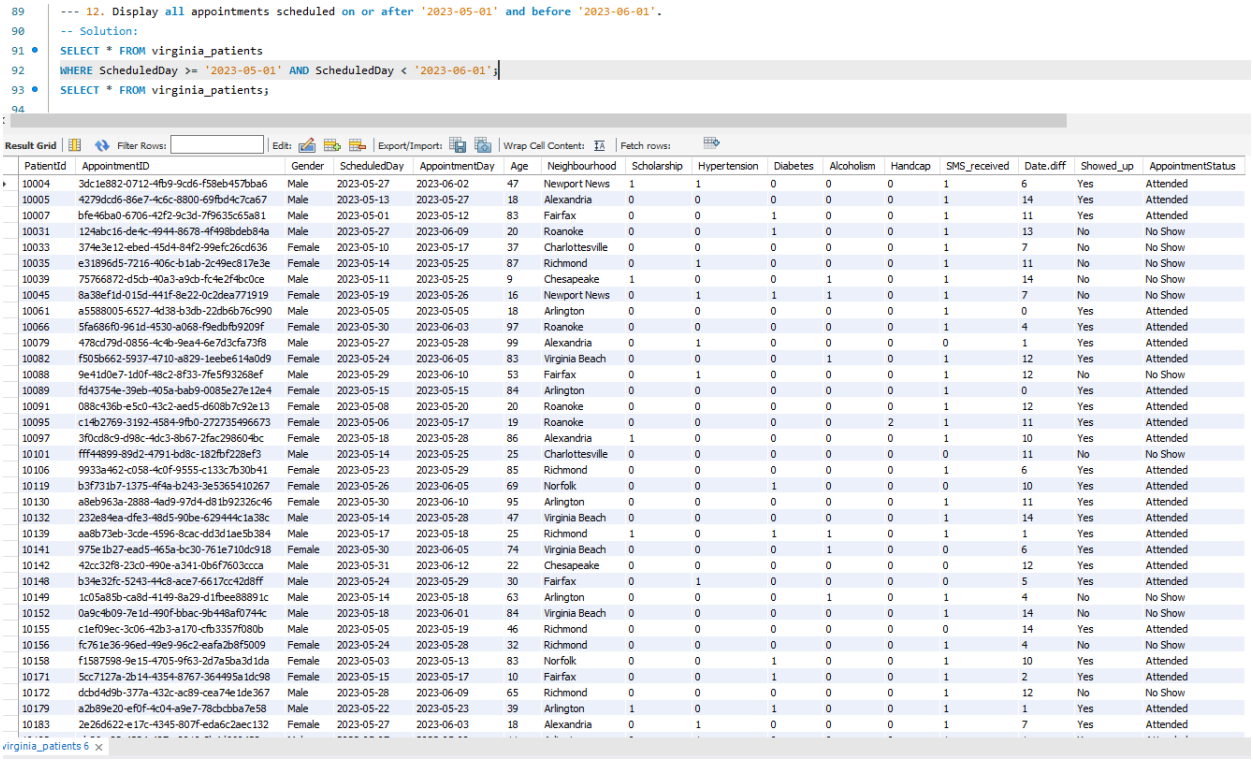
PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up	AppointmentStatus
10006	7a134ada-2e51-46bd-9857-0bc325ac7254	Female	2023-01-22	2023-01-25	5	Norfolk	0	1	0	0	0	0	3	Yes	Attended
10011	1b3e2271-1a71-4bdf-bb6d-e07125b08a57	Male	2023-01-21	2023-01-26	6	Roanoke	0	0	0	0	0	1	5	Yes	Attended
10014	1ee5abec-9b53-4a4f-ad09-82b77d76ea43	Male	2023-02-15	2023-02-28	10	Richmond	0	0	0	1	0	0	13	Yes	Attended
10015	6c456d93-1b4a-48ae-bde7-e713f7fd8283	Female	2023-02-08	2023-02-10	14	Alexandria	1	0	0	0	0	0	2	Yes	Attended
10017	8b0ed975-e20e-4794-9de5-d2eb28596e80	Female	2023-04-13	2023-04-20	2	Virginia Beach	0	0	1	0	0	0	7	No	No Show
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

This query filters the patients from the Virginia_pateints table only younger than eighteen (minors). It shows the first five appointments with the status attended, with one patient (10017) no show.

12. Display all appointments scheduled on or after '2023-05-01' and before '2023-06-01'.

Solution:

```
SELECT * FROM virginia_patients
WHERE ScheduledDay >= '2023-05-01' AND ScheduledDay < '2023-06-01';
SELECT * FROM virginia_patients;
```



PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up	AppointmentStatus
10004	3dc1e882-0712-4fb9-9cd6-f58eb457bba6	Male	2023-05-27	2023-06-02	47	Newport News	1	1	0	0	0	1	6	Yes	Attended
10005	4279dcd6-86e7-4c6c-8800-69fbd4c7ca67	Male	2023-05-13	2023-05-27	18	Alexandria	0	0	0	0	0	1	14	Yes	Attended
10007	bfe46ba0-6706-42f2-9c3d-7f9635c65a81	Male	2023-05-01	2023-05-12	83	Fairfax	0	0	1	0	0	1	11	Yes	Attended
10031	124abc16-de4c-4944-8678-4f498bdeb84a	Male	2023-05-27	2023-06-09	20	Roanoke	0	0	1	0	0	1	13	No	No Show
10033	374e3e12-ebed-45d4-84f2-99efc26cd636	Female	2023-05-10	2023-05-17	37	Charlottesville	0	0	0	0	0	1	7	No	No Show
10035	e31896d5-7216-406c-b1ab-2c49ec817e3e	Female	2023-05-14	2023-05-25	87	Richmond	0	1	0	0	0	1	11	No	No Show
10039	75766872-d5cb-40a3-a9cb-fc4e2f4bc0ce	Male	2023-05-11	2023-05-25	9	Chesapeake	1	0	0	1	0	1	14	No	No Show
10045	8a38ef1d-015d-441f-8e22-0c2dea771919	Female	2023-05-19	2023-05-26	16	Newport News	0	1	1	1	0	1	7	No	No Show
10061	a5588005-6527-4d38-b3db-22db6b76c990	Male	2023-05-05	2023-05-05	18	Arlington	0	0	0	0	0	1	0	Yes	Attended
10066	5fa686f0-961d-4530-a068-f9edbfb9209f	Female	2023-05-30	2023-06-03	97	Roanoke	0	0	0	0	0	1	4	Yes	Attended
10079	478cd79d-0856-4c4b-9ea4-6e7d3cfa73f8	Male	2023-05-27	2023-05-28	99	Alexandria	0	1	0	0	0	0	1	Yes	Attended
10082	f505b662-5937-4710-a829-1eebe614a0d9	Female	2023-05-24	2023-06-05	83	Virginia Beach	0	0	0	1	0	1	12	Yes	Attended
10088	9e41d0e7-1d0f-48c2-8f33-7fe5f93268ef	Male	2023-05-29	2023-06-10	53	Fairfax	0	1	0	0	0	1	12	No	No Show
10089	fd43754e-39eb-405a-bab9-0085e27e12e4	Female	2023-05-15	2023-05-15	84	Arlington	0	0	0	0	0	1	0	Yes	Attended
10091	088c436b-e5c0-43c2-aed5-d608b7c92e13	Female	2023-05-08	2023-05-20	20	Roanoke	0	0	0	0	0	1	12	Yes	Attended
10095	c14b2769-3192-4584-9fb0-272735496673	Female	2023-05-06	2023-05-17	19	Roanoke	0	0	0	0	2	1	11	Yes	Attended
10097	3f0cd8c9-d98c-4dc3-8b67-2fac298604bc	Female	2023-05-18	2023-05-28	86	Alexandria	1	0	0	0	0	1	10	Yes	Attended
10101	fff44899-89d2-4791-bd8c-182fbf228ef3	Male	2023-05-14	2023-05-25	25	Charlottesville	0	0	0	0	0	0	11	No	No Show
10106	9933a462-c058-4c0f-9555-c133c7b30b41	Female	2023-05-23	2023-05-29	85	Richmond	0	0	0	0	0	1	6	Yes	Attended
10119	b3f731b7-1375-4f4a-b243-3e5365410267	Female	2023-05-26	2023-06-05	69	Norfolk	0	0	1	0	0	0	10	Yes	Attended
10130	a8eb963a-2888-4ad9-97d4-d81b92326c46	Female	2023-05-30	2023-06-10	95	Arlington	0	0	0	0	0	1	11	Yes	Attended
10132	232e84ea-dfe3-48d5-90be-629444c1a38c	Male	2023-05-14	2023-05-28	47	Virginia Beach	0	0	0	0	0	1	14	Yes	Attended
10139	aa8b73eb-3cde-4596-8cac-dd3d1ae5b384	Male	2023-05-17	2023-05-18	25	Richmond	1	0	1	1	0	1	1	Yes	Attended
10141	975e1b27-ead5-465a-bc30-761e710dc918	Female	2023-05-30	2023-06-05	74	Virginia Beach	0	0	0	1	0	0	6	Yes	Attended
10142	42cc32f8-23c0-490e-a341-0b6f7603ccca	Male	2023-05-31	2023-06-12	22	Chesapeake	0	0	0	0	0	0	12	Yes	Attended
10148	b34e32fc-5243-44c8-ace7-6617cc42d8ff	Male	2023-05-24	2023-05-29	30	Fairfax	0	1	0	0	0	0	5	Yes	Attended
10149	1c05a85b-ca8d-4149-8a29-d1fbee88891c	Male	2023-05-14	2023-05-18	63	Arlington	0	0	0	1	0	1	4	No	No Show
10152	0a9c4b09-7e1d-490f-bbac-9b448af0744c	Male	2023-05-18	2023-06-01	84	Virginia Beach	0	0	0	0	0	1	14	No	No Show
10155	c1ef09ec-3c06-42b3-a170-cfb3357f080b	Male	2023-05-05	2023-05-19	46	Richmond	0	0	0	0	0	0	14	Yes	Attended
10156	fc761e36-96ed-49e9-96c2-eafa2b8f5009	Female	2023-05-24	2023-05-28	32	Richmond	0	0	0	0	0	1	4	No	No Show
10158	f1587598-9e15-4705-9f63-2d7a5ba3d1da	Female	2023-05-03	2023-05-13	83	Norfolk	0	0	1	0	0	1	10	Yes	Attended
10171	5cc7127a-2b14-4354-8767-364495a1dc98	Female	2023-05-15	2023-05-17	10	Fairfax	0	0	1	0	0	1	2	Yes	Attended
10172	dcbd4d9b-377a-432c-ac89-cea74e1de367	Female	2023-05-28	2023-06-09	65	Richmond	0	0	0	0	0	1	12	No	No Show
10179	a2b89e20-ef0f-4c04-a9e7-78cbcbba7e58	Male	2023-05-22	2023-05-23	39	Arlington	1	0	1	0	0	1	1	Yes	Attended
10183	2e26d622-e17c-4345-807f-eda6c2aec132	Female	2023-05-27	2023-06-03	18	Alexandria	0	1	0	0	0	1	7	Yes	Attended

This query filters or returns all the appointments that were scheduled in May. Also, it gave us insights into how some appointments were scheduled in a specific time frame, along with attendance rate and SMS reminders.

Aggregation and CASE:

13. Find the average age of patients for each gender.

Solution:

SELECT Gender, AVG(Age) as average_age FROM virginia_patients GROUP BY Gender;
SELECT * FROM virginia_patients;

```
97 •  SELECT Gender, AVG(Age) as average_age FROM virginia_patients GROUP BY Gender;
```

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

Gender	average_age
Male	50.3641
Female	49.7444

This query calculates the average age of the patients separately for each gender. So, the average age of male patients from the Virginia_patients is 50.36 years old, while the average age of female patients is 49.74 years old.

