# SQL PROJECT REPORT

1. **Introduction**. This report provides an analysis of patient appointment data stored in the 'virginia_patient' table. The aim is to explore patient demographics, appointment behaviors, and patterns of attendance or no-shows. Using SQL, various data retrieval, filtering, aggregation, and window functions were applied to extract insights that can support healthcare management decisions such as improving appointment attendance and optimizing resource allocation.

2. **Dataset Description**. The dataset, named 'virginia_patient', contains detailed records of patient appointments including demographic information, appointment dates, attendance status, and health conditions. The dataset contains 9916 records spanning multiple Neighborhoods and appointment dates. Key attributes include:

   a. **PatientId**: Unique patient identifier.

   b. **Age**: Patient age.

   c. **Gender**: Male/ Female.

   d. **Neighbourhood**: Patient's residential Neighbourhood.

   e. **ScheduledDay**: Date when appointment was scheduled.

   f. **AppointmentDay**: Date of appointment.

   g. **Showed_up**: Whether the patient attended the appointment.

   h. **SMS_received**: Indicator if an SMS reminder was sent.

   i. **Diabetes, Hypertension**: Patient health conditions (1 = Yes, 0 = No).

   j. **AppointmentStatus**: Computed column showing if appointment was 'No Show' or 'Attended'.

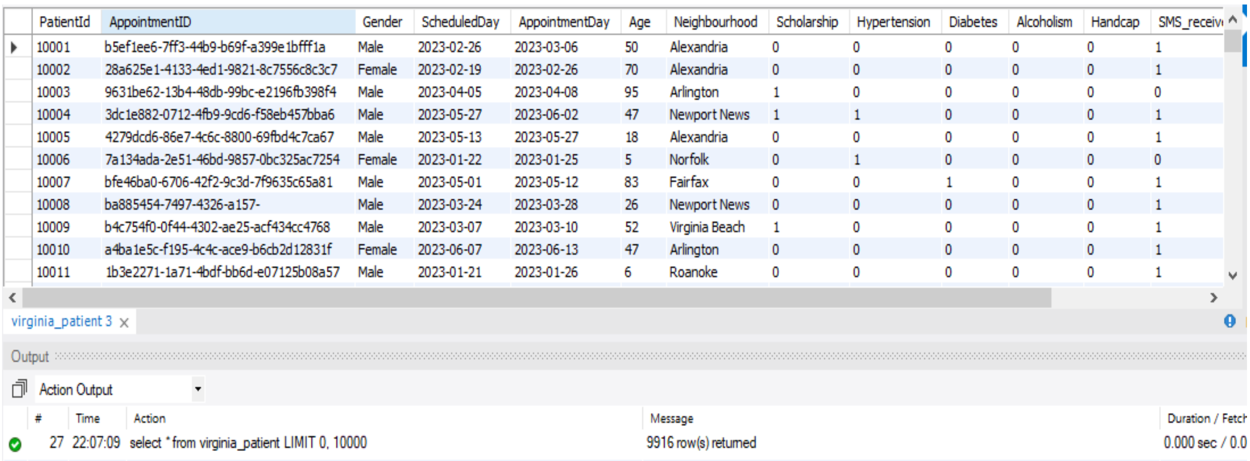
3. **Methodology and SQL Tasks**. The analysis was performed using standard SQL queries on the 'virginia_patient' table. The tasks ranged from simple data retrieval to advanced aggregation and window functions. Below are details of key questions, corresponding SQL queries, screenshot of the results and summarized analysis.

## SECTION-1: BASIC SQL & DATA RETRIEVAL

### Q1: Retrieve all columns from the Appointments table.

**SQL Query**:

```
SELECT *
FROM virginia_patient;
```



| PatientId | AppointmentID | Gender | ScheduledDay | AppointmentDay | Age | Neighbourhood | Scholarship | Hypertension | Diabetes | Alcoholism | Handcap | SMS_receiv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10001 | b5ef1ee6-7ff3-44b9-b69f-a399e1bfff1a | Male | 2023-02-26 | 2023-03-06 | 50 | Alexandria | 0 | 0 | 0 | 0 | 0 | 1 |
| 10002 | 28a625e1-4133-4ed1-9821-8c7556c8c3c7 | Female | 2023-02-19 | 2023-02-26 | 70 | Alexandria | 0 | 0 | 0 | 0 | 0 | 1 |
| 10003 | 9631be62-13b4-48db-99bc-e2196fb398f4 | Male | 2023-04-05 | 2023-04-08 | 95 | Arlington | 1 | 0 | 0 | 0 | 0 | 0 |
| 10004 | 3dc1e882-0712-4fb9-9cd6-f58eb457bba6 | Male | 2023-05-27 | 2023-06-02 | 47 | Newport News | 1 | 1 | 0 | 0 | 0 | 1 |
| 10005 | 4279dcd6-86e7-4c6c-8800-69fbd4c7ca67 | Male | 2023-05-13 | 2023-05-27 | 18 | Alexandria | 0 | 0 | 0 | 0 | 0 | 1 |
| 10006 | 7a134ada-2e51-46bd-9857-0bc325ac7254 | Female | 2023-01-22 | 2023-01-25 | 5 | Norfolk | 0 | 1 | 0 | 0 | 0 | 0 |
| 10007 | bfe46ba0-6706-42f2-9c3d-7f9635c65a81 | Male | 2023-05-01 | 2023-05-12 | 83 | Fairfax | 0 | 0 | 1 | 0 | 0 | 1 |
| 10008 | ba885454-7497-4326-a157- | Male | 2023-03-24 | 2023-03-28 | 26 | Newport News | 0 | 0 | 0 | 0 | 0 | 1 |
| 10009 | b4c754f0-0f44-4302-ae25-acf434cc4768 | Male | 2023-03-07 | 2023-03-10 | 52 | Virginia Beach | 1 | 0 | 0 | 0 | 0 | 1 |
| 10010 | a4ba1e5c-f195-4c4c-ace9-b6cb2d12831f | Female | 2023-06-07 | 2023-06-13 | 47 | Arlington | 0 | 0 | 0 | 0 | 0 | 1 |
| 10011 | 1b3e2271-1a71-4bdf-bb6d-e07125b08a57 | Male | 2023-01-21 | 2023-01-26 | 6 | Roanoke | 0 | 0 | 0 | 0 | 0 | 1 |

| # | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| 27 | 22:07:09 | select * from virginia_patient LIMIT 0, 10000 | 9916 row(s) returned | 0.000 sec / 0.0 |

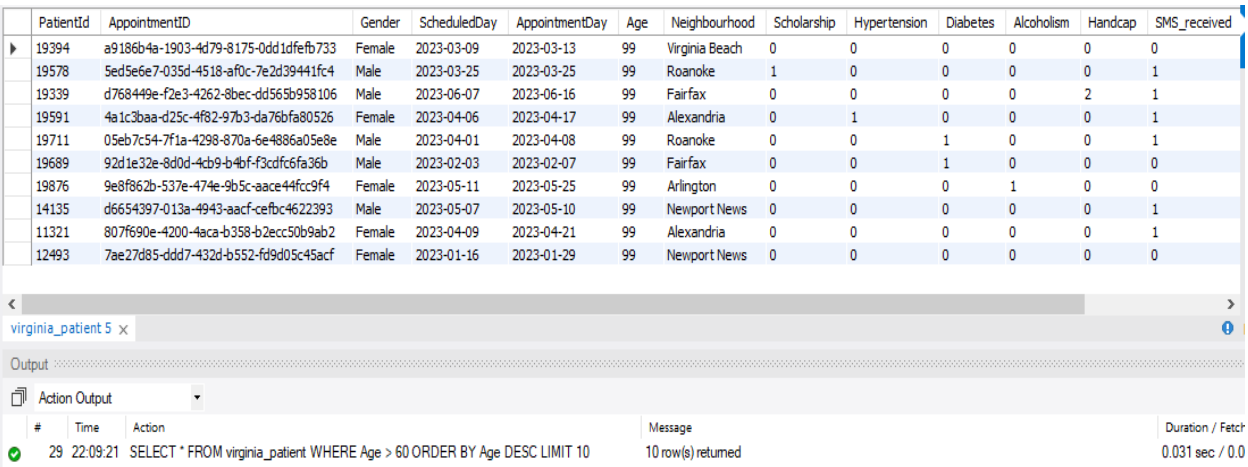
**Summary**:

This query retrieves every column and row from the table 'virginia_patient', showing all the data stored in it.

