SQL Mini Project Report – Patient Appointments

Introduction:

The main objective of this project is to explore and analyze patient appointment data to uncover meaningful insights. By using SQL, we want to understand patient behavior like who shows up for appointments, what factors affect no-shows, and how different variables such as age, gender, or appointment day influence outcomes. This can help clinics or hospitals improve scheduling, reduce missed appointments, and provide better healthcare services. The goal is to turn raw data into useful information that supports decision making.

Dataset Overview:

The dataset contains 9,916 rows, with each row representing a single patient appointment. It includes various details about the patient, their health conditions, appointment timings, and whether they showed up or not.

Some of the key columns in this dataset are:

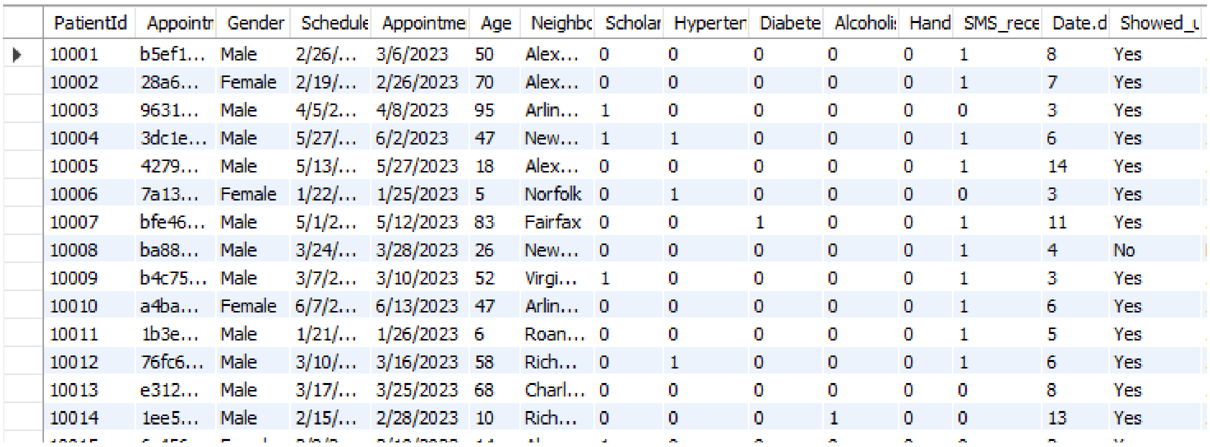
- PatientID and AppointmentID: these both are to identify each patient and their appointment uniquely.
- Gender and Age: helpful in analyzing trends based on gender or age group.
- ScheduledDay and AppointmentDay: show when the appointment was booked and when it actually happened.
- Neighbourhood: tells us where the patient lives, which helps identify location-based trends.
- Hypertension, Diabetes, Alcoholism, Handcap: show the patient's health conditions, which could affect their likelihood of showing up.
- Scholarship: indicates whether the patient is part of a health support program.
- SMS_received: tells us if a reminder was sent, which might influence attendance.
- Showed_up: this is the most important column, showing whether the patient came to their appointment or not.
- Date.diff: shows the gap between when the appointment was scheduled and when it was supposed to happen.

These columns are directly related to the goal of the project: understanding why some patients miss their appointments and what patterns exist in the data.

Methodology and SQL Tasks:

-- 1. Retrieve all columns from the Appointments table.

```
Select * from Appointments;
```