14. Count how many patients received SMS reminders, grouped by Showed_up status.

Solution:

SELECT Showed_up, Count(*) as SMS_reminders FROM virginia_patients WHERE SMS_received = 1 GROUP BY Showed_up;
SELECT * FROM virginia_patients;

```
102 •  SELECT Showed_up, Count(*) as SMS_reminders FROM virginia_patients WHERE SMS_received = 1 GROUP BY Showed_up;
103 •  SELECT * FROM virginia_patients;
```

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

Showed_up	SMS_reminders
Yes	5405
No	1478

This query counts the Showed_up and SMS_reminders of the patients from the Virginia_patients table, giving us the count of SMS reminders with patients who showed up for their appointment is 5405, with 1478 patients with no show.

So a high number of show-ups indicates that SMS reminders are helping patients to attend their appointments. However, the number of no-shows may indicate other factors influencing the patients' attendance, such as traffic, motivation, and their commitment to health issues and lifestyle.

15. Count no-show appointments in each neighborhood using GROUP BY.

Solution:

SELECT Neighbourhood, COUNT(*) AS no_show_count FROM virginia_patients WHERE Showed_up = 'No' GROUP BY Neighbourhood;
SELECT * FROM virginia_patients;

```
106 • SELECT Neighbourhood, COUNT(*) AS no_show_count FROM virginia_patients WHERE Showed_up = 'No' GROUP BY Neighbourhood;
107 • SELECT * FROM virginia_patients;
```

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

Neighbourhood	no_show_count
Newport News	188
Virginia Beach	196
Richmond	193
Chesapeake	210
Roanoke	214
Charlottesville	217
Norfolk	211
Alexandria	211
Fairfax	215
Arlington	204

This query filters the datasets for patients who did not show up for their appointments and then groups by the results for each neighbourhood to give no_show_count for each neighbourhood. The minimum no_show_count was in Newport News with a value of 188, while the maximum was in Charlottesvile with a value of 217.
This analysis helps us to identify which neighbourhoods need more mobile clinics and more reminder outreach.

16. Show neighborhoods with more than 100 total appointments (HAVING clause).
Solution:
SELECT Neighbourhood, COUNT(PatientId) as total_appointments FROM virginia_patients GROUP BY Neighbourhood Having COUNT(PatientId) > 100;

```
110 • SELECT Neighbourhood, COUNT(PatientId) as total_appointments FROM virginia_patients GROUP BY Neighbourhood Having COUNT(PatientId) > 100;
```

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

Neighbourhood	total_appointments
Alexandria	1018
Arlington	1027
Newport News	991
Norfolk	999
Fairfax	977
Virginia Beach	946
Roanoke	980
Richmond	1014
Charlottesville	956
Chesapeake	1008

This query helps us to identify the busiest neighbourhoods with total appointments from the Virginia_patients table, and helps us to allocate more resources (more doctors, SMS reminders) to these busy neighbourhoods for effective attendance among patients.
So by running this query, we are able to find the neighbourhood with the highest total appointments is Arlington with a value of 1027, followed by Alexandra with a value of 1018, Richmond with total appointments of 1014, and Chesapeake with appointments of 1008.

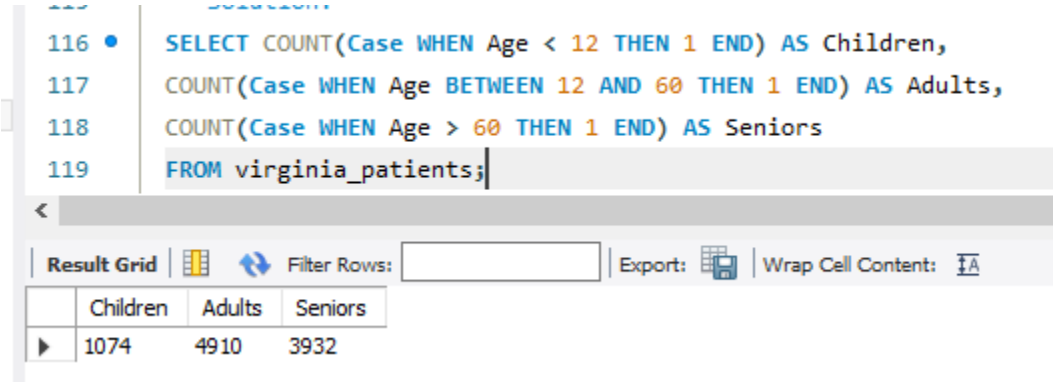
17. Use CASE to calculate the total number of:

- **children (Age < 12)**

- **adults (Age BETWEEN 12 AND 60)**
- **seniors (Age > 60)**

Solution:

SELECT COUNT(Case WHEN Age < 12 THEN 1 END) AS Children,
COUNT(Case WHEN Age BETWEEN 12 AND 60 THEN 1 END) AS Adults,
COUNT(Case WHEN Age > 60 THEN 1 END) AS Seniors
FROM virginia_patients;



This query uses a CASE() expression inside a COUNT() function to classify the patients into three age groups and then count them. It returns the value of 1074 children, 4910 adults, and 3932 seniors. It will help us to identify age-related health trend analysis and evaluate the effectiveness of SMS reminders within each age group.

18. Show whether patients are more likely to miss appointments on certain days of the week. (This can reveal patterns like more no-shows on Mondays or weekends, helping clinics to adjust scheduling.) Steps to follow for question # 20 (You can use any approach to solve this question): (Use the AppointmentDay column in the function dayname() to extract the day name (like Monday, Tuesday, etc.). Count how many appointments were scheduled, how many showed up (showed_up = "yes"), and how many were missed (Showed_up = 'No') on each day. Calculate the percentage of shows and no-shows for better comparison between days. Formula: (count of Showed_up = 'yes' / total appointment count) * 100. Use the round function to show up to two decimal points. Sort the result by No_Show_Percent in descending order to see the worst-performing days first.