### Q2: List the first 10 appointments where the patient is older than 60.

**SQL Query**:

```
SELECT *
FROM virginia_patient
WHERE Age > 60
ORDER BY Age DESC
LIMIT 10;
```



|  | PatientId | AppointmentID | Gender | ScheduledDay | AppointmentDay | Age | Neighbourhood | Scholarship | Hypertension | Diabetes | Alcoholism | Handcap | SMS_received |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ▶ | 19394 | a9186b4a-1903-4d79-8175-0dd1dfefb733 | Female | 2023-03-09 | 2023-03-13 | 99 | Virginia Beach | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 19578 | 5ed5e6e7-035d-4518-af0c-7e2d39441fc4 | Male | 2023-03-25 | 2023-03-25 | 99 | Roanoke | 1 | 0 | 0 | 0 | 0 | 1 |
|  | 19339 | d768449e-f2e3-4262-8bec-dd565b958106 | Male | 2023-06-07 | 2023-06-16 | 99 | Fairfax | 0 | 0 | 0 | 0 | 2 | 1 |
|  | 19591 | 4a1c3baa-d25c-4f82-97b3-da76bfa80526 | Female | 2023-04-06 | 2023-04-17 | 99 | Alexandria | 0 | 1 | 0 | 0 | 0 | 1 |
|  | 19711 | 05eb7c54-7f1a-4298-870a-6e4886a05e8e | Male | 2023-04-01 | 2023-04-08 | 99 | Roanoke | 0 | 0 | 1 | 0 | 0 | 1 |
|  | 19689 | 92d1e32e-8d0d-4cb9-b4bf-f3cdfc6fa36b | Male | 2023-02-03 | 2023-02-07 | 99 | Fairfax | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 19876 | 9e8f862b-537e-474e-9b5c-aace44fcc9f4 | Female | 2023-05-11 | 2023-05-25 | 99 | Arlington | 0 | 0 | 0 | 1 | 0 | 0 |
|  | 14135 | d6654397-013a-4943-aacf-cefbc4622393 | Male | 2023-05-07 | 2023-05-10 | 99 | Newport News | 0 | 0 | 0 | 0 | 0 | 1 |
|  | 11321 | 807f690e-4200-4aca-b358-b2ecc50b9ab2 | Female | 2023-04-09 | 2023-04-21 | 99 | Alexandria | 0 | 0 | 0 | 0 | 0 | 1 |
|  | 12493 | 7ae27d85-ddd7-432d-b552-fd9d05c45acf | Female | 2023-01-16 | 2023-01-29 | 99 | Newport News | 0 | 0 | 0 | 0 | 0 | 0 |

|  | # | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|---|
| ✓ | 29 | 22:09:21 | SELECT * FROM virginia_patient WHERE Age > 60 ORDER BY Age DESC LIMIT 10 | 10 row(s) returned | 0.031 sec / 0.0 |

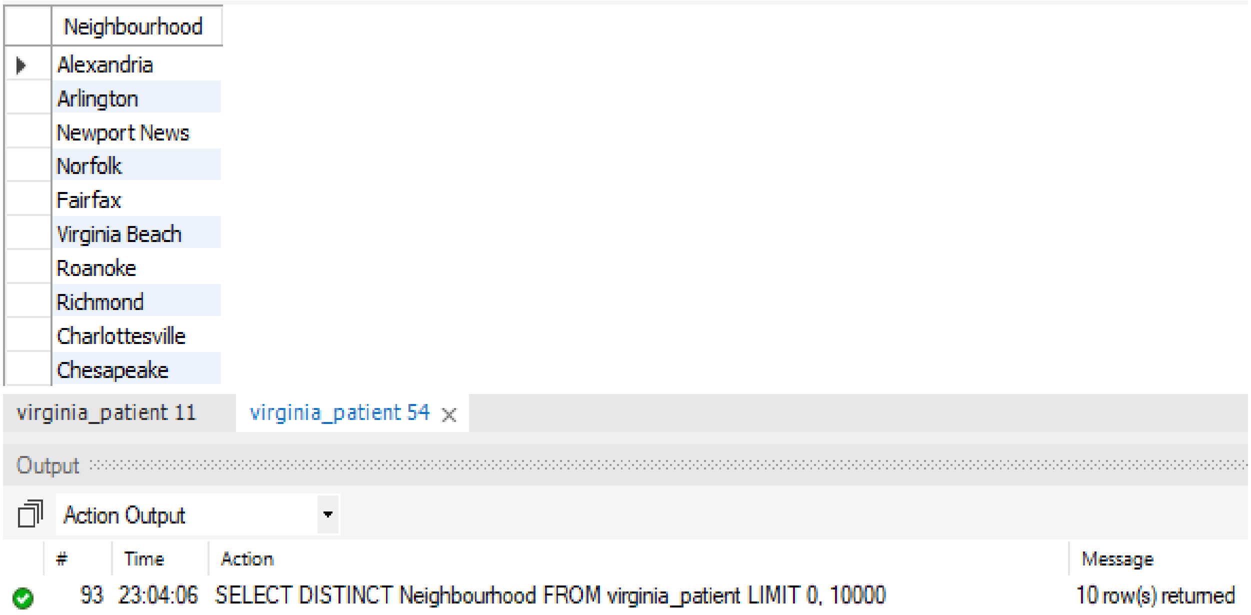
**Summary**:

This query gets the first 10 records of patients older than 60, showing the oldest patients first.

### Q3: Show the unique Neighbourhoods from which patients came.

**SQL Query**:

```
SELECT DISTINCT Neighbourhood
FROM virginia_patient;
```



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

|  | # | Time | Action | Message |
|---|---|---|---|---|
| ✓ | 93 | 23:04:06 | SELECT DISTINCT Neighbourhood FROM virginia_patient LIMIT 0, 10000 | 10 row(s) returned |

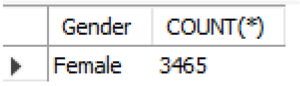
**Summary**:

This query retrieves all unique neighbourhoods i.e. 10 from the 'virginia_patient' table, showing each neighbourhood only once.

### Q4: Find all female patients who received an SMS reminder. Give count of them.

**SQL Query**:

```
SELECT Gender, COUNT(*)
FROM virginia_patient
WHERE gender = "Female" AND SMS_received = "1";
```



| Gender | COUNT(*) |
|---|---|
| Female | 3465 |

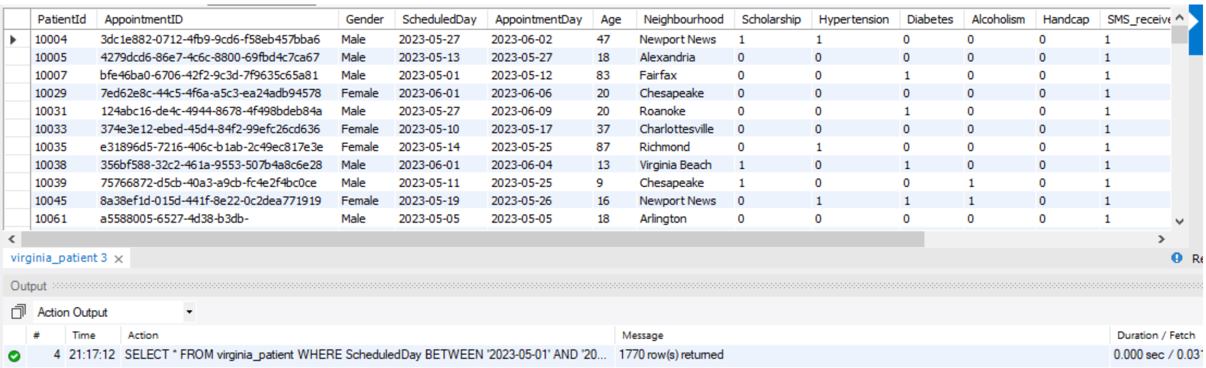
**Summary**:

This query counts how many female patients received an SMS reminder. It returns the gender and the total count of those patients i.e. 3465.

