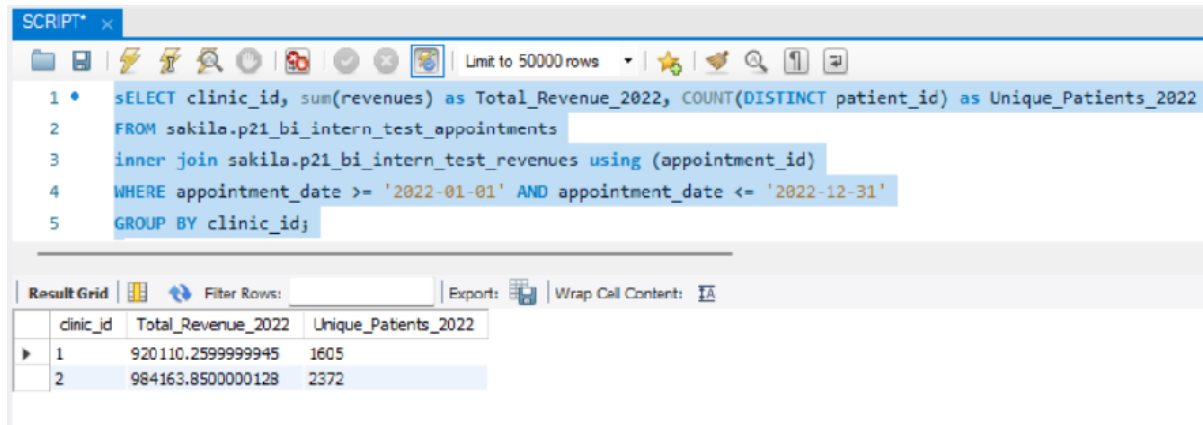


LOGIC

- Converted date format into month using **=text(A2,"mmmm")**.
- Used VLOOKUP to join two CSV files given, so that the revenue column is imported into the appointment table using appointment_id. (It is the same in both tables)
- Called this new table name **Dental_clinic_21**. Imported CSV file into SQL, and by SQL Querying found
- **Total_Revenue_2022 and unique_patients_2022.**



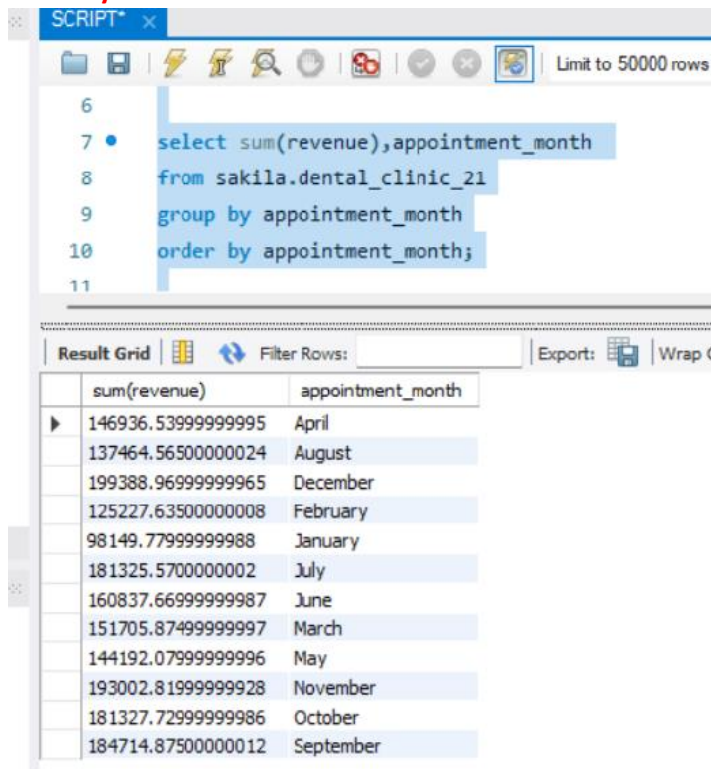
The screenshot shows a SQL script editor with a query to calculate total revenue and unique patients for 2022, grouped by clinic_id. The query is as follows:

```
1 • sSELECT clinic_id, sum(revenues) as Total_Revenue_2022, COUNT(DISTINCT patient_id) as Unique_Patients_2022
2 FROM sakila.p21_bi_intern_test_appointments
3 inner join sakila.p21_bi_intern_test_revenues using (appointment_id)
4 WHERE appointment_date >= '2022-01-01' AND appointment_date <= '2022-12-31'
5 GROUP BY clinic_id;
```