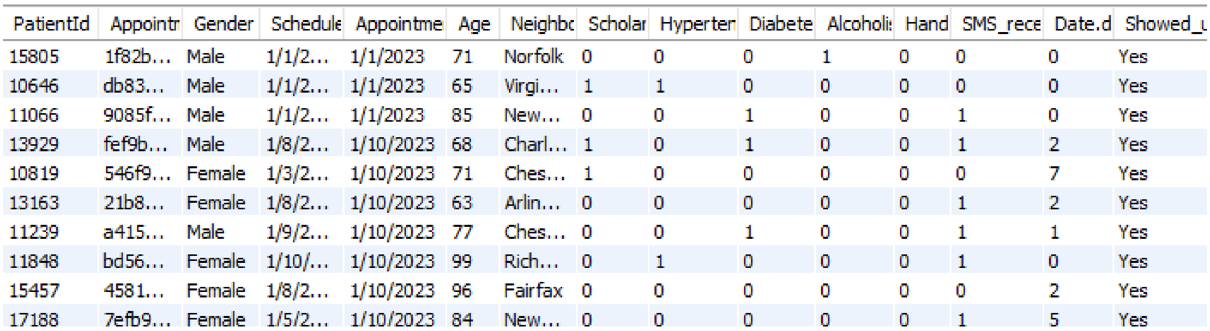


PatientId	Appointr	Gender	Schedule	Appointme	Age	Neighbc	Scholar	Hyperten	Diabete	Alcoholi	Hand	SMS_rece	Date.d	Showed_u
10001	b5ef1...	Male	2/26/...	3/6/2023	50	Alex...	0	0	0	0	0	1	8	Yes
10002	28a6...	Female	2/19/...	2/26/2023	70	Alex...	0	0	0	0	0	1	7	Yes
10003	9631...	Male	4/5/2...	4/8/2023	95	Arlin...	1	0	0	0	0	0	3	Yes
10004	3dc1e...	Male	5/27/...	6/2/2023	47	New...	1	1	0	0	0	1	6	Yes
10005	4279...	Male	5/13/...	5/27/2023	18	Alex...	0	0	0	0	0	1	14	Yes
10006	7a13...	Female	1/22/...	1/25/2023	5	Norfolk	0	1	0	0	0	0	3	Yes
10007	bfe46...	Male	5/1/2...	5/12/2023	83	Fairfax	0	0	1	0	0	1	11	Yes
10008	ba88...	Male	3/24/...	3/28/2023	26	New...	0	0	0	0	0	1	4	No
10009	b4c75...	Male	3/7/2...	3/10/2023	52	Virgi...	1	0	0	0	0	1	3	Yes
10010	a4ba...	Female	6/7/2...	6/13/2023	47	Arlin...	0	0	0	0	0	1	6	Yes
10011	1b3e...	Male	1/21/...	1/26/2023	6	Roan...	0	0	0	0	0	1	5	Yes
10012	76fc6...	Male	3/10/...	3/16/2023	58	Rich...	0	1	0	0	0	1	6	Yes
10013	e312...	Male	3/17/...	3/25/2023	68	Charl...	0	0	0	0	0	0	8	Yes
10014	1ee5...	Male	2/15/...	2/28/2023	10	Rich...	0	0	0	1	0	0	13	Yes

This query shows every detail available for all appointments. It helps us understand the full dataset and check the data before deeper analysis.

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

```
Select * from Appointments
where Age > 60
order by appointmentday
limit 10;
```

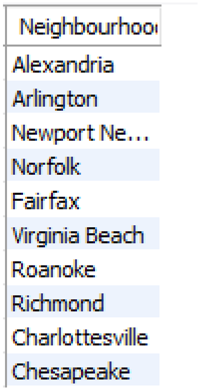


PatientId	Appointr	Gender	Schedule	Appointme	Age	Neighbc	Scholar	Hyperten	Diabete	Alcoholi	Hand	SMS_rece	Date.d	Showed_u
15805	1f82b...	Male	1/1/2...	1/1/2023	71	Norfolk	0	0	0	1	0	0	0	Yes
10646	db83...	Male	1/1/2...	1/1/2023	65	Virgi...	1	1	0	0	0	0	0	Yes
11066	9085f...	Male	1/1/2...	1/1/2023	85	New...	0	0	1	0	0	1	0	Yes
13929	fef9b...	Male	1/8/2...	1/10/2023	68	Charl...	1	0	1	0	0	1	2	Yes
10819	546f9...	Female	1/3/2...	1/10/2023	71	Ches...	1	0	0	0	0	0	7	Yes
13163	21b8...	Female	1/8/2...	1/10/2023	63	Arlin...	0	0	0	0	0	1	2	Yes
11239	a415...	Male	1/9/2...	1/10/2023	77	Ches...	0	0	1	0	0	1	1	Yes
11848	bd56...	Female	1/10/...	1/10/2023	99	Rich...	0	1	0	0	0	1	0	Yes
15457	4581...	Female	1/8/2...	1/10/2023	96	Fairfax	0	0	0	0	0	0	2	Yes
17188	7efb9...	Female	1/5/2...	1/10/2023	84	New...	0	0	0	0	0	1	5	Yes

This shows us some examples of appointments for older patients. It helps focus on senior patients and their appointment patterns.

-- **3.** Show the unique neighborhoods from which patients came.

```
select distinct Neighbourhood
from appointments;
```

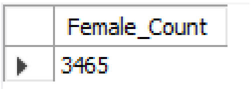


Neighbourhoo
Alexandria
Arlington
Newport Ne...
Norfolk
Fairfax
Virginia Beach
Roanoke
Richmond
Charlottesville
Chesapeake

This gives us a list of all neighborhoods where patients live, helping us know the geographic areas covered by the dataset.

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

```
select count(gender) as Female_Count
from appointments
where gender = 'Female' and SMS_received = 1;
```

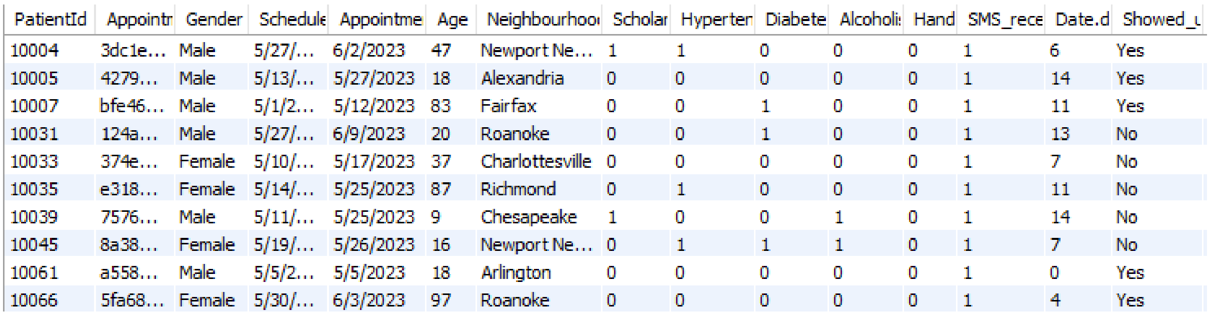


	Female_Count
▶	3465

This counts how many female patients got SMS reminders. It helps check if reminders are being sent properly to women.

