Tool used:

> PostgreSQL

> Tableau public

Tasks and Solutions:

#1. Import the data: Create a SQL database using the attached CSV files

- ‘assignment\_data\_clinic\_group.csv’ contains clinic group data (clinic IDs and Clinic Groups)

- ‘assignment\_data\_clinics\_with\_patients.csv’ contains patient/clinic data (patient IDs, clinic title, clinic IDs, patient created at date and patient deleted at date)

- ‘assignment\_data\_modules.csv’ contains module data (patient IDs, date of module generation, number of modules)

Solution-#1

[SQL Code]

//Step1: Database creation

CREATE DATABASE rehab\_db;

//Step2: Changing date format, setting the date format to match DD.MM.YY

SET datestyle = 'DMY';

Note: PostgreSQL expects dates in a specific format by default (YYYY-MM-DD) which was mismatching with the date in CSV file.

So I set the datestyle in PostgreSQL to match CSV date format.

Step3: Table creation for all 3 datasets (with Primary key and Foreign key relationship)

CREATE TABLE clinic\_group (

clinic\_id VARCHAR(255) PRIMARY KEY,

clinic\_group VARCHAR(255)

);

CREATE TABLE clinics\_with\_patients (

patient\_id INT PRIMARY KEY,

clinic\_title VARCHAR(255),

clinic\_id VARCHAR(255),

patient\_created\_date DATE,

patient\_deleted\_date DATE,

FOREIGN KEY (clinic\_id) REFERENCES clinic\_group(clinic\_id)

);

CREATE TABLE modules (

patient\_id INT,

module\_completion\_date DATE,

number\_of\_modules INT,

FOREIGN KEY (patient\_id) REFERENCES clinics\_with\_patients(patient\_id)

);

Step4: Importing datasets into newly created table in Step3 (.csv file)

//Upload data into clinic\_group table

COPY clinic\_group(clinic\_id, clinic\_group)

FROM '/Users/priyankamittal/Desktop/Data Analyst Challenge/assignment\_data\_clinic\_group.csv'

DELIMITER ','

CSV HEADER;

//Verifying successful data imports in clinic\_group table

SELECT \* FROM clinic\_group;

Output:



//Upload data into clinics\_with\_patients table

COPY clinics\_with\_patients(patient\_id, clinic\_title, clinic\_id, patient\_created\_date, patient\_deleted\_date )

FROM '/Users/priyankamittal/Desktop/Data Analyst Challenge/assignment\_data\_clinics\_with\_patients.csv'

DELIMITER ','

CSV HEADER;

//Verifying successful data imports in clinics\_with\_patients table (output displayed for first 10 records only)

SELECT \* FROM clinics\_with\_patients

LIMIT 10

Output:



//Upload data into modules table

COPY modules(patient\_id, module\_completion\_date, number\_of\_modules)

FROM '/Users/priyankamittal/Desktop/Data Analyst Challenge/assignment\_data\_modules.csv'

DELIMITER ','

CSV HEADER;

//Verifying successful data imports in modules table (output displayed for first 10 records only)

SELECT \* FROM modules

LIMIT 10

Output:



#2. Create a dataset: use SQL to combine data from imported tables. Final table should contain following columns:

patient ID, clinic ID, clinic title, clinic group, patient created at date, module completion date, number of modules

Solution-#2

// Joining the tables clinics\_with\_patients and clinic\_group using INNER JOIN and the resulting table with modules using LEFT JOIN

CREATE TABLE combined\_data AS

SELECT

p.patient\_id,

p.clinic\_id,

p.clinic\_title,

c.clinic\_group,

p.patient\_created\_date,

m.module\_completion\_date,

m.number\_of\_modules

FROM

clinics\_with\_patients p

INNER JOIN

clinic\_group c ON p.clinic\_id = c.clinic\_id

LEFT JOIN

modules m ON p.patient\_id = m.patient\_id;

Note:

- An inner join is used to connect clinics\_with\_patients and clinic\_groups because all unique clinic ID's from clinics\_with\_patients table have corresponding record in clinic\_groups

- A left join is used to connect resulting table from above step with modules because the module dataset only contains information for patients who have generated modules.

- This approach also accounts for patients who have not generated any modules.

//Verifying successful joining of 3 tables (output displayed for first 10 records only)

SELECT \* FROM combined\_data

LIMIT 10

Output:



Note: Total rows : 42104

#3. Explore the dataset: perform exploratory data analysis. Include your written observations

Make a list of things you find ‘strange’ or things that don’t make sense. The data is not perfect...