Solution:

SELECT * FROM virginia_patients;
SELECT DAYNAME(AppointmentDay) as Appointment_Day, COUNT(*) AS Total_Appointments,
COUNT(CASE WHEN Showed_up = 'Yes' THEN 1 END) AS Showed_up,
COUNT(CASE WHEN Showed_up = 'No' THEN 1 END) AS No_Show,

ROUND((COUNT(CASE WHEN Showed_up = 'No' THEN 1 END) * 100.0) / COUNT(*), 2)
AS No_Show_Percent FROM virginia_patients
GROUP BY DAYNAME(AppointmentDay)
ORDER BY No_Show_Percent DESC;

```
127 • SELECT * FROM virginia_patients;
128 • SELECT DAYNAME(AppointmentDay) as Appointment_Day, COUNT(*) AS Total_Appointments,
129   COUNT(CASE WHEN Showed_up = 'Yes' THEN 1 END) AS Showed_up,
130   COUNT(CASE WHEN Showed_up = 'No' THEN 1 END) AS No_Show,
131   ROUND((COUNT(CASE WHEN Showed_up = 'No' THEN 1 END) * 100.0) / COUNT(*), 2) AS No_Show_Percent FROM virginia_patients
132   GROUP BY DAYNAME(AppointmentDay)
133   ORDER BY No_Show_Percent DESC;
```

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

Appointment_Day	Total_Appointments	Showed_up	No_Show	No_Show_Percent
Sunday	1417	1101	316	22.30
Monday	1365	1074	291	21.32
Saturday	1419	1122	297	20.93
Thursday	1488	1180	308	20.70
Tuesday	1463	1162	301	20.57
Friday	1382	1102	280	20.26
Wednesday	1382	1116	266	19.25

This query helps us to find the total appointments, showed_up, no_show, with no_show_percent from each day of the week, and gives us an overview of the worst day of the week in regard to total appointments with no shows.
The results show us Sunday and Monday with the worst attendance rate for appointments, with percentages of 22% and 21%. On the other hand the Wednesday has the lowest no-show rate of appointments with a percentage of 19.25.

Window Functions

19. Track how appointments accumulate over time in each neighbourhood. (Running Total of Appointments per Day) In simple words: How many appointments were there each day, and how do the total appointments keep adding up over time in each neighborhood?

Solution:

SELECT Neighbourhood, AppointmentDay, COUNT(*) AS daily_appointments,
SUM(COUNT(*)) OVER(PARTITION BY Neighbourhood
ORDER BY AppointmentDay ROWS BETWEEN UNBOUNDED preceding AND CURRENT ROW) AS Running_total
FROM virginia_patients
GROUP BY Neighbourhood, AppointmentDay
ORDER BY Neighbourhood, AppointmentDay;
SELECT * FROM virginia_patients;

```sql
SELECT Neighbourhood, AppointmentDay, COUNT(*) AS daily_appointments,
SUM(COUNT(*)) OVER(PARTITION BY Neighbourhood
ORDER BY AppointmentDay ROWS BETWEEN UNBOUNDED preceding AND CURRENT ROW) AS Running_total
FROM virginia_patients
GROUP BY Neighbourhood, AppointmentDay
ORDER BY Neighbourhood, AppointmentDay;
SELECT * FROM virginia_patients;
```

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

Neighbourhood	AppointmentDay	daily_appointments	Running_total
Alexandria	2023-01-03	1	1
Alexandria	2023-01-05	1	2
Alexandria	2023-01-06	1	3
Alexandria	2023-01-07	5	8
Alexandria	2023-01-08	3	11
Alexandria	2023-01-09	3	14
Alexandria	2023-01-10	5	19
Alexandria	2023-01-11	6	25
Alexandria	2023-01-12	5	30
Alexandria	2023-01-13	2	32
Alexandria	2023-01-14	5	37
Alexandria	2023-01-15	2	39
Alexandria	2023-01-16	2	41
Alexandria	2023-01-17	3	44
Alexandria	2023-01-18	8	52
Alexandria	2023-01-19	5	57
Alexandria	2023-01-20	5	62
Alexandria	2023-01-21	5	67
Alexandria	2023-01-22	5	72
Alexandria	2023-01-23	4	76
Alexandria	2023-01-24	4	80
Alexandria	2023-01-25	2	82
Alexandria	2023-01-26	8	90
Alexandria	2023-01-27	9	99
Alexandria	2023-01-28	5	104
Alexandria	2023-01-29	6	110
Alexandria	2023-01-30	10	120
Alexandria	2023-01-31	8	128
Alexandria	2023-02-01	10	138
Alexandria	2023-02-02	4	142
Alexandria	2023-02-03	5	147
Alexandria	2023-02-04	7	154
Alexandria	2023-02-05	9	163
Alexandria	2023-02-06	11	174
Alexandria	2023-02-07	7	181

This query finds the daily appointments in each neighbourhood by calculating the total running time. It gives us how total volume of patients growing over time and helps us to monitor workload trends and higher demands of healthcare in each neighbourhood.