The result grid shows the following data:

clinic_id	Total_Revenue_2022	Unique_Patients_2022
1	920110.2599999945	1605
2	984163.8500000128	2372

- **Monthly revenue in 2022**



The screenshot shows a SQL script editor with a query to calculate monthly revenue for 2022, grouped by appointment_month. The query is as follows:

```
6
7 • select sum(revenue),appointment_month
8 from sakila.dental_clinic_21
9 group by appointment_month
10 order by appointment_month;
11
```

The result grid shows the following data:

sum(revenue)	appointment_month
146936.53999999995	April
137464.56500000024	August
199388.96999999965	December
125227.63500000008	February
98149.77999999988	January
181325.57000000002	July
160837.66999999987	June
151705.87499999997	March
144192.07999999996	May
193002.81999999928	November
181327.72999999986	October
184714.87500000012	September

- Monthly revenue in 2022 with clinic_id_1 and clinic_id_2

Limit to 50000 rows

```

33
34 • select sum(revenue),appointment_month,clinic_id
35 from sakila.dental_clinic_21
36 group by clinic_id,appointment_month
37 order by appointment_month;
38

```

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

	sum(revenue)	appointment_month	clinic_id
▶	87678.26999999989	April	1
	59258.27000000004	April	2
	79411.47999999995	August	1
	58053.08500000007	August	2
	102380.63499999985	December	2
	97008.33499999996	December	1
	67169.45000000003	February	1
	58058.18500000005	February	2
	49252.009999999995	January	1
	48897.770000000055	January	2
	101954.10000000008	July	2
	79371.46999999999	July	1
	107148.67000000006	June	2

- Monthly Unique patients visited in 2022.

SCRIPT x

Limit to 50000 rows

```

23
24 • SELECT count(DISTINCT patient_id) as monthly_Unique_Patients_2022,appointment_month
25 FROM sakila.dental_clinic_21
26 GROUP BY appointment_month
27 order by appointment_month;
28

```

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

	monthly_Unique_Patients_2022	appointment_month
▶	555	April
	644	August
	888	December
	510	February
	499	January
	631	July
	536	June
	515	March
	582	May
	816	November
	769	October
	740	September

- Monthly Unique patients visited in clinic_id_1 and clinic_id_2 in 2022

Limit to 50000 rows

```

23
24 • SELECT count(DISTINCT patient_id) as monthly_Unique_Patients_2022,appointment_month,clinic_id
25 FROM sakila.dental_clinic_21
26 GROUP BY appointment_month,clinic_id
27 order by appointment_month;
28

```

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

monthly_Unique_Patients_2022	appointment_month	clinic_id
261	April	1
294	April	2
348	August	1
296	August	2
378	December	1
510	December	2
244	February	1
266	February	2
260	January	1
239	January	2
315	July	1
316	July	2
243	June	1

- Average monthly revenue in 2022

SCRIPT x

Limit to

```

18
19 • select avg(revenue),appointment_month
20 from sakila.dental_clinic_21
21 group by appointment_month
22 order by appointment_month desc;
23

```

Result Grid | Filter Rows: | Export: |

avg(revenue)	appointment_month
175.5844819391636	September
158.78084938704015	October
162.59715248525634	November
176.2739364303178	May
202.54455941255003	March
227.17185028248568	June
208.90042626728135	July
138.43410437235525	January
165.42620211360645	February
163.03268192968082	December
157.10236000000026	August
180.51171990171983	April

- **Average monthly revenue in clinic_id_1 and clinic_id_2 in 2022**

SCRIPT x

Limit to 50000 rows

```

28
29 • select avg(revenue),appointment_month,clinic_id
30 from sakila.dental_clinic_21
31 group by clinic_id,appointment_month
32 order by appointment_month;
33

```

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

	avg(revenue)	appointment_month	clinic_id
▶	213.84943902438997	April	1
	146.67888613861396	April	2
	158.50594810379232	August	1
	155.2221524064173	August	2
	162.25140253565743	December	2
	163.86543074324317	December	1
	169.19256926952147	February	1
	161.27273611111124	February	2
	124.37376262626262	January	1
	156.22290734824298	January	2
	233.30457665903907	July	2
	184.15654292343385	July	1
	281.97018421052644	June	2

Following are the Excel steps followed to find the final solution -:

- 1) **Estimated revenue in 2023.**
- 2) **Estimated unique patients in 2023.**
- 3) **I also Estimated the average revenue per month in 2023.**

Two sheets are generated to find the above parameters. I consider the above three solutions as PART 1 procedure

PART 1 – (PROCEDURE involved on sheets of estimated revenue/unique patients/average revenue per month in 2023.)