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

```
select *
from appointments
where ScheduledDay >= '5/1/2023' and ScheduledDay < '6/1/2023';
```

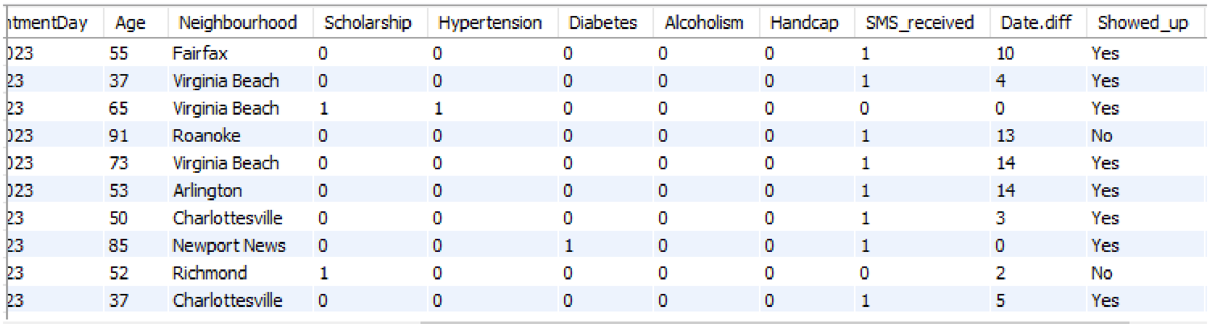


PatientId	Appointr	Gender	Schedule	Appointme	Age	Neighbourhoo	Scholar	Hyperten	Diabete	Alcoholi	Hand	SMS_rece	Date.d	Showed_u
10004	3dc1e...	Male	5/27/...	6/2/2023	47	Newport Ne...	1	1	0	0	0	1	6	Yes
10005	4279...	Male	5/13/...	5/27/2023	18	Alexandria	0	0	0	0	0	1	14	Yes
10007	bfe46...	Male	5/1/2...	5/12/2023	83	Fairfax	0	0	1	0	0	1	11	Yes
10031	124a...	Male	5/27/...	6/9/2023	20	Roanoke	0	0	1	0	0	1	13	No
10033	374e...	Female	5/10/...	5/17/2023	37	Charlottesville	0	0	0	0	0	1	7	No
10035	e318...	Female	5/14/...	5/25/2023	87	Richmond	0	1	0	0	0	1	11	No
10039	7576...	Male	5/11/...	5/25/2023	9	Chesapeake	1	0	0	1	0	1	14	No
10045	8a38...	Female	5/19/...	5/26/2023	16	Newport Ne...	0	1	1	1	0	1	7	No
10061	a558...	Male	5/5/2...	5/5/2023	18	Arlington	0	0	0	0	0	1	0	Yes
10066	5fa68...	Female	5/30/...	6/3/2023	97	Roanoke	0	0	0	0	0	1	4	Yes

This filters appointments in May 2023. It can be useful for analyzing appointment trends in a specific month.

-- **6.** Update the 'Showed_up' status to 'Yes' where it is null or empty

```
SET SQL_SAFE_UPDATES = 0;
Update appointments
set showed_up = 'Yes'
where showed_up is null or showed_up = ''
AND patientid IS NOT NULL;
SET SQL_SAFE_UPDATES = 1;
```



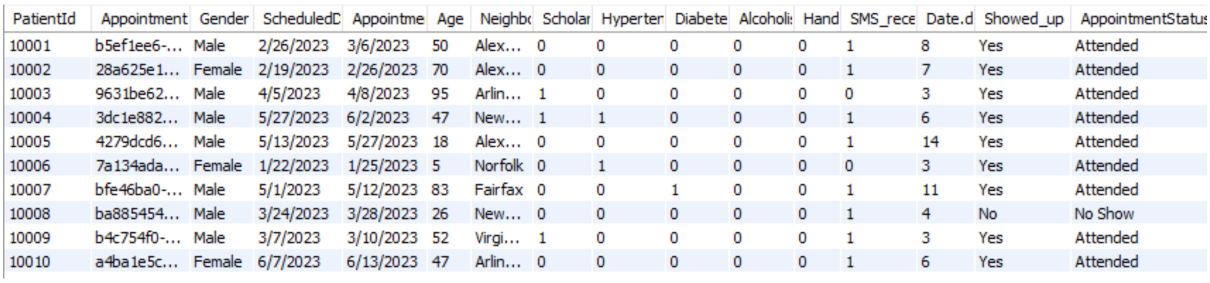
ıtmentDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handcap	SMS_received	Date.diff	Showed_up
)23	55	Fairfax	0	0	0	0	0	1	10	Yes
23	37	Virginia Beach	0	0	0	0	0	1	4	Yes
23	65	Virginia Beach	1	1	0	0	0	0	0	Yes
)23	91	Roanoke	0	0	0	0	0	1	13	No
)23	73	Virginia Beach	0	0	0	0	0	1	14	Yes
)23	53	Arlington	0	0	0	0	0	1	14	Yes
23	50	Charlottesville	0	0	0	0	0	1	3	Yes
23	85	Newport News	0	0	1	0	0	1	0	Yes
23	52	Richmond	1	0	0	0	0	0	2	No
23	37	Charlottesville	0	0	0	0	0	1	5	Yes

This fixes missing attendance data by marking empty or unknown statuses as "Yes," making the data more complete and reliable.

-- **7.** Add a new column AppointmentStatus using a CASE statement:

-- 'No Show' if Showed_up = 'No', 'Attended' otherwise

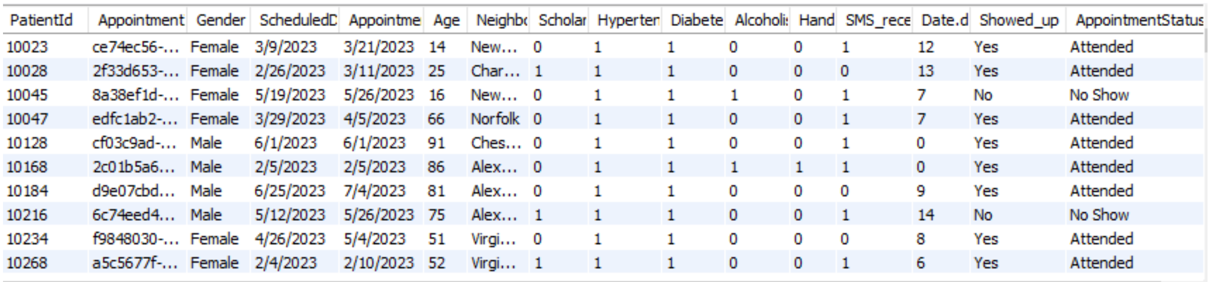
```
ALTER TABLE appointments
ADD COLUMN AppointmentStatus VARCHAR(20);

SET SQL_SAFE_UPDATES = 0;
UPDATE appointments
SET AppointmentStatus =
  CASE
    WHEN Showed_up = 'No' THEN 'No Show'
    ELSE 'Attended'
  END;
SET SQL_SAFE_UPDATES = 1;
```



PatientId	Appointment	Gender	ScheduledD	Appointme	Age	Neighbo	Scholar	Hyperten	Diabete	Alcoholi	Hand	SMS_rece	Date.d	Showed_up	AppointmentStatus
10001	b5ef1ee6-...	Male	2/26/2023	3/6/2023	50	Alex...	0	0	0	0	0	1	8	Yes	Attended
10002	28a625e1...	Female	2/19/2023	2/26/2023	70	Alex...	0	0	0	0	0	1	7	Yes	Attended
10003	9631be62...	Male	4/5/2023	4/8/2023	95	Arlin...	1	0	0	0	0	0	3	Yes	Attended
10004	3dc1e882...	Male	5/27/2023	6/2/2023	47	New...	1	1	0	0	0	1	6	Yes	Attended
10005	4279dcd6...	Male	5/13/2023	5/27/2023	18	Alex...	0	0	0	0	0	1	14	Yes	Attended
10006	7a134ada...	Female	1/22/2023	1/25/2023	5	Norfolk	0	1	0	0	0	0	3	Yes	Attended
10007	bfe46ba0-...	Male	5/1/2023	5/12/2023	83	Fairfax	0	0	1	0	0	1	11	Yes	Attended
10008	ba885454...	Male	3/24/2023	3/28/2023	26	New...	0	0	0	0	0	1	4	No	No Show
10009	b4c754f0-...	Male	3/7/2023	3/10/2023	52	Virgi...	1	0	0	0	0	1	3	Yes	Attended
10010	a4ba1e5c...	Female	6/7/2023	6/13/2023	47	Arlin...	0	0	0	0	0	1	6	Yes	Attended

This creates a new column that clearly labels each appointment as "No Show" or "Attended," simplifying attendance analysis.

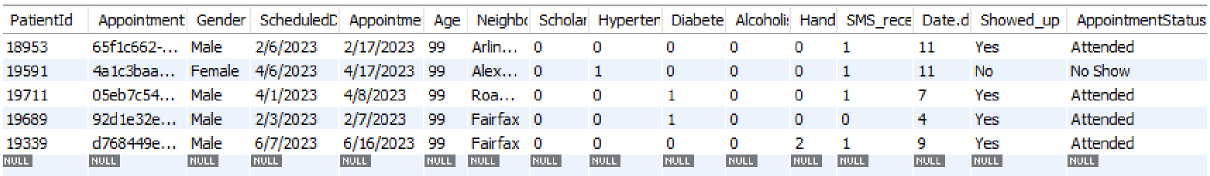
```sql
-- 8. Filter appointments for diabetic patients with hypertension.
SELECT *
FROM appointments
WHERE diabetes = 1 AND hypertension = 1;
```



PatientId	Appointment	Gender	ScheduledD	Appointme	Age	Neighb	Scholar	Hyperter	Diabete	Alcoholi	Hand	SMS_rece	Date.d	Showed_up	AppointmentStatus
10023	ce74ec56-...	Female	3/9/2023	3/21/2023	14	New...	0	1	1	0	0	1	12	Yes	Attended
10028	2f33d653-...	Female	2/26/2023	3/11/2023	25	Char...	1	1	1	0	0	0	13	Yes	Attended
10045	8a38ef1d-...	Female	5/19/2023	5/26/2023	16	New...	0	1	1	1	0	1	7	No	No Show
10047	edfc1ab2-...	Female	3/29/2023	4/5/2023	66	Norfolk	0	1	1	0	0	1	7	Yes	Attended
10128	cf03c9ad-...	Male	6/1/2023	6/1/2023	91	Ches...	0	1	1	0	0	1	0	Yes	Attended
10168	2c01b5a6...	Male	2/5/2023	2/5/2023	86	Alex...	0	1	1	1	1	1	0	Yes	Attended
10184	d9e07cbd...	Male	6/25/2023	7/4/2023	81	Alex...	0	1	1	0	0	0	9	Yes	Attended
10216	6c74eed4...	Male	5/12/2023	5/26/2023	75	Alex...	1	1	1	0	0	1	14	No	No Show
10234	f9848030-...	Female	4/26/2023	5/4/2023	51	Virgi...	0	1	1	0	0	0	8	Yes	Attended
10268	a5c5677f-...	Female	2/4/2023	2/10/2023	52	Virgi...	1	1	1	0	0	1	6	Yes	Attended

This selects patients who have both diabetes and hypertension, helping to study appointment behavior of patients with these conditions.

