B1: Describe Technical Environment

The technical environment used to create the dashboard

PostgreSQL

pgAdmin4

Tableau 2021.4

Notepad ++

B2: Demonstration of Dashboard Functionality

The submission fully demonstrates the functionality of *the* dashboard by using SQL, external csv file, joins alter table for rounding, blending data sources, creating worksheets, dashboards and story board

B3: SQL Scripts

The SQL scripts used to support the creation of the dashboard are

**CREATE TABLE public."hospital data analysis"**

**(**

**"Patient\_ID" text COLLATE pg\_catalog."default" NOT NULL,**

**"Age" integer,**

**"Gender" text COLLATE pg\_catalog."default",**

**"Condition" text COLLATE pg\_catalog."default",**

**"Procedure" text COLLATE pg\_catalog."default",**

**"Cost" numeric,**

**"Length\_of\_Stay" numeric,**

**"Readmission" text COLLATE pg\_catalog."default",**

**"Outcome" text COLLATE pg\_catalog."default",**

**"Satisfaction" integer,**

**CONSTRAINT "hospital data analysis\_pkey" PRIMARY KEY ("Patient\_ID")**

**)**

**TABLESPACE pg\_default;**

**ALTER TABLE public."hospital data analysis"**

**OWNER to postgres;**

**ALTER TABLE patient**

**ALTER COLUMN initial\_days TYPE integer USING ROUND(initial\_days)::integer**

**SELECT "patient"."additional\_charges" AS "additional\_charges",**

**"patient"."admis\_id" AS "admis\_id",**

**"patient"."age" AS "age",**

**"patient"."children" AS "children",**

**"patient"."compl\_id" AS "compl\_id",**

**"patient"."doc\_visits" AS "doc\_visits",**

**"patient"."full\_meals" AS "full\_meals",**

**CAST("patient"."gender" AS TEXT) AS "gender",**

**CAST("patient"."hignblood" AS TEXT) AS "hignblood",**

**"patient"."income" AS "income",**

**"patient"."initial\_days" AS "initial\_days",**

**"patient"."job\_id" AS "job\_id",**

**"patient"."lat" AS "lat",**

**"patient"."lng" AS "lng",**

**"patient"."location\_id" AS "location\_id",**

**CAST("patient"."marital" AS TEXT) AS "marital",**

**CAST("patient"."patient\_id" AS TEXT) AS "patient\_id",**

**"patient"."population" AS "population",**

**CAST("patient"."readmis" AS TEXT) AS "readmis",**

**CAST("patient"."soft\_drink" AS TEXT) AS "soft\_drink",**

**CAST("patient"."stroke" AS TEXT) AS "stroke",**

**"patient"."totalcharge" AS "totalcharge",**

**"patient"."vitd\_levels" AS "vitd\_levels",**

**"patient"."vitd\_supp" AS "vitd\_supp",**

**"hda"."Patient\_ID",**

**"hda"."Age" as "hda\_Age",**

**"hda"."Gender" as "hda\_Gender",**

**"hda"."Condition",**

**"hda"."Procedure",**

**"hda"."Cost",**

**"hda"."Length\_of\_Stay" AS "hda\_LOS",**

**"hda"."Readmission",**

**"hda"."Outcome",**

**"hda"."Satisfaction"**

**FROM "public"."patient" "patient"**

**LEFT JOIN "public"."hospital data analysis" "hda"**

**ON "patient"."age" = "hda"."Age"**

**AND "patient"."gender" = "hda"."Gender"**

**AND "patient"."initial\_days" = "hda"."Length\_of\_Stay"**

**AND "patient"."readmis" = "hda"."Readmission"**

B4: Explain how the Data Streams were prepared

The data streams were prepared to support the analysis by joining the patients table and hospital data analysis table on age, gender and survey/satisfaction questions. Also used were the admission and survey\_responses\_addon tables to create a robust picture of the data.

B5: Describe how the Data Points were aligned

Data points were aligned by using a left join on the patient and hospital data analysis tables to provide all data from the patient table with records that matched from the hospital data analysis table. This did create some null values as expected since the data was joined on age, gender, survey/satisfaction. The other two tables were created via Tableau using relationships with the patient table

B6: Demonstrate Database Creation

The database was created using pgAdmin. There was a base setup for medical\_data supplied. SQL scripts were ran to create the additional table needed. The pgAdmin function for importing data was used to populate the newly created data. Referential integrity was maintained to ensure that there was no corruption of data. An ERD was created to visually see the data.

B7: What is Referential Integrity

Referential integrity is the concept of enforcing rules in the database to ensure that data is not corrupted. By setting up primary keys for each table and ensuring that those cannot be null, we are establishing a path to avoiding data corruption. This property of relational databases ensures the consistency and accuracy of data by enforcing relationships between tables. It ensures that any foreign key value in one table matches a primary key value in another table. Essentially, it prevents "orphan" records—records in a child table that reference a non-existent record in the parent table.