20. Use Dense_Rank() to rank patients by age within each gender group.

Solution:

SELECT PatientId, Gender, Age, DENSE_RANK() OVER(PARTITION BY Gender ORDER BY Age DESC) AS age_rank FROM virginia_patients;

SELECT * FROM virginia_patients;

```
148    --- 20. Use Dense_Rank() to rank patients by age within each gender group--
149    --- Solution:
150    SELECT PatientId, Gender, Age, DENSE_RANK() OVER(PARTITION BY Gender
151    ORDER BY Age DESC) AS age_rank FROM virginia_patients;
152    SELECT * FROM virginia_patients;
153    --- 21. How many days have passed since the last appointment in the same neighborhood?
```

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

PatientId	Gender	Age	age_rank
19876	Female	99	1
11155	Female	99	1
11848	Female	99	1
13411	Female	99	1
10546	Female	99	1
10246	Female	99	1
11321	Female	99	1
13468	Female	99	1
13184	Female	99	1
12660	Female	99	1
14162	Female	99	1
11204	Female	99	1
12911	Female	99	1
14877	Female	99	1
13704	Female	99	1
11077	Female	99	1
11141	Female	99	1
13827	Female	99	1
12436	Female	99	1
12330	Female	99	1
14734	Female	99	1
13167	Female	99	1
14219	Female	99	1
14005	Female	99	1
12493	Female	99	1
10425	Female	99	1
12982	Female	99	1
11766	Female	99	1
17039	Female	99	1
18042	Female	99	1
18501	Female	99	1
18729	Female	99	1
17160	Female	99	1
18236	Female	99	1
17985	Female	99	1

Result 11

This query helps us to rank patients based on their age within their gender group. The results show the oldest patient is a female with age 99 and rank 1, helping us to find the individuals at high risk of health issues.

21. How many days have passed since the last appointment in the same neighborhood?
(Hint: DATEDIFF and Lag) (This helps to see how frequently appointments are happening in each neighborhood.)
Solution:

```
SELECT PatientID, Neighbourhood, AppointmentDay,
LAG(AppointmentDay) OVER(PARTITION BY Neighbourhood
ORDER BY AppointmentDay)
AS Previous_Appointment, DATEDIFF(AppointmentDay, LAG(AppointmentDay)
OVER(PARTITION BY Neighbourhood ORDER BY AppointmentDay)) AS
Since_Last_Appointment
FROM virginia_patients ORDER BY Neighbourhood, AppointmentDay;
SELECT * FROM virginia_patients;
```

```
156 • SELECT PatientID, Neighbourhood, AppointmentDay,
157   LAG(AppointmentDay) OVER(PARTITION BY Neighbourhood
158   ORDER BY AppointmentDay)
159   AS Previous_Appointment, DATEDIFF(AppointmentDay, LAG(AppointmentDay)
160   OVER(PARTITION BY Neighbourhood ORDER BY AppointmentDay)) AS Since_Last_Appointment
161   FROM virginia_patients ORDER BY Neighbourhood, AppointmentDay;
```

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

PatientID	Neighbourhood	AppointmentDay	Previous_Appointment	Since_Last_Appointment
12161	Alexandria	2023-01-03	NULL	NULL
16689	Alexandria	2023-01-05	2023-01-03	2
15883	Alexandria	2023-01-06	2023-01-05	1
14045	Alexandria	2023-01-07	2023-01-06	1
15384	Alexandria	2023-01-07	2023-01-07	0
17174	Alexandria	2023-01-07	2023-01-07	0
11764	Alexandria	2023-01-07	2023-01-07	0
19437	Alexandria	2023-01-07	2023-01-07	0
12710	Alexandria	2023-01-08	2023-01-07	1
13764	Alexandria	2023-01-08	2023-01-08	0
11807	Alexandria	2023-01-08	2023-01-08	0
11839	Alexandria	2023-01-09	2023-01-08	1
16026	Alexandria	2023-01-09	2023-01-09	0
19779	Alexandria	2023-01-09	2023-01-09	0
19435	Alexandria	2023-01-10	2023-01-09	1
18487	Alexandria	2023-01-10	2023-01-10	0
19748	Alexandria	2023-01-10	2023-01-10	0
13612	Alexandria	2023-01-10	2023-01-10	0
15878	Alexandria	2023-01-10	2023-01-10	0
15286	Alexandria	2023-01-11	2023-01-10	1
19752	Alexandria	2023-01-11	2023-01-11	0
19706	Alexandria	2023-01-11	2023-01-11	0
19312	Alexandria	2023-01-11	2023-01-11	0
12564	Alexandria	2023-01-11	2023-01-11	0
11660	Alexandria	2023-01-11	2023-01-11	0
12060	Alexandria	2023-01-12	2023-01-11	1

Result 31

This query tells how many days have passed since the last appointment in each neighbourhood from the Virginia_patients table. The Alexandria neighbourhood has an appointment every 2 or 1 days, with some days having no appointment.

These results help us to identify neighbourhoods that need more resource allocation (high volume of appointments) and spot patterns in appointments from regular to irregular.

22. Which neighborhoods have the highest number of missed appointments? Use DENSE_RANK() to rank neighborhoods based on the number of no-show appointments.

Solution:

SELECT Neighbourhood, COUNT(*) AS missed_appointments,
DENSE_RANK() OVER(ORDER BY COUNT(*) DESC)

AS Missed_rank FROM virginia_patients WHERE Showed_up = 'No'
GROUP BY Neighbourhood ORDER BY Missed_rank;
SELECT * FROM virginia_patients;

```
166 •  SELECT Neighbourhood, COUNT(*) AS missed_appointments,
167    DENSE_RANK() OVER(ORDER BY COUNT(*) DESC)
168    AS Missed_rank FROM virginia_patients WHERE Showed_up = 'No'
169    GROUP BY Neighbourhood ORDER BY Missed_rank;
170 •  SELECT * FROM virginia_patients;
```

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

Neighbourhood	missed_appointments	Missed_rank
Charlottesville	217	1
Fairfax	215	2
Roanoke	214	3
Norfolk	211	4
Alexandria	211	4
Chesapeake	210	5
Arlington	204	6
Virginia Beach	196	7
Richmond	193	8
Newport News	188	9

This query shows us the most important information because it identifies which neighbourhood had the most missed appointments (meaning patients can not show up). This information is found through the DENSE_RANK() function. This function first filters the entire data set for missed appointments. After that, it filters each neighbourhood and ranks them from 1 to 10 based on total missed appointments. For example, Charlottesvile ranked 1 as the worst neighbourhood for total missed appointments compared to Newport News, which ranked 9 with only 188 missed appointments.