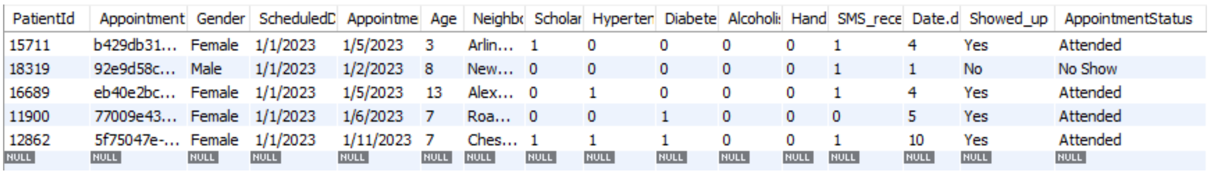
```sql
-- 9. Order the records by Age in descending order and show only the top 5 oldest –
-- patients.
SELECT *
FROM appointments
ORDER BY Age DESC
LIMIT 5;
```



PatientId	Appointment	Gender	ScheduledD	Appointme	Age	Neighb	Scholar	Hyperter	Diabete	Alcoholi	Hand	SMS_rece	Date.d	Showed_up	AppointmentStatus
18953	65f1c662-...	Male	2/6/2023	2/17/2023	99	Arlin...	0	0	0	0	0	1	11	Yes	Attended
19591	4a1c3baa...	Female	4/6/2023	4/17/2023	99	Alex...	0	1	0	0	0	1	11	No	No Show
19711	05eb7c54...	Male	4/1/2023	4/8/2023	99	Roa...	0	0	1	0	0	1	7	Yes	Attended
19689	92d1e32e...	Male	2/3/2023	2/7/2023	99	Fairfax	0	0	1	0	0	0	4	Yes	Attended
19339	d768449e...	Male	6/7/2023	6/16/2023	99	Fairfax	0	0	0	0	2	1	9	Yes	Attended
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

This identifies the oldest five patients, useful for focusing on the elderly population in appointment data.

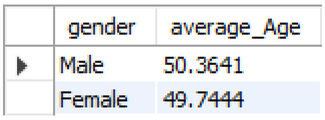
```sql
-- 10. Limit results to the first 5 appointments for patients under age 18.
SELECT *
FROM appointments
WHERE Age < 18
ORDER BY ScheduledDay
LIMIT 5;
```



PatientId	Appointment	Gender	ScheduledD	Appointme	Age	Neighb	Scholar	Hyperter	Diabete	Alcoholi	Hand	SMS_rece	Date.d	Showed_up	AppointmentStatus
15711	b429db31...	Female	1/1/2023	1/5/2023	3	Arlin...	1	0	0	0	0	1	4	Yes	Attended
18319	92e9d58c...	Male	1/1/2023	1/2/2023	8	New...	0	0	0	0	0	1	1	No	No Show
16689	eb40e2bc...	Female	1/1/2023	1/5/2023	13	Alex...	0	1	0	0	0	1	4	Yes	Attended
11900	77009e43...	Female	1/1/2023	1/6/2023	7	Roa...	0	0	1	0	0	0	5	Yes	Attended
12862	5f75047e-...	Female	1/1/2023	1/11/2023	7	Ches...	1	1	1	0	0	1	10	Yes	Attended
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

This shows a few examples of appointments for children and teenagers, helping analyze younger patients’ appointment habits.

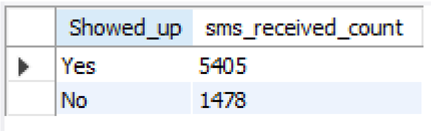
```
-- 11. Find the average age of patients for each gender.
Select gender, AVG(age) AS average_Age
FROM appointments
GROUP BY gender;
```



gender	average_Age
Male	50.3641
Female	49.7444

This tells us the typical age for male and female patients, helping compare age distributions between genders.

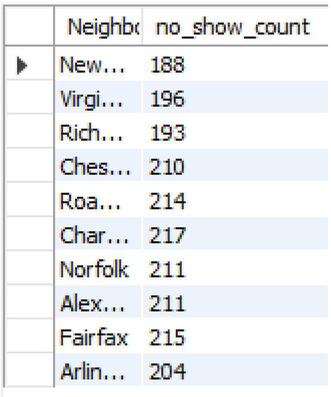
```
-- 12. Count how many patients received SMS reminders, grouped by Showed_up
-- status.
SELECT Showed_up, COUNT(SMS_received) AS sms_received_count
FROM appointments
WHERE SMS_received = 1
GROUP BY Showed_up;
```



Showed_up	sms_received_count
Yes	5405
No	1478

This counts how many people who got SMS reminders showed up or missed their appointment, helping measure reminder effectiveness.

```
-- 13. Count no-show appointments in each neighborhood using GROUP BY.
SELECT Neighbourhood, COUNT(*) AS no_show_count
FROM appointments
WHERE Showed_up = 'No'
GROUP BY Neighbourhood;
```

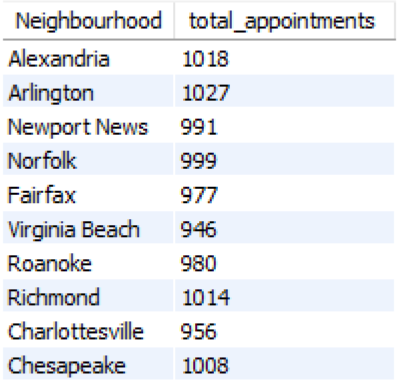


Neighbc	no_show_count
New...	188
Virgi...	196
Rich...	193
Ches...	210
Roa...	214
Char...	217
Norfolk	211
Alex...	211
Fairfax	215
Arlin...	204

This shows which neighborhoods have the most missed appointments, helping identify areas where patients tend to skip visits.