Solution-#3

// Extracting summary statistics (Understanding of the data)

SELECT

MIN(patient\_created\_date) AS earliest\_patient\_created\_date,

MAX(patient\_created\_date) AS latest\_patient\_created\_date,

MIN(module\_completion\_date) AS earliest\_module\_completion\_date,

MAX(module\_completion\_date) AS latest\_module\_completion\_date,

MIN(number\_of\_modules) AS min\_number\_of\_modules,

MAX(number\_of\_modules) AS max\_number\_of\_modules,

AVG(number\_of\_modules) AS avg\_number\_of\_modules

FROM

combined\_data;

Output:



// Checking for duplicate records

SELECT

patient\_id, clinic\_id, clinic\_title, clinic\_group, patient\_created\_date, module\_completion\_date, number\_of\_modules,

COUNT(\*) AS count\_duplicates

FROM

combined\_data

GROUP BY

patient\_id, clinic\_id, clinic\_title, clinic\_group, patient\_created\_date, module\_completion\_date, number\_of\_modules

HAVING

COUNT(\*) > 1;

Output



Key Findings:

- Total 7 duplicate records found in the combined data

// Deleting duplicate records using internal pseudo ID available in PostgreSQL

DELETE FROM combined\_data

WHERE ctid in (

SELECT MAX(ctid)

FROM combined\_data

GROUP BY patient\_id, clinic\_id, clinic\_title, clinic\_group, patient\_created\_date, module\_completion\_date, number\_of\_modules

HAVING COUNT(\*) > 1)

Output:

DELETE 7

Query returned successfully in 229 msec.

// Calculating total records after removing duplicates

SELECT \* FROM combined\_data

Key Findings : Total rows is 42097

// Checking for Null Values

SELECT

SUM(CASE WHEN patient\_id IS NULL THEN 1 ELSE 0 END) AS null\_patient\_id,

SUM(CASE WHEN clinic\_id IS NULL THEN 1 ELSE 0 END) AS null\_clinic\_id,

SUM(CASE WHEN clinic\_title IS NULL THEN 1 ELSE 0 END) AS null\_clinic\_title,

SUM(CASE WHEN clinic\_group IS NULL THEN 1 ELSE 0 END) AS null\_clinic\_group,

SUM(CASE WHEN patient\_created\_date IS NULL THEN 1 ELSE 0 END) AS null\_patient\_created\_date,

SUM(CASE WHEN module\_completion\_date IS NULL THEN 1 ELSE 0 END) AS null\_module\_completion\_date,

SUM(CASE WHEN number\_of\_modules IS NULL THEN 1 ELSE 0 END) AS null\_number\_of\_modules

FROM

combined\_data;

Output



Key Findings:

- 9,286 rows have missing clinic group value

- 16,306 patients with no module completion record, therefore module completion date and number of modules values are not punched

// Checking if null values for clinic groups can be filled using clinic group name assigned to the same clinic title in the dataset

SELECT

clinic\_group, COUNT(\*) AS count, clinic\_id,clinic\_title

FROM

combined\_data

GROUP BY

clinic\_group , clinic\_id,clinic\_title

ORDER BY

count DESC;

Output:



Key Findings:

- No clinic group assigned for all records of clinic title "Rehab 101" and "Rehab on the hill" and clinic id "abc11" and "abc7"

// Filling NULL values for clinic\_groups with 'Not assigned' as it contributes to 22 % of records and cannot be deleted

UPDATE combined\_data

SET clinic\_group = 'Not Assigned'

WHERE clinic\_group IS NULL;

// Validating all null values in clinic\_group are replaced with "Not Assigned"

SELECT clinic\_group

FROM combined\_data

WHERE clinic\_group IS NULL

// Filling NULL values for number\_of\_modules with '0' as it contributes to 38.73 % of records and cannot be deleted

UPDATE combined\_data

SET number\_of\_modules = 0

WHERE number\_of\_modules IS NULL;

Output

UPDATE 16306

Query returned successfully in 70 msec.