This function in SQL gave us not only data insights but also helped us to allocate more resources to the neighbourhoods with higher missed appointments. Also, it helped us to focus more on SMS reminders and more outreach for the neighbourhoods with the highest missed attendance rate.

Subqueries and CTEs:

23. Continuing from Q. 21, show the neighbourhoods with the second and third highest no-show counts.

Solution:

WITH Noshow_counts AS(SELECT Neighbourhood, COUNT(*) AS Missed_appointments,
DENSE_RANK() OVER(ORDER BY COUNT(*) DESC) AS missed_rank
FROM virginia_patients
WHERE Showed_up = 'No'
GROUP BY Neighbourhood)

SELECT * FROM Noshow_counts WHERE missed_rank IN (2, 3);
SELECT * FROM virginia_patients;

```
173    --- Solution:
174    WITH Noshow_counts AS(SELECT Neighbourhood, COUNT(*) AS Missed_appointments,
175    DENSE_RANK() OVER(ORDER BY COUNT(*) DESC) AS missed_rank
176    FROM virginia_patients
177    WHERE Showed_up = 'No'
178    GROUP BY Neighbourhood)
179    SELECT * FROM Noshow_counts WHERE missed_rank IN (2, 3);
```

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

Neighbourhood	Missed_appointments	missed_rank
Fairfax	215	2
Roanoke	214	3

This query identifies and ranks the second and third worst neighbourhoods with the highest missed appointments. For this, a common table expression and the dense_rank function were used to filter our datasets in descending no-show counts and then rank them second and third with the most missed appointments or no-shows.
In our datasets, Fairfax and Roanoke rank second and third with the highest number of missed appointments.

24. Find out all female patients that have an age higher than the average age of all female patients

Solution:

SELECT * FROM virginia_patients
WHERE Gender = 'Female'
AND Age > (SELECT AVG(Age)
FROM virginia_patients
WHERE Gender = 'Female');
SELECT * FROM virginia_patients;

```sql
--- 24. Find out all female patients that have an age higher than the average age of all female patients.
--- Solution:
SELECT * FROM virginia_patients
WHERE Gender = 'Female'
AND Age > (SELECT AVG(Age)
FROM virginia_patients
WHERE Gender = 'Female');
```

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

PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up	AppointmentStatus
10002	28a625e1-4133-4ed1-9821-8c7556c8c3c7	Female	2023-02-19	2023-02-26	70	Alexandria	0	0	0	0	0	1	7	Yes	Attended
10018	44a912f7-49f8-4292-9d30-6afa5f380b5b	Female	2023-06-18	2023-06-18	68	Newport News	0	1	0	1	0	0	0	Yes	Attended
10025	feb1e270-69c8-4b17-bf33-5ada2e20766f	Female	2023-04-09	2023-04-20	92	Chesapeake	0	0	0	0	0	1	11	Yes	Attended
10026	cb725702-9192-418f-bbe3-f251f6096928	Female	2023-02-16	2023-02-23	69	Richmond	0	1	0	0	0	1	7	Yes	Attended
10030	6c1c0ab6-0f80-44cc-93d2-9821d42de47a	Female	2023-03-10	2023-03-24	71	Roanoke	0	1	0	0	0	0	14	No	No Show
10034	666a7861-1494-4dbd-bf6f-4f21ce64a89a	Female	2023-03-09	2023-03-16	71	Norfolk	0	1	0	0	0	1	7	No	No Show
10035	e31896d5-7216-406c-b1ab-2c49ec817e3e	Female	2023-05-14	2023-05-25	87	Richmond	0	1	0	0	0	1	11	No	No Show
10047	edfc1ab2-c198-4cfb-a5cf-9236bdce8c96	Female	2023-03-29	2023-04-05	66	Norfolk	0	1	1	0	0	1	7	Yes	Attended
10049	ba80c88b-9da3-4777-998c-1a48e187ab6d	Female	2023-04-28	2023-05-05	94	Charlottesville	0	0	0	0	0	1	7	Yes	Attended
10056	d27ebe78-1e2c-4bca-9295-626be5121da2	Female	2023-02-20	2023-03-06	84	Newport News	0	0	0	0	0	0	14	No	No Show
10057	a6f7b85d-591a-41ea-b8e2-aae3633ad91a	Female	2023-06-02	2023-06-08	85	Virginia Beach	0	0	0	0	0	1	6	Yes	Attended
10066	5fa686f0-961d-4530-a068-f9edbfb9209f	Female	2023-05-30	2023-06-03	97	Roanoke	0	0	0	0	0	1	4	Yes	Attended
10069	6b2fa6bc-00b6-4b48-90e9-9af04d257207	Female	2023-01-14	2023-01-15	50	Richmond	0	0	0	1	0	1	1	No	No Show
10071	defc4d67-d8b3-480d-a9ac-2a33c0e7d2f1	Female	2023-04-11	2023-04-15	62	Richmond	1	0	0	0	0	0	4	No	No Show
10072	8baf1516-0b6a-4167-ab6d-ee52058e2844	Female	2023-02-06	2023-02-13	56	Charlottesville	0	0	0	0	0	1	7	Yes	Attended
10074	d7a7d4f6-f5b7-4414-90a6-c5b41a70bcdb	Female	2023-04-12	2023-04-12	72	Alexandria	0	0	0	0	0	0	0	No	No Show
10078	d05858ea-e1d2-4681-8c9e-7893f1ca8560	Female	2023-01-17	2023-01-30	52	Norfolk	1	0	1	0	0	1	13	Yes	Attended
10082	f505b662-5937-4710-a829-1eebe614a0d9	Female	2023-05-24	2023-06-05	83	Virginia Beach	0	0	0	1	0	1	12	Yes	Attended
10089	fd43754e-39eb-405a-bab9-0085e27e12e4	Female	2023-05-15	2023-05-15	84	Arlington	0	0	0	0	0	1	0	Yes	Attended
10090	cbd4b8db-541c-4f78-889f-3f1a566fe0df	Female	2023-03-06	2023-03-07	85	Roanoke	0	0	0	0	2	1	1	Yes	Attended
10093	cefc3837-8dec-4c17-902b-368081196352	Female	2023-03-16	2023-03-29	93	Chesapeake	1	1	0	0	0	1	13	Yes	Attended
10096	e009a2cd-3157-409a-bedf-dad0c6988643	Female	2023-03-31	2023-04-06	87	Fairfax	0	1	0	0	0	1	6	Yes	Attended
10097	3f0cd8c9-d98c-4dc3-8b67-2fac298604bc	Female	2023-05-18	2023-05-28	86	Alexandria	1	0	0	0	0	1	10	Yes	Attended
10104	2964aa99-1728-4576-852d-52fac0ae3ba6	Female	2023-03-02	2023-03-05	60	Roanoke	1	1	0	0	0	0	3	Yes	Attended
10106	9933a462-c058-4c0f-9555-c133c7b30b41	Female	2023-05-23	2023-05-29	85	Richmond	0	0	0	0	0	1	6	Yes	Attended
10108	224717f9-9d62-405d-99e4-8a169c9a3db3	Female	2023-01-15	2023-01-24	73	Arlington	0	0	0	0	0	0	9	Yes	Attended
10109	24843f73-1e2e-4b93-8e72-f8dff38d906a	Female	2023-01-11	2023-01-25	61	Richmond	0	0	0	0	0	1	14	Yes	Attended
10119	b3f731b7-1375-4f4a-b243-3e5365410267	Female	2023-05-26	2023-06-05	69	Norfolk	0	0	1	0	0	0	10	Yes	Attended
10121	20cbbd96-6699-4ac3-b8f8-d92a3b6c28c5	Female	2023-04-04	2023-04-16	56	Richmond	0	0	0	0	0	1	12	Yes	Attended
10123	26b16c7c-5189-479a-b934-2fbd46296ada	Female	2023-01-24	2023-01-24	87	Charlottesville	0	1	0	0	0	1	0	Yes	Attended
10127	9eed7567-e3ee-42db-8680-ee6db3831f62	Female	2023-03-25	2023-04-05	55	Arlington	0	0	0	0	0	1	11	Yes	Attended
10129	01278844-06c5-4863-ba83-795dc4363bea	Female	2023-02-08	2023-02-17	56	Norfolk	1	1	0	0	0	0	9	Yes	Attended
10130	a8eb963a-2888-4ad9-97d4-d81b92326c46	Female	2023-05-30	2023-06-10	95	Arlington	0	0	0	0	0	1	11	Yes	Attended
10131	e702c0d8-8875-4591-a11d-b469bde85aab	Female	2023-01-04	2023-01-16	75	Richmond	0	0	0	0	0	0	12	Yes	Attended
10141	975e1b27-ead5-465a-bc30-761e710dc918	Female	2023-05-30	2023-06-05	74	Virginia Beach	0	0	0	1	0	0	6	Yes	Attended