-- **14.** Show neighborhoods with more than 100 total appointments (HAVING clause).

```
SELECT Neighbourhood, COUNT(*) AS total_appointments
FROM appointments
GROUP BY Neighbourhood
HAVING COUNT(*) > 100;
```

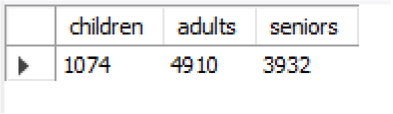


Neighbourhood	total_appointments
Alexandria	1018
Arlington	1027
Newport News	991
Norfolk	999
Fairfax	977
Virginia Beach	946
Roanoke	980
Richmond	1014
Charlottesville	956
Chesapeake	1008

This filters for neighborhoods with a large number of appointments, focusing on busy areas for deeper analysis.

-- **15.** Use CASE to calculate the total number of: children (Age < 12)

```
-- adults (Age BETWEEN 12 AND 60), seniors (Age > 60)
SELECT
  SUM(CASE WHEN Age < 12 THEN 1 ELSE 0 END) AS children,
  SUM(CASE WHEN Age BETWEEN 12 AND 60 THEN 1 ELSE 0 END) AS adults,
  SUM(CASE WHEN Age > 60 THEN 1 ELSE 0 END) AS seniors
FROM appointments;
```

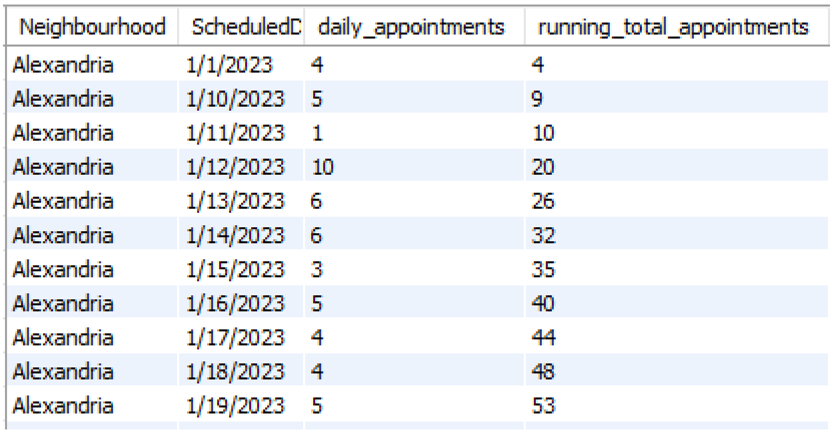


	children	adults	seniors
▶	1074	4910	3932

This counts how many patients fall into different age groups, helping us understand the age makeup of the patient population.

-- 16. Tracks 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?

```
SELECT
 Neighbourhood,
 ScheduledDay,
 COUNT(*) AS daily_appointments,
 SUM(COUNT(*)) OVER (
  PARTITION BY Neighbourhood
  ORDER BY ScheduledDay
  ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
 ) AS running_total_appointments
FROM appointments
GROUP BY Neighbourhood, ScheduledDay
ORDER BY Neighbourhood, ScheduledDay;
```

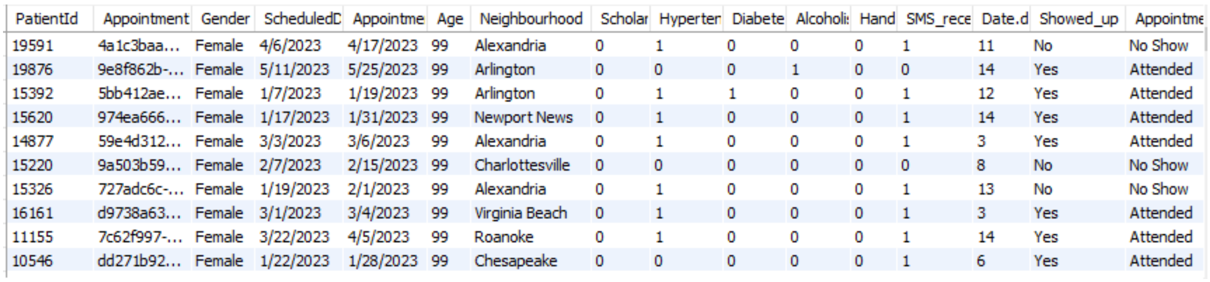


Neighbourhood	ScheduledD	daily_appointments	running_total_appointments
Alexandria	1/1/2023	4	4
Alexandria	1/10/2023	5	9
Alexandria	1/11/2023	1	10
Alexandria	1/12/2023	10	20
Alexandria	1/13/2023	6	26
Alexandria	1/14/2023	6	32
Alexandria	1/15/2023	3	35
Alexandria	1/16/2023	5	40
Alexandria	1/17/2023	4	44
Alexandria	1/18/2023	4	48
Alexandria	1/19/2023	5	53

This shows how the number of appointments adds up day by day in each neighborhood, helping spot appointment trends over time.

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

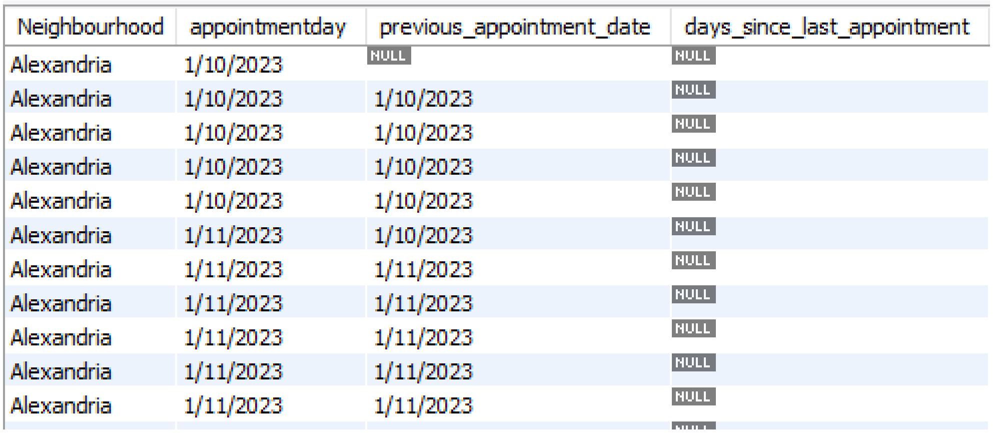
```
SELECT  *,
 DENSE_RANK() OVER (
  PARTITION BY gender
  ORDER BY age DESC) AS age_rank_within_gender
FROM appointments;
```