// Validating all null values in number\_of\_modules are replaced with "Not Assigned"

SELECT number\_of\_modules

FROM combined\_data

WHERE number\_of\_modules IS NULL

// Validating if module completion date is less than patient creation date (Check for Anomalous Dates)

SELECT

patient\_id,

patient\_created\_date,

module\_completion\_date

FROM

combined\_data

WHERE

module\_completion\_date < patient\_created\_date;

Output

No data retrieved

// Calculating the distribution of patient accounts and number of modules with clinic\_group 'Not Assigned'

SELECT COUNT(DISTINCT patient\_id) AS count\_of\_patients, SUM(number\_of\_modules) AS total\_module, clinic\_group, clinic\_title

FROM combined\_data

WHERE clinic\_group = 'Not Assigned'

GROUP BY clinic\_group, clinic\_title

Output:

|  |  |  |  |
| --- | --- | --- | --- |
| **count\_of\_patients** | **total\_module** | **clinic\_group** | **clinic\_title** |
| 5038 | 3111 | Not Assigned | Rehab 101 |
| 1852 | 654 | Not Assigned | Rehab on the hill |

Key Findings:

- Total 6,890 patients with no clinic group assigned

- Total 3,765 modules with no clinic group assigned

// Identify Patients with More Than 24 Modules count

SELECT

patient\_id,

SUM(number\_of\_modules) AS total\_modules

FROM

combined\_data

GROUP BY

patient\_id

HAVING

SUM(number\_of\_modules) > 24;

Output:

|  |  |
| --- | --- |
| **patient\_id** | **total\_modules** |
| 359041 | 28 |
| 425721 | 30 |
| 324984 | 33 |

Key Findings:

- There are 3 parents id whose number of module is more than 24

#4. Create the following visualizations in Tableau (or similar data visualisation tool):

Number of patient accounts created per month

Number of modules generated per month

with filtering per clinic title, year and clinic Group (bonus point for a filter to switch between week/month/quarter)

Solution-#4: Priyanka\_Mittal\_assignment.twbx

#5. To assess the performance of a clinic based on the provided criteria (24 modules to complete a therapy = successful),

Solution-#5:

// Created a view(temporary table) to assess the therapy status per patient with following categories

- Successful: total module completed >=24

- On-going: total module completed >= 1 and < 24

- Not Started: 0 modules

CREATE VIEW patient\_success\_status AS

SELECT

patient\_id,

clinic\_id,

clinic\_title,

clinic\_group,

patient\_created\_date,

SUM(number\_of\_modules) AS total\_modules,

CASE

WHEN SUM(number\_of\_modules) >= 24 THEN 'Successful'

WHEN SUM(number\_of\_modules) = 0 THEN 'Not Started'

ELSE 'On-going'

END AS therapy\_status

FROM

combined\_data

GROUP BY

patient\_id, clinic\_id, clinic\_title, clinic\_group, patient\_created\_date;

SELECT \* FROM patient\_success\_status

LIMIT 10

Output:



Key findings

- 722 patients have successfully completed their modules

- 1568 patients are on-going with their modules

- 16306 patients have not started any modules yet

// Aggregate the success data by clinic to determine the performance of each clinic.

CREATE VIEW clinic\_performance AS

SELECT

clinic\_id,

clinic\_title,

clinic\_group,

COUNT(DISTINCT patient\_id) AS total\_patients,

SUM(CASE WHEN therapy\_status = 'Successful' THEN 1 ELSE 0 END) AS successful\_patients,

SUM(CASE WHEN therapy\_status IN ('On-going', 'Not Started') THEN 1 ELSE 0 END) AS unsuccessful\_patients,

ROUND((SUM(CASE WHEN therapy\_status = 'Successful' THEN 1 ELSE 0 END) \* 100.0) / COUNT(patient\_id), 2) AS success\_rate

FROM

patient\_success\_status

GROUP BY

clinic\_id, clinic\_title, clinic\_group;

SELECT \* FROM clinic\_performance

Output:



// Time to Complete Therapy

//Analyzing the average time it takes for patients to complete the therapy (24 modules) across different clinics.This can help identify clinics where patients complete their therapy faster.

CREATE VIEW clinic\_completion\_time AS

SELECT

clinic\_id,

clinic\_title,

clinic\_group,

patient\_id,

MAX(module\_completion\_date) - patient\_created\_date AS no\_of\_days

FROM

combined\_data

GROUP BY

clinic\_id, clinic\_title, clinic\_group , patient\_created\_date, patient\_id

HAVING

SUM(number\_of\_modules) >= 24;

SELECT

clinic\_id,

clinic\_title,

clinic\_group,

ROUND(AVG(no\_of\_days)) as avg\_completion\_days

FROM clinic\_completion\_time

GROUP BY

clinic\_id, clinic\_title, clinic\_group;

Output:



Key Findings

- clinic\_title 'Rehab on the hill' has the least turn around time i.e 106 days

- clinic\_title 'Rehab on the go"' has the highest turn around time i.e 186 days