It is a similar process I followed to find all the above-mentioned parameters. So I am mentioning the process I followed.

STEP-1 Exported the result of SQL from the above-mentioned screen shorts into Excel respectively.

STEP-2 Used sort to arrange the (total /average - revenue/unique patients per month in 2022)

STEP-3 Used GROWTH FUNCTION TO Forecast the respected values in the growth forecast column.
(Formula – **GROWTH(C2: C13)**)

STEP-4 Dragged the growth forecast column as values. (COPY-PASTED AS VALUES BEFORE DRAGGING).

Because it will calculate according to the growth formula used.

Step 5 Made an actual growth forecast column where clinics 2 will be added in March and July 2023 respectively.

Step 6 Dragged the values till feb-2023. In March, clinic 3 and Clinic 4 were introduced respectively.

Clinic 3 – I took the average of both clinic 1 and clinic 2 to calculate (total/average revenue and unique patients) in their respective parameters/Sheets.

Clinic 4 - I took the average of three clinics 1, 2 and 3 to calculate (total/average revenue and unique patients) in their respective parameters/Sheets.

STEP 7 – I also calculated % growth for respective sheets.

PART 2-: (Sheets where an average of both is calculated)

STEP 1 - Exported the result of SQL from the above-mentioned screen shorts into Excel respectively.

Step 2 Used sort to arrange the (total /average - revenue/unique patients per month in 2022), but concerning clinic id 1 and clinic id 2

Step 3 Average of clinic id 1 and clinic id 2. Check the respective formula in their respective sheets.

Step 4 calculated the average of both clinics. Check the respective formula in their respective sheets.

Step 5 Used GROWTH FUNCTION TO find the growth forecast for the average of both clinics' columns.

Dragged the growth forecast column as values. (COPY-PASTED AS VALUES BEFORE DRAGGING).

Because it will calculate according to the growth formula used.

For Better understanding -: Made Dashboards

SQL QUERIES USED-:

- **Total_Revenue_2022 and unique_patients_2022.**

```
Select clinic_id, sum(revenues) as Total_Revenue_2022, COUNT(DISTINCT patient_id) as Unique_Patients_2022
```

```
FROM sakila.p21_bi_intern_test_appointments
```

```
inner join sakila.p21_bi_intern_test_revenues using (appointment_id)
```

```
WHERE appointment_date >= '2022-01-01' AND appointment_date <= '2022-12-31'
```

```
GROUP BY clinic_id;
```

- **Monthly revenue in 2022**

```
select sum(revenue),appointment_month
```

```
from sakila.dental_clinic_21
```

```
group by appointment_month
```

```
order by appointment_month desc;
```

- **Monthly revenue in 2022 with clinic_id_1 and clinic_id_2**

```
select sum(revenue),appointment_month,clinic_id
from sakila.dental_clinic_21
group by clinic_id,appointment_month
order by appointment_month desc;
```

- **Monthly Unique patients visited in 2022**

```
SELECT COUNT(DISTINCT patient_id) as monthly_Unique_Patients_2022,appointment_month
FROM sakila.dental_clinic_21
GROUP BY appointment_month
order by appointment_month;
```

- **Monthly Unique patients visited in clinic_id_1 and clinic_id_2 in 2022**

```
SELECT COUNT (DISTINCT patient_id) as
monthly_Unique_Patients_2022,appointment_month,clinic_id
FROM sakila.dental_clinic_21
GROUP BY appointment_month,clinic_id
order by appointment_month;
```

- **Average monthly revenue in 2022**

```
select avg(revenue),appointment_month
from sakila.dental_clinic_21
group by appointment_month
order by appointment_month desc;
```

- **Average monthly revenue in clinic_id_1 and clinic_id_2 in 2022**

```
select avg(revenue),appointment_month,clinic_id
from sakila.dental_clinic_21
group by clinic_id,appointment_month
order by appointment_month;
```