PatientId	Appointment	Gender	ScheduledD	Appointme	Age	Neighbourhood	Scholar	Hyperten	Diabete	Alcoholi	Hand	SMS_rece	Date.d	Showed_up	Appointme
19591	4a1c3baa...	Female	4/6/2023	4/17/2023	99	Alexandria	0	1	0	0	0	1	11	No	No Show
19876	9e8f862b-...	Female	5/11/2023	5/25/2023	99	Arlington	0	0	0	1	0	0	14	Yes	Attended
15392	5bb412ae...	Female	1/7/2023	1/19/2023	99	Arlington	0	1	1	0	0	1	12	Yes	Attended
15620	974ea666...	Female	1/17/2023	1/31/2023	99	Newport News	0	1	0	0	0	1	14	Yes	Attended
14877	59e4d312...	Female	3/3/2023	3/6/2023	99	Alexandria	0	1	0	0	0	1	3	Yes	Attended
15220	9a503b59...	Female	2/7/2023	2/15/2023	99	Charlottesville	0	0	0	0	0	0	8	No	No Show
15326	727adc6c-...	Female	1/19/2023	2/1/2023	99	Alexandria	0	1	0	0	0	1	13	No	No Show
16161	d9738a63...	Female	3/1/2023	3/4/2023	99	Virginia Beach	0	1	0	0	0	1	3	Yes	Attended
11155	7c62f997-...	Female	3/22/2023	4/5/2023	99	Roanoke	0	1	0	0	0	1	14	Yes	Attended
10546	dd271b92...	Female	1/22/2023	1/28/2023	99	Chesapeake	0	0	0	0	0	1	6	Yes	Attended

This ranks patients by age within males and females separately, highlighting the oldest patients in each gender group.

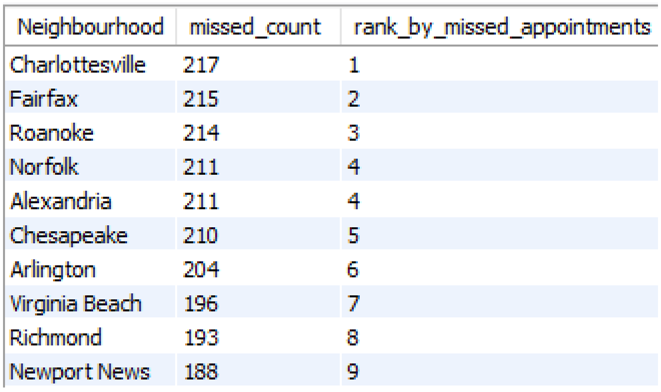
```
-- 18. How many days have passed since the last appointment in the same
-- neighborhood? (Hint: DATEDIFF and Lag)  (This helps to see how frequently
-- appointments are happening in each neighborhood.)
SELECT
  Neighbourhood,
  appointmentday,
  LAG(appointmentday) OVER (
    PARTITION BY Neighbourhood
    ORDER BY appointmentday ) AS previous_appointment_date,
  DATEDIFF(
    appointmentday,
    LAG(appointmentday) OVER (
      PARTITION BY Neighbourhood
      ORDER BY appointmentday)) AS days_since_last_appointment
FROM appointments
ORDER BY Neighbourhood, appointmentday;
```



Neighbourhood	appointmentday	previous_appointment_date	days_since_last_appointment
Alexandria	1/10/2023	NULL	NULL
Alexandria	1/10/2023	1/10/2023	NULL
Alexandria	1/10/2023	1/10/2023	NULL
Alexandria	1/10/2023	1/10/2023	NULL
Alexandria	1/10/2023	1/10/2023	NULL
Alexandria	1/11/2023	1/10/2023	NULL
Alexandria	1/11/2023	1/11/2023	NULL
Alexandria	1/11/2023	1/11/2023	NULL
Alexandria	1/11/2023	1/11/2023	NULL
Alexandria	1/11/2023	1/11/2023	NULL
Alexandria	1/11/2023	1/11/2023	NULL

This measures the gap between appointments in each neighborhood, showing how often patients in an area come for appointments.

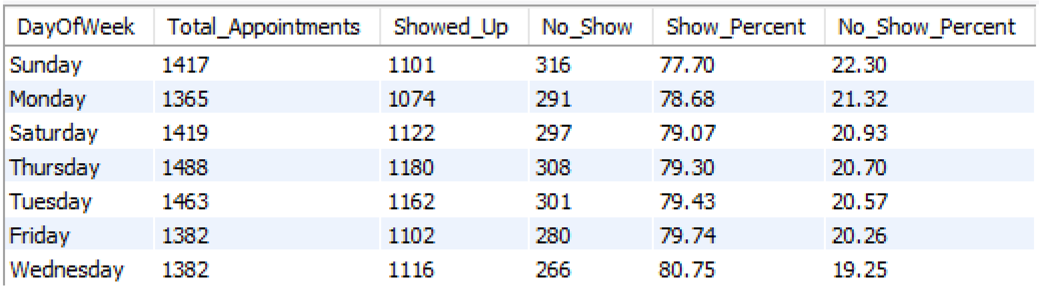
```
-- 19. Which neighborhoods have the highest number of missed appointments? Use
-- DENSE_RANK() to rank neighborhoods based on the number of no-show
-- appointments.
WITH MissedAppointments AS (
  SELECT
    Neighbourhood,
    COUNT(*) AS missed_count
  FROM appointments
  WHERE Showed_up = 'No'
  GROUP BY Neighbourhood)

SELECT
  Neighbourhood,
  missed_count,
  DENSE_RANK() OVER (
    ORDER BY missed_count DESC
  ) AS rank_by_missed_appointments
FROM MissedAppointments
ORDER BY rank_by_missed_appointments;
```