virginia_patients 12

This subquery filters the dataset and returns only those females whose average age is greater than the average age of all female patients in the Virginia_patients table. And this query helps us to identify the age-related patterns of missed or attended appointments. It will also help us to find out the chronic illness of these older female patients and how we can make our SMS reminders more effective for them to not miss a health appointment.

25. Find out all details of the most recent appointment in each neighbourhood based on appointmentday. Hint: First, find out what the most recent appointmentday in each neighbourhood is. Then, you can simply join this result (think CTE) to your main table to get details of the latest appointment within each neighbourhood

Solution:

```sql
WITH Latest_appointments AS(SELECT Neighbourhood,
MAX(AppointmentDay) AS Most_recent_day
FROM virginia_patients
GROUP BY Neighbourhood)
SELECT vp.* FROM virginia_patients vp
JOIN Latest_appointments la ON vp.Neighbourhood = la.Neighbourhood
AND vp.AppointmentDay = la.Most_recent_day;
SELECT * FROM virginia_patients;
```

```
193 • WITH Latest_appointments AS(SELECT Neighbourhood,
194   MAX(AppointmentDay) AS Most_recent_day
195   FROM virginia_patients
196   GROUP BY Neighbourhood)
197   SELECT vp.* FROM virginia_patients vp
198   JOIN Latest_appointments la ON vp.Neighbourhood = la.Neighbourhood
199   AND vp.AppointmentDay = la.Most_recent_day;
```

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

PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up	AppointmentStatus
10288	0a7f9c87-55a3-4387-8edd-6053845ba1d0	Male	2023-06-26	2023-07-10	77	Roanoke	0	0	0	0	0	1	14	Yes	Attended
12050	ec4643db-3c20-47bf-9815-c81bca43f7c5	Male	2023-06-27	2023-07-11	59	Charlottesville	0	1	0	0	0	1	14	Yes	Attended
13426	158118d-19d9-4470-a4a9-c4277dcded63	Female	2023-06-28	2023-07-12	69	Virginia Beach	0	0	0	0	0	0	14	Yes	Attended
13911	62b6c581-22ac-4bed-ad11-2b4cbe6ad03d	Female	2023-06-29	2023-07-12	85	Newport News	1	0	0	0	0	1	13	Yes	Attended
14213	b8011c29-6c03-41b1-be0e-00229aa28ce1	Male	2023-06-29	2023-07-12	76	Alexandria	0	0	0	0	0	1	13	Yes	Attended
14858	3b711723-dff7-4df4-8647-7f744c54e2e3	Male	2023-06-26	2023-07-10	87	Fairfax	0	0	1	0	0	0	14	Yes	Attended
15202	2159da73-dadf-41c5-bdcb-a8f8c9601d84	Male	2023-06-28	2023-07-11	54	Arlington	0	0	0	0	0	1	13	No	No Show
15561	47764ac6-9d73-4b20-b22c-055524e8811a	Male	2023-06-29	2023-07-12	6	Chesapeake	0	0	0	0	0	1	13	Yes	Attended
15638	d25669ed-80f9-4ca0-bae8-9e86e750184f	Female	2023-06-29	2023-07-11	44	Arlington	0	0	0	0	0	1	12	No	No Show
15727	9b488a0b-60e0-44a7-8642-7b81157e8c58	Male	2023-06-29	2023-07-13	77	Richmond	0	0	0	1	0	1	14	Yes	Attended
15935	1ee69b68-e7e0-42b0-8176-f8102761cfcf	Male	2023-06-28	2023-07-12	62	Newport News	0	0	0	0	0	1	14	Yes	Attended
15972	4f998d73-0796-483a-a286-85f7795dc229	Female	2023-06-28	2023-07-10	13	Roanoke	0	0	0	0	0	0	12	Yes	Attended
17190	e6e526ab-61e4-4d01-a694-bbcf2df78554	Female	2023-06-28	2023-07-11	17	Charlottesville	1	0	0	0	0	0	13	Yes	Attended
19521	b48d5c1b-6f83-4541-93ae-fbc7c0d6c3d3	Female	2023-06-29	2023-07-11	35	Norfolk	1	0	0	0	0	1	12	No	No Show
19790	28689678-32c3-472d-b552-3f0fc6131ec1	Male	2023-06-28	2023-07-12	91	Newport News	0	0	0	0	0	1	14	Yes	Attended