### Q5: Display all appointments scheduled on or after '2023-05-01' and before '2023-06-01'.

**SQL Query**:

```
SELECT *
FROM virginia_patient
WHERE ScheduledDay BETWEEN '2023-05-01' AND '2023-06-01';
```



| PatientId | AppointmentID | Gender | ScheduledDay | AppointmentDay | Age | Neighbourhood | Scholarship | Hypertension | Diabetes | Alcoholism | Handcap | SMS_receive |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10004 | 3dc1e882-0712-4fb9-9cd6-f58eb457bba6 | Male | 2023-05-27 | 2023-06-02 | 47 | Newport News | 1 | 1 | 0 | 0 | 0 | 1 |
| 10005 | 4279dcd6-86e7-4c6c-8800-69fbd4c7ca67 | Male | 2023-05-13 | 2023-05-27 | 18 | Alexandria | 0 | 0 | 0 | 0 | 0 | 1 |
| 10007 | bfe46ba0-6706-42f2-9c3d-7f9635c65a81 | Male | 2023-05-01 | 2023-05-12 | 83 | Fairfax | 0 | 0 | 1 | 0 | 0 | 1 |
| 10029 | 7ed62e8c-44c5-4f6a-a5c3-ea24adb94578 | Female | 2023-06-01 | 2023-06-06 | 20 | Chesapeake | 0 | 0 | 0 | 0 | 0 | 1 |
| 10031 | 124abc16-de4c-4944-8678-4f498bdeb84a | Male | 2023-05-27 | 2023-06-09 | 20 | Roanoke | 0 | 0 | 1 | 0 | 0 | 1 |
| 10033 | 374e3e12-ebed-45d4-84f2-99efc26cd636 | Female | 2023-05-10 | 2023-05-17 | 37 | Charlottesville | 0 | 0 | 0 | 0 | 0 | 1 |
| 10035 | e31896d5-7216-406c-b1ab-2c49ec817e3e | Female | 2023-05-14 | 2023-05-25 | 87 | Richmond | 0 | 1 | 0 | 0 | 0 | 1 |
| 10038 | 356bf588-32c2-461a-9553-507b4a8c6e28 | Male | 2023-06-01 | 2023-06-04 | 13 | Virginia Beach | 1 | 0 | 1 | 0 | 0 | 1 |
| 10039 | 75766872-d5cb-40a3-a9cb-fc4e2f4bc0ce | Male | 2023-05-11 | 2023-05-25 | 9 | Chesapeake | 1 | 0 | 0 | 1 | 0 | 1 |
| 10045 | 8a38ef1d-015d-441f-8e22-0c2dea771919 | Female | 2023-05-19 | 2023-05-26 | 16 | Newport News | 0 | 1 | 1 | 1 | 0 | 1 |
| 10061 | a5588005-6527-4d38-b3db- | Male | 2023-05-05 | 2023-05-05 | 18 | Arlington | 0 | 0 | 0 | 0 | 0 | 1 |

```
#  Time      Action                                                                    Message               Duration / Fetch
4  21:17:12  SELECT * FROM virginia_patient WHERE ScheduledDay BETWEEN '2023-05-01' AND '20...  1770 row(s) returned  0.000 sec / 0.03
```

**Summary**:

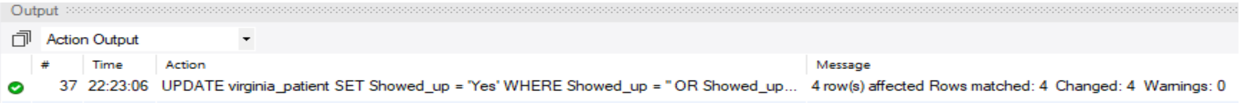
This query retrieves 1770 appointments that are scheduled between May 1, 2023, and June 1, 2023, inclusive of both dates.

## SECTION-2: DATA MODIFICATION & FILTERING

### Q6: Update the 'Showed_up' status to 'Yes' where it is null or empty.

**SQL Query**:

```
UPDATE virginia_patient
SET Showed_up = 'Yes'
WHERE Showed_up = '' OR Showed_up IS NULL;
```



```
#   Time      Action                                                           Message
37  22:23:06  UPDATE virginia_patient SET Showed_up = 'Yes' WHERE Showed_up = '' OR Showed_up...  4 row(s) affected Rows matched: 4  Changed: 4  Warnings: 0
```

**Summary**:

This query updates the Showed_up column by setting it to 'Yes' wherever the value is either empty or null, fixing 4 missing entries.

### Q7: Add a new column AppointmentStatus using a CASE statement:
### 'No Show' if Showed_up = 'No'
### 'Attended' otherwise.

**SQL Query**:

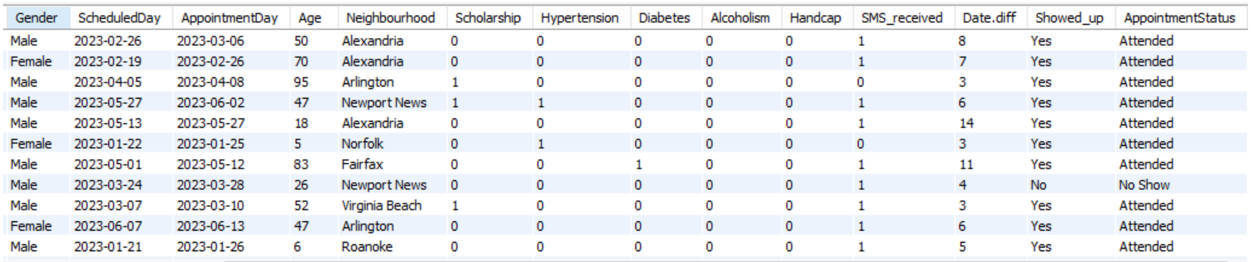
**-- Option: 1**

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

```
UPDATE virginia_patient
SET AppointmentStatus = CASE WHEN Showed_up = 'No' THEN 'No Show'
                                                       ELSE 'Attended'
                                                        END;
```

-- **Option: 2**

```
ALTER TABLE virginia_patient
ADD COLUMN AppointmentStatus  VARCHAR (20) AS (CASE WHEN Showed_up = 'No' THEN 'No Show'
                                                       ELSE 'Attended'
                                                       END);
```



| Gender | ScheduledDay | AppointmentDay | Age | Neighbourhood | Scholarship | Hypertension | Diabetes | Alcoholism | Handcap | SMS_received | Date.diff | Showed_up | AppointmentStatus |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Male | 2023-02-26 | 2023-03-06 | 50 | Alexandria | 0 | 0 | 0 | 0 | 0 | 1 | 8 | Yes | Attended |
| Female | 2023-02-19 | 2023-02-26 | 70 | Alexandria | 0 | 0 | 0 | 0 | 0 | 1 | 7 | Yes | Attended |
| Male | 2023-04-05 | 2023-04-08 | 95 | Arlington | 1 | 0 | 0 | 0 | 0 | 0 | 3 | Yes | Attended |
| Male | 2023-05-27 | 2023-06-02 | 47 | Newport News | 1 | 1 | 0 | 0 | 0 | 1 | 6 | Yes | Attended |
| Male | 2023-05-13 | 2023-05-27 | 18 | Alexandria | 0 | 0 | 0 | 0 | 0 | 1 | 14 | Yes | Attended |
| Female | 2023-01-22 | 2023-01-25 | 5 | Norfolk | 0 | 1 | 0 | 0 | 0 | 0 | 3 | Yes | Attended |
| Male | 2023-05-01 | 2023-05-12 | 83 | Fairfax | 0 | 0 | 1 | 0 | 0 | 1 | 11 | Yes | Attended |
| Male | 2023-03-24 | 2023-03-28 | 26 | Newport News | 0 | 0 | 0 | 0 | 0 | 1 | 4 | No | No Show |
| Male | 2023-03-07 | 2023-03-10 | 52 | Virginia Beach | 1 | 0 | 0 | 0 | 0 | 1 | 3 | Yes | Attended |
| Female | 2023-06-07 | 2023-06-13 | 47 | Arlington | 0 | 0 | 0 | 0 | 0 | 1 | 6 | Yes | Attended |
| Male | 2023-01-21 | 2023-01-26 | 6 | Roanoke | 0 | 0 | 0 | 0 | 0 | 1 | 5 | Yes | Attended |