Neighbourhood	missed_count	rank_by_missed_appointments
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 ranks neighborhoods by how many appointments were missed, helping focus on locations with the biggest attendance problems.

```
-- 20.  Are patients more likely to miss appointments on certain days of the week?
-- • (Use the AppointmentDay column in 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 round
-- function to  show upto two decimal points
-- • Sort the result by No_Show_Percent in descending order to see the worst performing
-- days first.
SELECT
  DAYNAME(STR_TO_DATE(AppointmentDay, '%m/%d/%Y')) AS DayOfWeek,
  COUNT(*) AS Total_Appointments,
  SUM(CASE WHEN Showed_up = 'Yes' THEN 1 ELSE 0 END) AS Showed_Up,
  SUM(CASE WHEN Showed_up = 'No' THEN 1 ELSE 0 END) AS No_Show,
  ROUND(SUM(CASE WHEN Showed_up = 'Yes' THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS
Show_Percent,
  ROUND(SUM(CASE WHEN Showed_up = 'No' THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS
No_Show_Percent
FROM appointments
GROUP BY DayOfWeek
ORDER BY No_Show_Percent DESC;
```



DayOfWeek	Total_Appointments	Showed_Up	No_Show	Show_Percent	No_Show_Percent
Sunday	1417	1101	316	77.70	22.30
Monday	1365	1074	291	78.68	21.32
Saturday	1419	1122	297	79.07	20.93
Thursday	1488	1180	308	79.30	20.70
Tuesday	1463	1162	301	79.43	20.57
Friday	1382	1102	280	79.74	20.26
Wednesday	1382	1116	266	80.75	19.25

This analysis shows which weekdays have more missed appointments by calculating and comparing no-show percentages for each day. It helps identify days with the worst attendance.

Key Findings:

- Older adults, children, and seniors form a big part of the patient group, so clinics should plan services based on age-related needs and prioritize senior care.
- Female patients are slightly more active and are more likely to respond to SMS reminders, but reminders alone don't guarantee attendance.
- The neighborhood with the highest number of missed appointments is Charlottesville, followed closely by Fairfax and Roanoke. These areas show a pattern of more no-shows compared to others, suggesting that focused attention or reminders may be needed in these locations to improve attendance.
- Appointments are increasing over time in several areas, and some neighborhoods see patients returning more regularly, showing growing healthcare demand and frequency. Returning max are coming from Arlington.
- Wednesday has the best attendance rate with the lowest no-shows (only 19.25%), while Sunday has the highest no-show rate (22.30%). This means patients are more likely to miss appointments on Sundays and Mondays. Clinics could use this insight to schedule fewer or non-critical appointments on high no-show days and focus reminders more on those days.
- Most of the patients who had appointments were adults (aged 12 to 60), followed by seniors (aged above 60), and the least were children (under 12). This shows that adults form the largest group visiting for appointments, while children make up the smallest portion.
- Cleaning up the data and adding new status labels like "Attended" or "No Show" helped make analysis clearer and easier to understand.
- Using rankings and averages showed differences in attendance behavior across genders, age groups, and neighborhoods, helping clinics focus their efforts where needed most.

Recommendations:

- Send timely SMS reminders: Patients who received SMS reminders showed better attendance. Continue using SMS reminders and ensure they are sent a day or two before the appointment.
- Focus on weekends and Mondays: No-shows were slightly higher on Sundays and Mondays. Consider sending extra reminders for appointments scheduled on these days or scheduling fewer appointments when possible.
- Target high no-show neighborhoods: Areas like Charlottesville, Fairfax, and Roanoke had the most missed appointments. More follow-ups, local awareness, or education in these regions could help reduce no-shows.
- Special care for seniors and children: Seniors (age > 60) had a large number of appointments. Offering transport support or follow-up calls may help them attend regularly. For children, make sure guardians understand the importance of attending.
- Monitor chronic illness patients: Patients with both diabetes and hypertension were part of the no-show group. Personalized care or check-ins may encourage them to attend.
- Use historical trends: Analyze days with low attendance historically and schedule fewer appointments or more flexible slots on those days.
- Simplify rescheduling: Make it easier for patients to cancel or reschedule if they can't attend, rather than just not showing up.

Conclusion:

The project successfully identified key patterns and factors contributing to patient no-shows using appointment data. By analyzing demographics, days of the week, neighborhood trends, and reminder effectiveness, we gained valuable insights to inform strategies for reducing missed appointments.

- SQL played a crucial role in achieving this by helping:
- Extract and clean relevant data from large tables efficiently
- Group and compare appointment behavior by day, age group, and neighborhood
- Calculate no-show rates and spot trends across different patient segments
- Rank locations based on missed appointments to target interventions
- Evaluate the impact of SMS reminders on attendance

Overall, SQL enabled fast and accurate analysis, making it possible to turn raw data into clear, actionable insights that can support better healthcare scheduling and reduce no-show rates.