This query identifies the most recent appointments in each neighbourhood based on appointmentday column. The first part of the query is a common table expression (CTE), which groups the data by neighbourhood and then calculates the most recent appointmentday using the MAX() function.

Then the second part joins the Virginia_patients table with a common table expression and filters the data to match only the most recent day per neighbourhood. This helps us identify the appointment scheduling pattern and the latest patient activity in each neighbourhood.

Key Findings:

After our detailed SQL analysis of the Virginia_patients dataset, the following key findings will help in data-driven decisions for future appointments, SMS reminders, and increased attendance among patients, particularly the older ones.

1. SMS reminder programs are helpful and positively influence the attendance rate of patients for their appointments. Such as the total females who received SMS for their appointments was 3465 from the gender column. Similarly, we grouped the patients who showed up for their appointments after receiving SMS reminders. These numbers are 5405, who showed up after receiving SMS reminders. In contrast to the 1478 ones who did not show for their appointment. So these findings suggest that SMS reminders positively influence the appointment attendance rate.
2. Running total analysis of the total appointments shows the growing health care demand in a specific neighbourhood and how we can track this for future decisions on resource allocation.
3. The average age of female and male patients is 49.74 and 50.36, and this can help us to focus more on older patients whose age is above the average age, and we can allocate and prioritize targeted health care.
4. Dense_rank function helped us to identify and rank the neighbourhoods with the highest missed appointments. It also identified the older female patients from the gender column. So we can prioritize our outreach and community help programs.

5. We also found the neighbourhood-level insights, meaning areas with total appointments above 1000, and they need more medical staff and more efficient SMS outreach because of higher demand and more medical appointments.
6. We also run queries on our datasets to find patients with chronic diseases, and it helped us to identify patients with both hypertension and diabetes. So this suggests that chronic disease can impact attendance rate, and we must invest in additional follow-up mechanisms to ensure these patients properly follow their appointment schedules.
7. We also ran queries to find the frequent scheduling of appointments in various neighbourhoods to support better tracking of appointments. Also, we ran a CASE() expression to identify the children, adults, and seniors. By running this expression, we successfully extract the information with values 1074 children, 4910 adults, and 3932 seniors. This information will help us to understand attendance trends in patients based on their age group. Moreover, it will help us to focus more on senior patients for regular monitoring.

Recommendations:

These are the key recommendations from our SQL analysis of the Virginia_patients datasets:

1. Expand SMS reminder programs, especially for neighbourhoods with higher attendance rates and patients (or senior patients) with a higher risk of chronic diseases.
2. There should be minimal to no appointment scheduling on Sundays and Mondays because these are the days with higher rates of no-show from patients of all age groups. Moreover, appointment scheduling should be encouraged on midweekdays because these are days with higher appointment show-ups.
3. Allocate more resources, such as mobile outreach, medical staff, and latest technology, to the neighbourhoods that continuously show higher no-show rates of appointments, such as Charlottesville, Fairfax, Roanoke, Norfolk, and Alexandria.
4. Most importantly, prioritize older patients who have both diabetes and hypertension for regular follow-ups, and more SMS reminders for their appointments to avoid no-shows.
5. Allocate more resources, such as medical staff and expansion of SMS reminders to the neighbourhoods (such as Arlington, Alexandria, Richmond, and Chesapeake) with higher total appointment and show-up rates. It will not only encourage no appointment misses from patients but also make the tending process more effective and feasible.
6. Also, make proper use of running total trends to track daily appointment volume that will help in clinic staffing and resource allocation to neighbourhoods with higher appointment rates.
7. To prevent future missed appointments from senior patients or patients with chronic diseases, use no-show data to understand why they missed their appointments in the first place.

Conclusion:

The SQL-based analysis of the medical appointments of West Virginia State provided us with critical insights into the following factors that influence patients' attendance rate.

1. Patient age and the pattern of show-ups for their medical appointments
2. Role of SMS reminders in follow-ups and showing up for their appointments
3. Health conditions
4. Neighbourhoods

In this analysis report, we were able to identify and rank the neighbourhoods based on their total appointment and the patients' attendance rate. For this purpose, different MYSQL expressions, window functions, aggregations, filtering, and subqueries were used to extract the information and devise the data-driven insights and decisions based on the extracted information.

Based on these analyses, valuable information and insights were extracted to help healthcare professionals for better resource allocation to different neighbourhoods with improvements in communication. Moreover, prioritizing the senior patients with chronic illness for more follow-ups to avoid barriers in keeping up with their appointments. Not only this, but keeping track of daily appointments also offered useful information for the optimization of appointment scheduling.