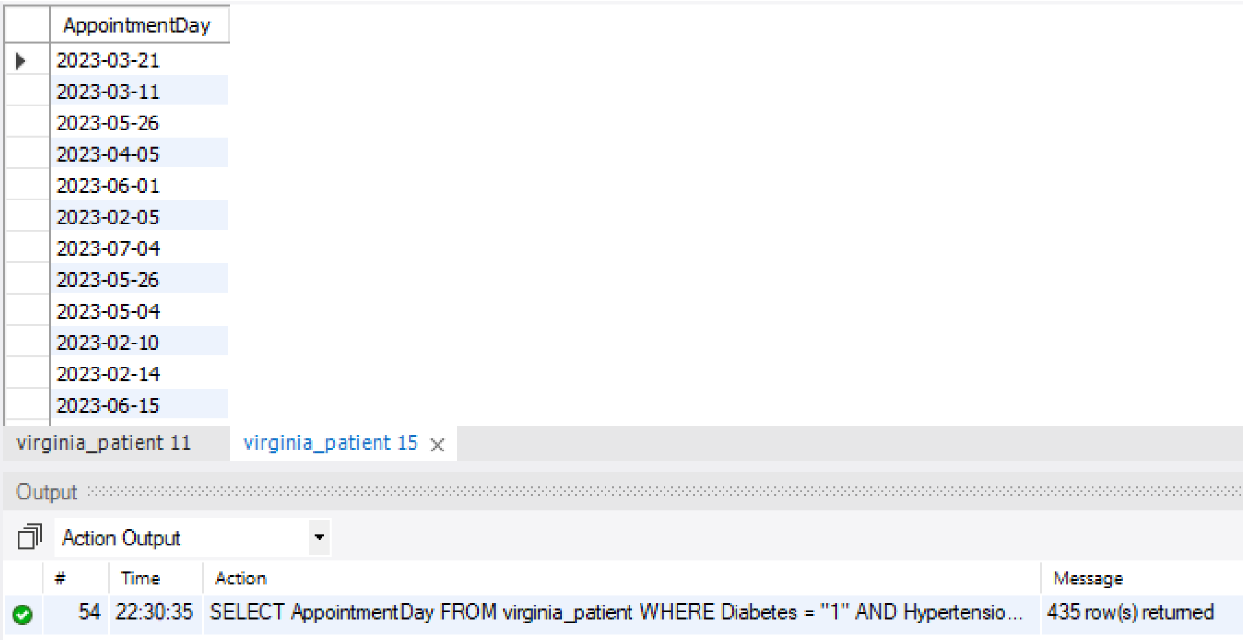
**Summary**:

This query adds a new column called AppointmentStatus to the virginia_patient table. It then fills this column using a CASE statement: if Showed_up is 'No', it sets AppointmentStatus to 'No Show'; otherwise, it sets it to 'Attended'. The first option adds the column and updates values separately, while the second option creates a computed/generated column directly.

### Q8: Filter appointments for diabetic patients with hypertension.

**SQL Query**:

```
SELECT AppointmentDay
FROM virginia_patient
WHERE Diabetes = "1" AND Hypertension = "1";
```



| AppointmentDay |
|---|
| 2023-03-21 |
| 2023-03-11 |
| 2023-05-26 |
| 2023-04-05 |
| 2023-06-01 |
| 2023-02-05 |
| 2023-07-04 |
| 2023-05-26 |
| 2023-05-04 |
| 2023-02-10 |
| 2023-02-14 |
| 2023-06-15 |

| # | Time | Action | Message |
|---|---|---|---|
| 54 | 22:30:35 | SELECT AppointmentDay FROM virginia_patient WHERE Diabetes = "1" AND Hypertensio... | 435 row(s) returned |

**Summary**:

This query retrieves the appointment dates (AppointmentDay) for patients who have both diabetes and hypertension, filtering the data to show only 435 cases.

### Q9: Order the records by Age in descending order and show only the top 5 oldest patients.

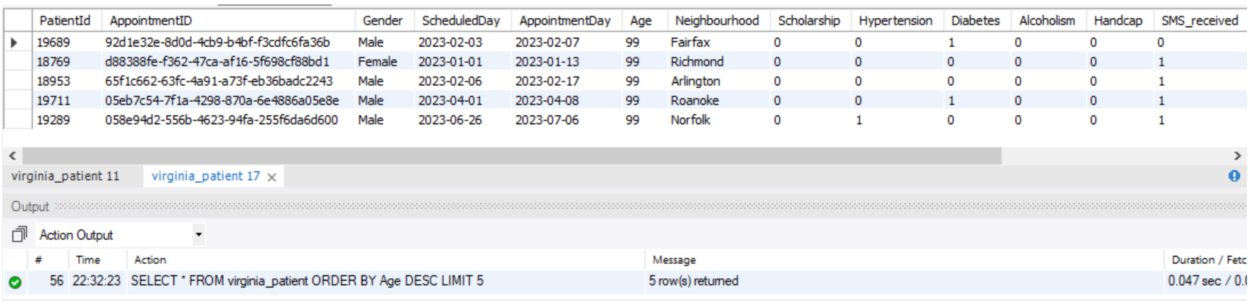
**SQL Query**:

SELECT *

FROM virginia_patient

ORDER BY Age DESC

LIMIT 5;



| | PatientId | AppointmentID | Gender | ScheduledDay | AppointmentDay | Age | Neighbourhood | Scholarship | Hypertension | Diabetes | Alcoholism | Handcap | SMS_received |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ▶ | 19689 | 92d1e32e-8d0d-4cb9-b4bf-f3cdfc6fa36b | Male | 2023-02-03 | 2023-02-07 | 99 | Fairfax | 0 | 0 | 1 | 0 | 0 | 0 |
| | 18769 | d88388fe-f362-47ca-af16-5f698cf88bd1 | Female | 2023-01-01 | 2023-01-13 | 99 | Richmond | 0 | 0 | 0 | 0 | 0 | 1 |
| | 18953 | 65f1c662-63fc-4a91-a73f-eb36badc2243 | Male | 2023-02-06 | 2023-02-17 | 99 | Arlington | 0 | 0 | 0 | 0 | 0 | 1 |
| | 19711 | 05eb7c54-7f1a-4298-870a-6e4886a05e8e | Male | 2023-04-01 | 2023-04-08 | 99 | Roanoke | 0 | 0 | 1 | 0 | 0 | 1 |
| | 19289 | 058e94d2-556b-4623-94fa-255f6da6d600 | Male | 2023-06-26 | 2023-07-06 | 99 | Norfolk | 0 | 1 | 0 | 0 | 0 | 1 |

| | # | Time | Action | Message | Duration / Fetc |
|---|---|---|---|---|---|
| ✓ | 56 | 22:32:23 | SELECT * FROM virginia_patient ORDER BY Age DESC LIMIT 5 | 5 row(s) returned | 0.047 sec / 0.( |

**Summary**:

This query retrieves all columns for the top 5 oldest patients, sorting the records by age in descending order to show the oldest first.

### Q10: Limit results to the first 5 appointments for patients under age 18.

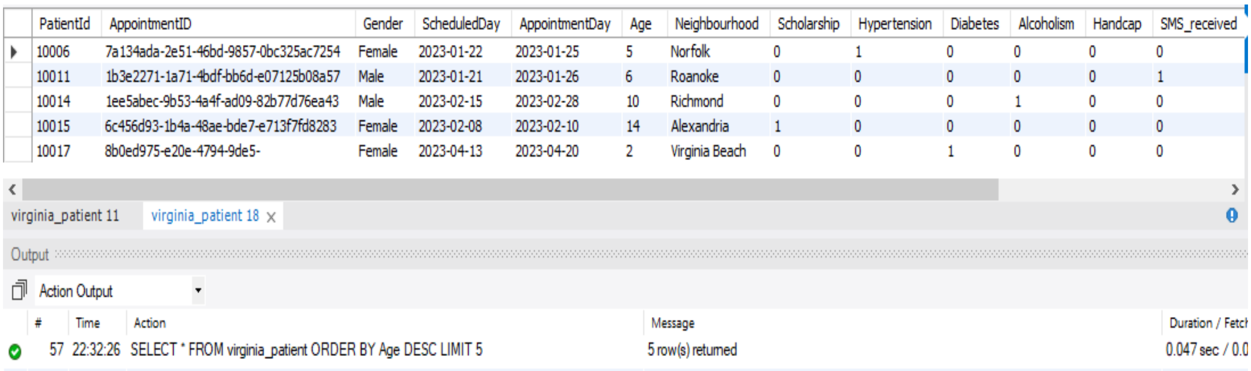
**SQL Query**:

SELECT *

FROM virginia_patient

WHERE Age < 18

LIMIT 5;



| | PatientId | AppointmentID | Gender | ScheduledDay | AppointmentDay | Age | Neighbourhood | Scholarship | Hypertension | Diabetes | Alcoholism | Handcap | SMS_received |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ▶ | 10006 | 7a134ada-2e51-46bd-9857-0bc325ac7254 | Female | 2023-01-22 | 2023-01-25 | 5 | Norfolk | 0 | 1 | 0 | 0 | 0 | 0 |
| | 10011 | 1b3e2271-1a71-4bdf-bb6d-e07125b08a57 | Male | 2023-01-21 | 2023-01-26 | 6 | Roanoke | 0 | 0 | 0 | 0 | 0 | 1 |
| | 10014 | 1ee5abec-9b53-4a4f-ad09-82b77d76ea43 | Male | 2023-02-15 | 2023-02-28 | 10 | Richmond | 0 | 0 | 0 | 1 | 0 | 0 |
| | 10015 | 6c456d93-1b4a-48ae-bde7-e713f7fd8283 | Female | 2023-02-08 | 2023-02-10 | 14 | Alexandria | 1 | 0 | 0 | 0 | 0 | 0 |
| | 10017 | 8b0ed975-e20e-4794-9de5- | Female | 2023-04-13 | 2023-04-20 | 2 | Virginia Beach | 0 | 0 | 1 | 0 | 0 | 0 |

| | # | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|---|
| ✓ | 57 | 22:32:26 | SELECT * FROM virginia_patient ORDER BY Age DESC LIMIT 5 | 5 row(s) returned | 0.047 sec / 0.0 |

**Summary**:

This query retrieves all columns for the first 5 appointments of patients younger than 18 years old, limiting the result to those five records.

## SECTION-3: AGGREGATION & GROUPING

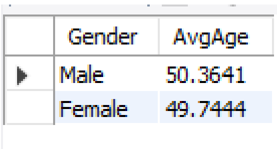
### Q11: Find the average age of patients for each gender.

**SQL Query**:

SELECT Gender, AVG(Age) AS AvgAge

FROM virginia_patient

GROUP BY Gender;



| | Gender | AvgAge |
|---|---|---|
| ▶ | Male | 50.3641 |
| | Female | 49.7444 |

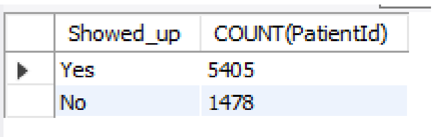
**Summary**:

This query calculates the average age of patients grouped by gender, showing the average age for each gender category.

### Q12: Count how many patients received SMS reminders, grouped by Showed_up status.

**SQL Query**:

```
SELECT Showed_up, COUNT(PatientId) AS sms_reminder_count
FROM virginia_patient
WHERE SMS_received = "1"
GROUP BY Showed_up;
```



| Showed_up | COUNT(PatientId) |
|---|---|
| Yes | 5405 |
| No | 1478 |

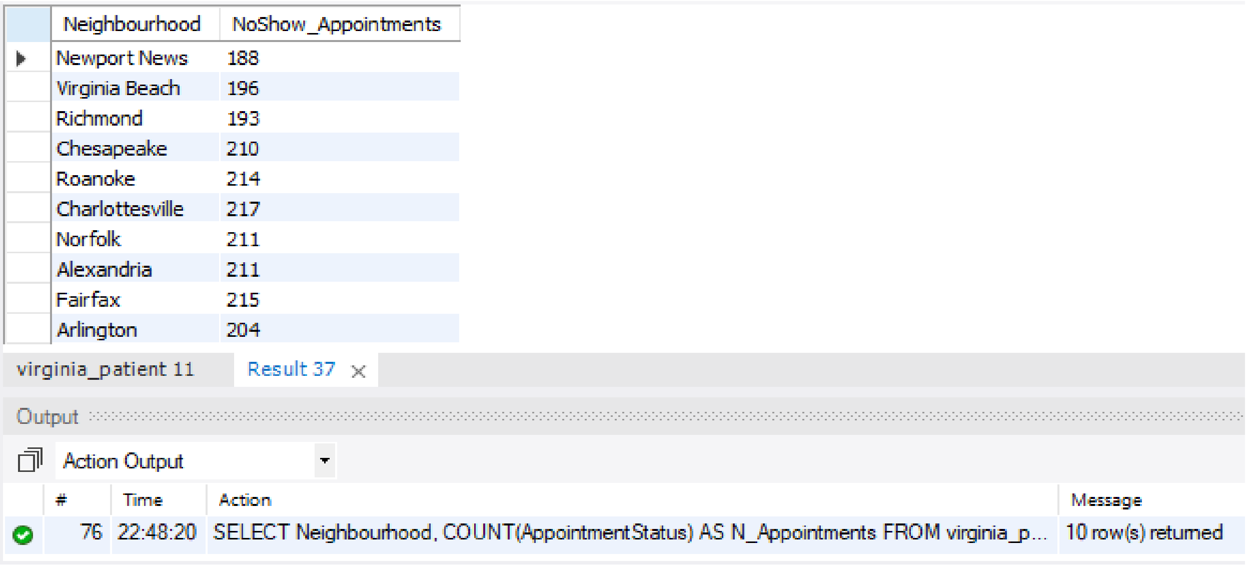
**Summary**:

This query counts the number of patients i.e. 6883 who received SMS reminders, grouping the results by their Showed_up status portraying that 5405 showed up versus 1478 who did not show up.

### Q13: Count no-show appointments in each Neighbourhood using GROUP BY.

**SQL Query**:

```
SELECT Neighbourhood, COUNT(AppointmentStatus) AS NoShow_Appointments
FROM virginia_patient
WHERE AppointmentStatus = "No Show"
GROUP BY Neighbourhood;
```



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

| | # | Time | Action | Message |
|---|---|---|---|---|
| ✓ | 76 | 22:48:20 | SELECT Neighbourhood, COUNT(AppointmentStatus) AS N_Appointments FROM virginia_p... | 10 row(s) returned |

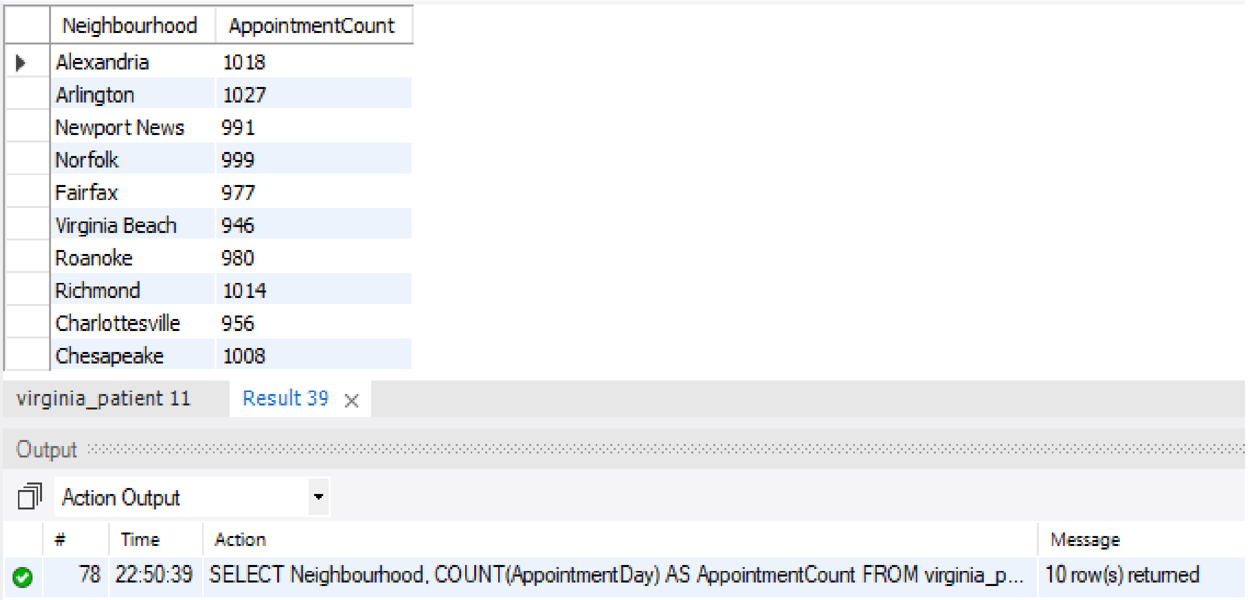
**Summary**:

This query counts the number of 'No Show' appointments for each neighbourhood by grouping the data based on the Neighbourhood column and filtering only those appointments marked as 'No Show'.

### Q14: Show Neighbourhoods with more than 100 total appointments (HAVING clause).

**SQL Query**:

```
SELECT Neighbourhood, COUNT(AppointmentDay) AS AppointmentCount
FROM virginia_patient
GROUP BY Neighbourhood
HAVING COUNT(AppointmentDay) > 100;
```



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

| # | Time | Action | Message |
|---|---|---|---|
| 78 | 22:50:39 | SELECT Neighbourhood, COUNT(AppointmentDay) AS AppointmentCount FROM virginia_p... | 10 row(s) returned |

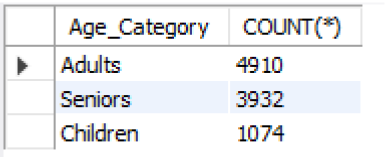
**Summary**:

This query lists neighbourhoods that have more than 100 total appointments by grouping appointments by neighbourhood and filtering groups with counts greater than 100 using the HAVING clause.

### Q15: Use CASE to calculate the total number of: children (Age < 12), adults (Age BETWEEN 12 AND 60), seniors (Age > 60).

**SQL Query**:

```
SELECT CASE WHEN Age < 12 THEN "Children"
                       WHEN Age BETWEEN 12 AND 60 THEN "Adults"
        WHEN Age > 60 THEN "Seniors"
        END AS Age_Category, COUNT(*)
FROM virginia_patient
GROUP BY Age_Category;
```



| Age_Category | COUNT(*) |
|---|---|
| Adults | 4910 |
| Seniors | 3932 |
| Children | 1074 |

**Summary**:

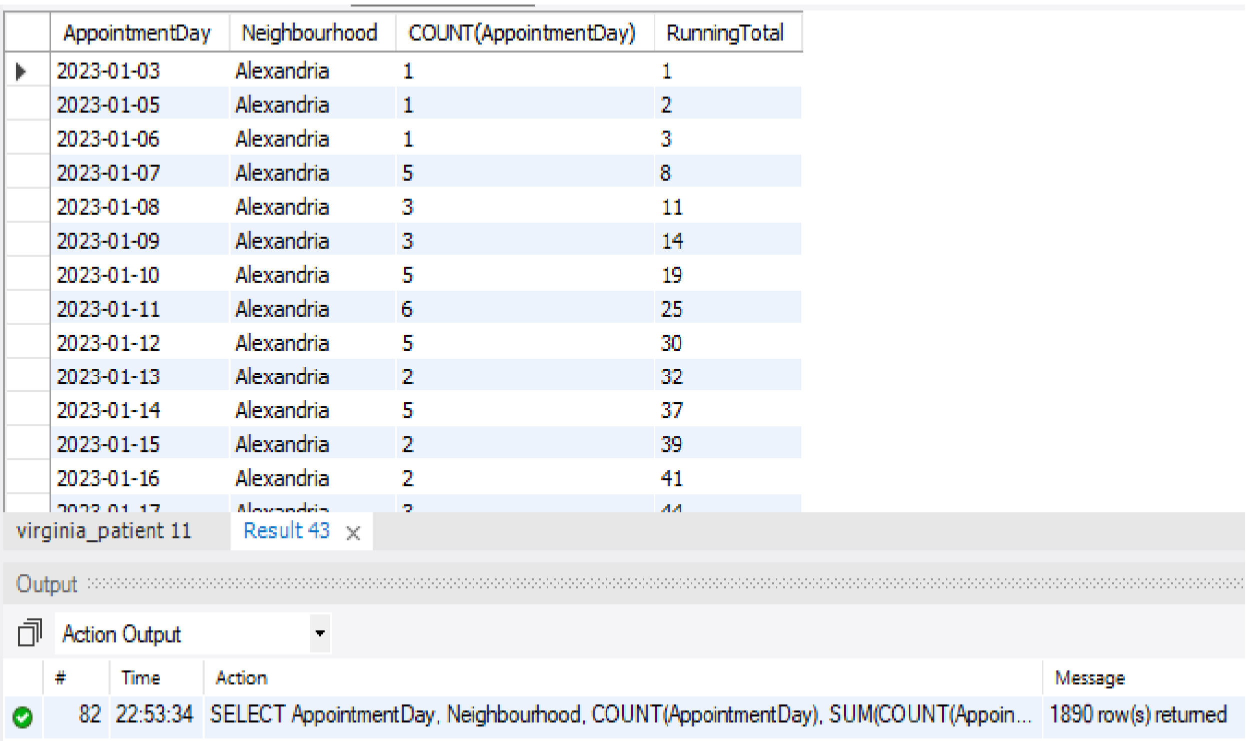
This query categorizes patients into 'Children', 'Adults', and 'Seniors' based on their age using a CASE statement, then counts how many patients fall into each age category by grouping the results accordingly.

## SECTION-4: WINDOWS FUNCTION

### Q16: How many appointments were there each day and how do the total appointments keep adding up over time in each Neighbourhood?

**SQL Query**:

```
SELECT AppointmentDay, Neighbourhood,
                       COUNT(AppointmentDay), SUM(COUNT(AppointmentDay)) OVER(PARTITION BY
                       Neighbourhood ORDER BY AppointmentDay) AS RunningTotal
FROM virginia_patient
GROUP BY Neighbourhood, AppointmentDay
ORDER BY Neighbourhood, AppointmentDay;
```



| | AppointmentDay | Neighbourhood | COUNT(AppointmentDay) | RunningTotal |
|---|---|---|---|---|
| ▶ | 2023-01-03 | Alexandria | 1 | 1 |
| | 2023-01-05 | Alexandria | 1 | 2 |
| | 2023-01-06 | Alexandria | 1 | 3 |
| | 2023-01-07 | Alexandria | 5 | 8 |
| | 2023-01-08 | Alexandria | 3 | 11 |
| | 2023-01-09 | Alexandria | 3 | 14 |
| | 2023-01-10 | Alexandria | 5 | 19 |
| | 2023-01-11 | Alexandria | 6 | 25 |
| | 2023-01-12 | Alexandria | 5 | 30 |
| | 2023-01-13 | Alexandria | 2 | 32 |
| | 2023-01-14 | Alexandria | 5 | 37 |
| | 2023-01-15 | Alexandria | 2 | 39 |
| | 2023-01-16 | Alexandria | 2 | 41 |
| | 2023-01-17 | Alexandria | 3 | 44 |

| | # | Time | Action | Message |
|---|---|---|---|---|
| ✓ | 82 | 22:53:34 | SELECT AppointmentDay, Neighbourhood, COUNT(AppointmentDay), SUM(COUNT(Appoin... | 1890 row(s) returned |

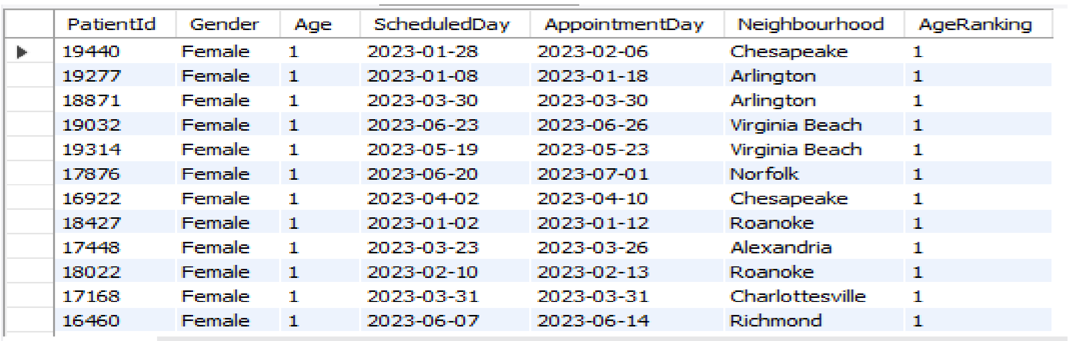
**Summary**:

This query shows how many appointments happened each day in every neighbourhood and calculates a running total that adds up appointments day by day within each neighbourhood, tracking the accumulation over time.

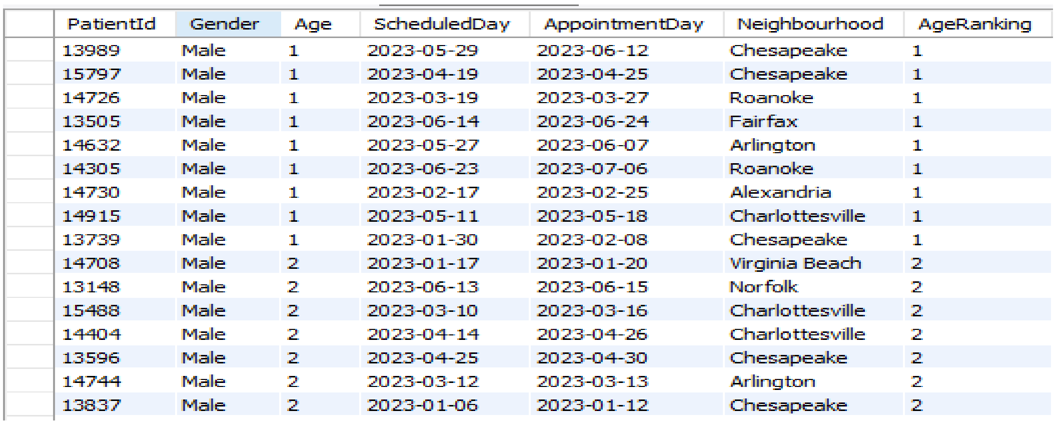
**Q17: Use Dense_Rank() to rank patients by age within each gender group.**

**SQL Query**:

```
SELECT PatientId, Gender, Age, ScheduledDay, AppointmentDay, Neighbourhood,
                    DENSE_RANK() OVER (PARTITION BY Gender ORDER BY Age) AS AgeRanking
FROM virginia_patient;
```



| | PatientId | Gender | Age | ScheduledDay | AppointmentDay | Neighbourhood | AgeRanking |
|---|---|---|---|---|---|---|---|
| ▶ | 19440 | Female | 1 | 2023-01-28 | 2023-02-06 | Chesapeake | 1 |
| | 19277 | Female | 1 | 2023-01-08 | 2023-01-18 | Arlington | 1 |
| | 18871 | Female | 1 | 2023-03-30 | 2023-03-30 | Arlington | 1 |
| | 19032 | Female | 1 | 2023-06-23 | 2023-06-26 | Virginia Beach | 1 |
| | 19314 | Female | 1 | 2023-05-19 | 2023-05-23 | Virginia Beach | 1 |
| | 17876 | Female | 1 | 2023-06-20 | 2023-07-01 | Norfolk | 1 |
| | 16922 | Female | 1 | 2023-04-02 | 2023-04-10 | Chesapeake | 1 |
| | 18427 | Female | 1 | 2023-01-02 | 2023-01-12 | Roanoke | 1 |
| | 17448 | Female | 1 | 2023-03-23 | 2023-03-26 | Alexandria | 1 |
| | 18022 | Female | 1 | 2023-02-10 | 2023-02-13 | Roanoke | 1 |
| | 17168 | Female | 1 | 2023-03-31 | 2023-03-31 | Charlottesville | 1 |
| | 16460 | Female | 1 | 2023-06-07 | 2023-06-14 | Richmond | 1 |



| | PatientId | Gender | Age | ScheduledDay | AppointmentDay | Neighbourhood | AgeRanking |
|---|---|---|---|---|---|---|---|
| | 13989 | Male | 1 | 2023-05-29 | 2023-06-12 | Chesapeake | 1 |
| | 15797 | Male | 1 | 2023-04-19 | 2023-04-25 | Chesapeake | 1 |
| | 14726 | Male | 1 | 2023-03-19 | 2023-03-27 | Roanoke | 1 |
| | 13505 | Male | 1 | 2023-06-14 | 2023-06-24 | Fairfax | 1 |
| | 14632 | Male | 1 | 2023-05-27 | 2023-06-07 | Arlington | 1 |
| | 14305 | Male | 1 | 2023-06-23 | 2023-07-06 | Roanoke | 1 |
| | 14730 | Male | 1 | 2023-02-17 | 2023-02-25 | Alexandria | 1 |
| | 14915 | Male | 1 | 2023-05-11 | 2023-05-18 | Charlottesville | 1 |
| | 13739 | Male | 1 | 2023-01-30 | 2023-02-08 | Chesapeake | 1 |
| | 14708 | Male | 2 | 2023-01-17 | 2023-01-20 | Virginia Beach | 2 |
| | 13148 | Male | 2 | 2023-06-13 | 2023-06-15 | Norfolk | 2 |
| | 15488 | Male | 2 | 2023-03-10 | 2023-03-16 | Charlottesville | 2 |
| | 14404 | Male | 2 | 2023-04-14 | 2023-04-26 | Charlottesville | 2 |
| | 13596 | Male | 2 | 2023-04-25 | 2023-04-30 | Chesapeake | 2 |
| | 14744 | Male | 2 | 2023-03-12 | 2023-03-13 | Arlington | 2 |
| | 13837 | Male | 2 | 2023-01-06 | 2023-01-12 | Chesapeake | 2 |

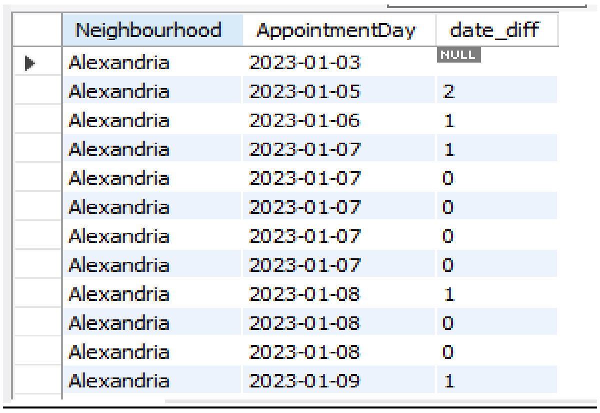
**Summary**:

This query assigns a rank to patients based on their age within each gender group using DENSE_RANK(), so patients of the same age get the same rank, and the ranking resets for each gender.

### Q18: How many days have passed since the last appointment in the same Neighbourhood?

**SQL Query**:

```
SELECT Neighbourhood, AppointmentDay,
                    DATEDIFF(AppointmentDay, LAG (AppointmentDay) OVER (PARTITION BY
Neighbourhood ORDER BY AppointmentDay)) AS date_diff
FROM virginia_patient;
```



| Neighbourhood | AppointmentDay | date_diff |
|---|---|---|
| Alexandria | 2023-01-03 | NULL |
| Alexandria | 2023-01-05 | 2 |
| Alexandria | 2023-01-06 | 1 |
| Alexandria | 2023-01-07 | 1 |
| Alexandria | 2023-01-07 | 0 |
| Alexandria | 2023-01-07 | 0 |
| Alexandria | 2023-01-07 | 0 |
| Alexandria | 2023-01-07 | 0 |
| Alexandria | 2023-01-08 | 1 |
| Alexandria | 2023-01-08 | 0 |
| Alexandria | 2023-01-08 | 0 |
| Alexandria | 2023-01-09 | 1 |

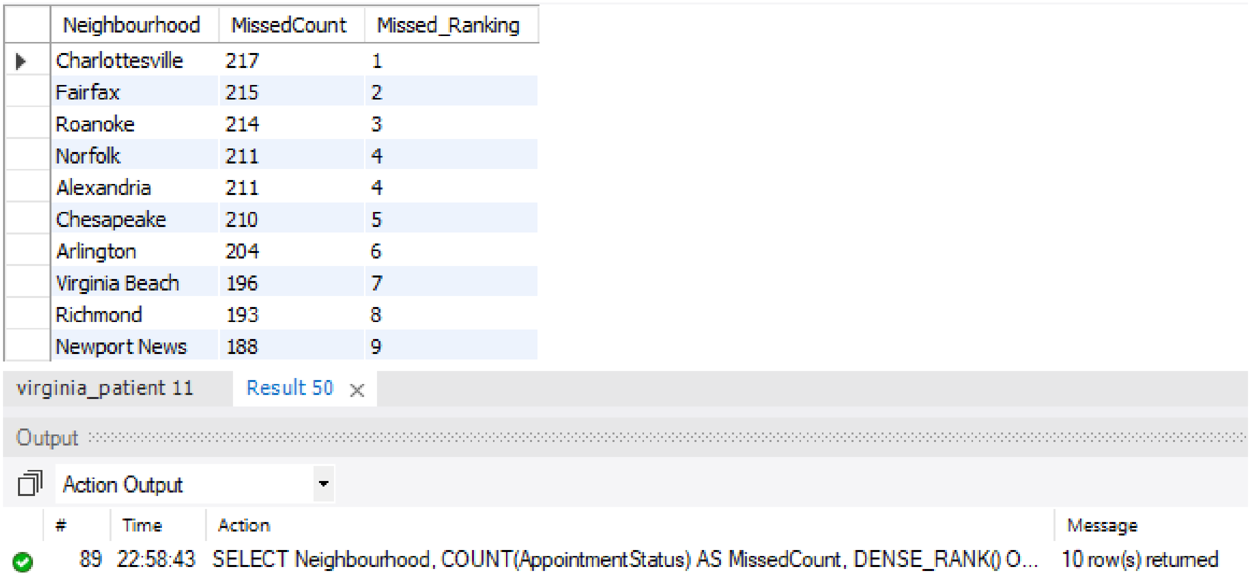
**Summary**:

This query calculates the number of days between each appointment and the previous appointment within the same neighbourhood, showing how frequently appointments happen over time in each area.

### Q19: Which Neighbourhoods have the highest number of missed appointments?

**SQL Query**:

```
SELECT Neighbourhood, COUNT(AppointmentStatus) AS MissedCount,
                              DENSE_RANK() OVER (ORDER BY COUNT(AppointmentStatus) DESC) AS
                              Missed_Ranking
FROM virginia_patient
WHERE AppointmentStatus = 'No Show'
GROUP BY Neighbourhood;
```



| Neighbourhood | MissedCount | Missed_Ranking |
|---|---|---|
| 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 |

```
#   Time      Action                                                                   Message
89  22:58:43  SELECT Neighbourhood, COUNT(AppointmentStatus) AS MissedCount, DENSE_RANK() O...  10 row(s) returned
```

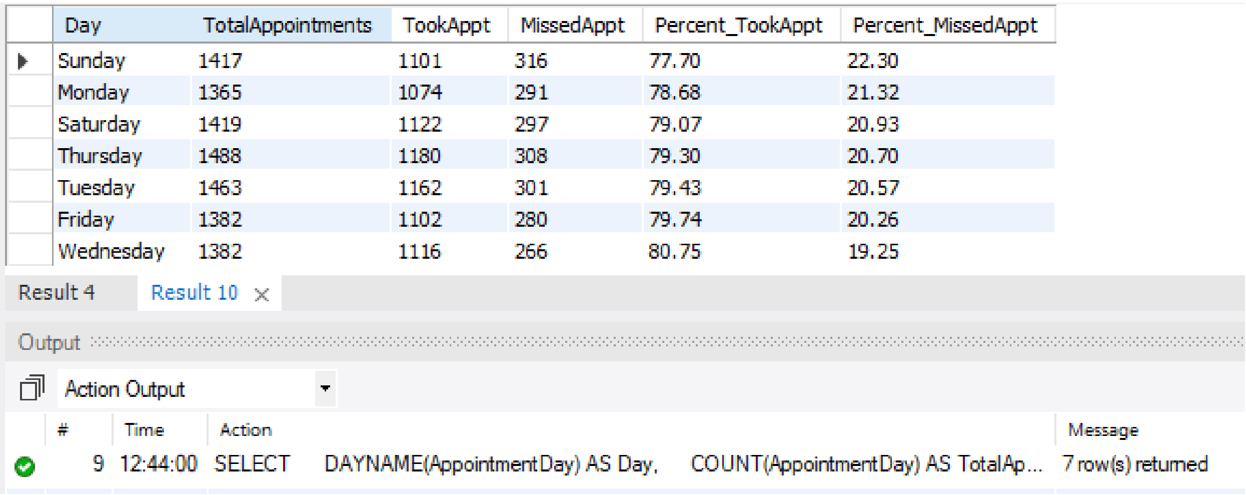
**Summary**:

This query counts the number of missed (no-show) appointments in each neighbourhood and ranks the neighbourhoods from highest to lowest based on these counts using DENSE_RANK().

**Q20: Are patients more likely to miss appointments on certain days of the week?**

__SQL Query__:

```
SELECT
	DAYNAME(AppointmentDay) AS Day,
	COUNT(AppointmentDay) AS TotalAppointments,
	COUNT(CASE WHEN Showed_up = 'Yes' THEN 1 END) AS TookAppt,
	COUNT(CASE WHEN Showed_up = 'No' THEN 1 END) AS MissedAppt,
	ROUND(COUNT(CASE WHEN Showed_up = 'Yes' THEN 1 END) /COUNT(AppointmentDay)  * 100,
	2) AS Percent_TookAppt,
	ROUND(COUNT(CASE WHEN Showed_up = 'No' THEN 1 END)/ COUNT(AppointmentDay) * 100,
	2) AS Percent_MissedAppt
FROM virginia_patient
GROUP BY DAYNAME(AppointmentDay)
ORDER BY Percent_MissedAppt DESC;
```



| Day | TotalAppointments | TookAppt | MissedAppt | Percent_TookAppt | Percent_MissedAppt |
|---|---|---|---|---|---|
| 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 |

| # | Time | Action | Message |
|---|---|---|---|
| 9 | 12:44:00 | SELECT  DAYNAME(AppointmentDay) AS Day,  COUNT(AppointmentDay) AS TotalAp... | 7 row(s) returned |

**Summary**:

This query analyses appointment attendance by day of the week. It extracts the day name from AppointmentDay, counts total appointments, how many were attended, and how many were missed. It then calculates the percentage of shows and no-shows per day, rounding to two decimals, and sorts the days to highlight those with the highest no-show rates first.

## 4. Key Findings

a. Senior patients (>60 years) form a significant group with specific healthcare needs.

b. SMS reminders effectively correlate with higher attendance, though 1478 patients still missed appointments despite SMS reminders.

c. No-show rates vary significantly by Neighbourhood and day of the week, indicating potential areas for operational improvements.

d. Chronic conditions like diabetes and hypertension are present in 435 appointments, emphasizing the need for tailored follow-ups.

e. Children, adults, and seniors show distinct appointment behaviors and volume distributions.

f. Appointment frequency varies by Neighbourhood with some areas showing longer gaps between visits.

## 5. Recommendations

a. Enhance SMS reminder campaigns especially targeting Neighbourhoods and days with high no-show rates.

b. Implement Neighbourhood-specific outreach programs focusing on areas with frequent missed appointments.

c. Focus on senior and chronic condition patients with personalized care plans and appointment follow-ups.

d. Schedule appointments strategically on days with lower no-show percentages to maximize attendance.

e. Use running total and appointment frequency data for dynamic resource allocation in clinics.

f. Monitor age group distributions to design age-appropriate health interventions.

**6. Conclusion**. The analysis of the 'virginia_patient' dataset highlights key demographic, geographic, and temporal patterns in patient appointment behaviours. While SMS reminders help improve attendance, certain neighbourhoods and days still experience high no-show rates, suggesting room for targeted interventions. By leveraging these insights, healthcare providers can optimize appointment management, improve patient engagement, and enhance care delivery.