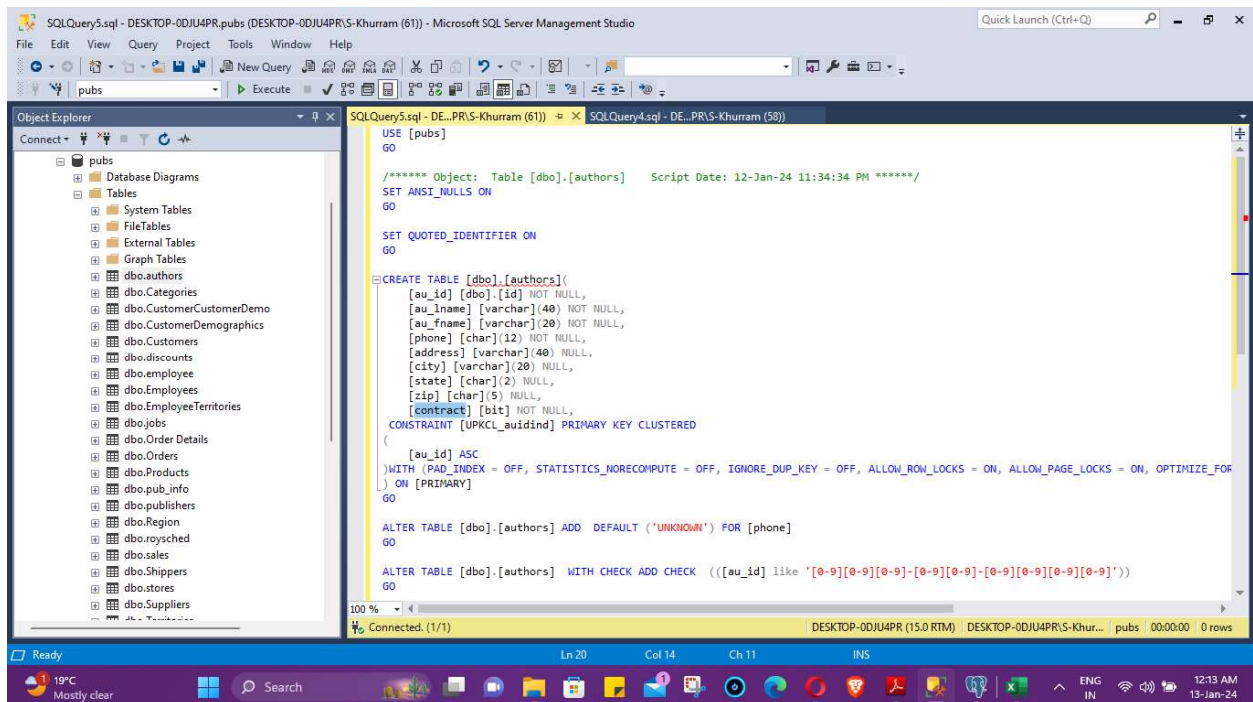
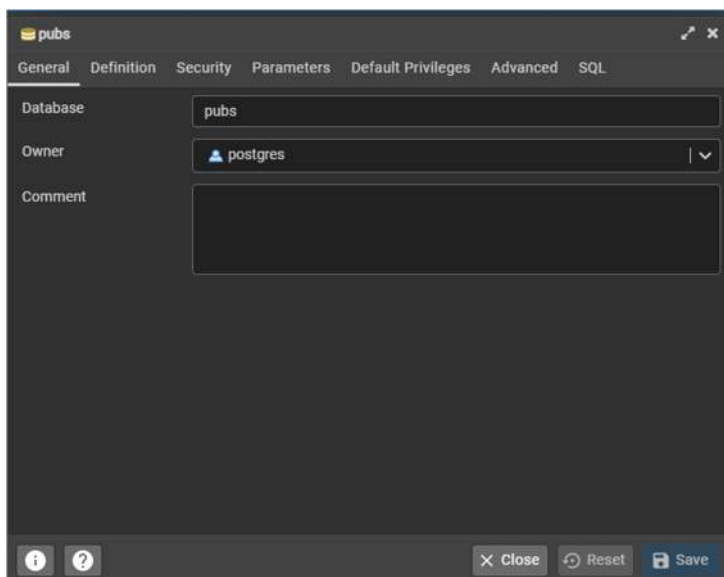


As per the sample database provided, I created the database in sql server using the file provided.

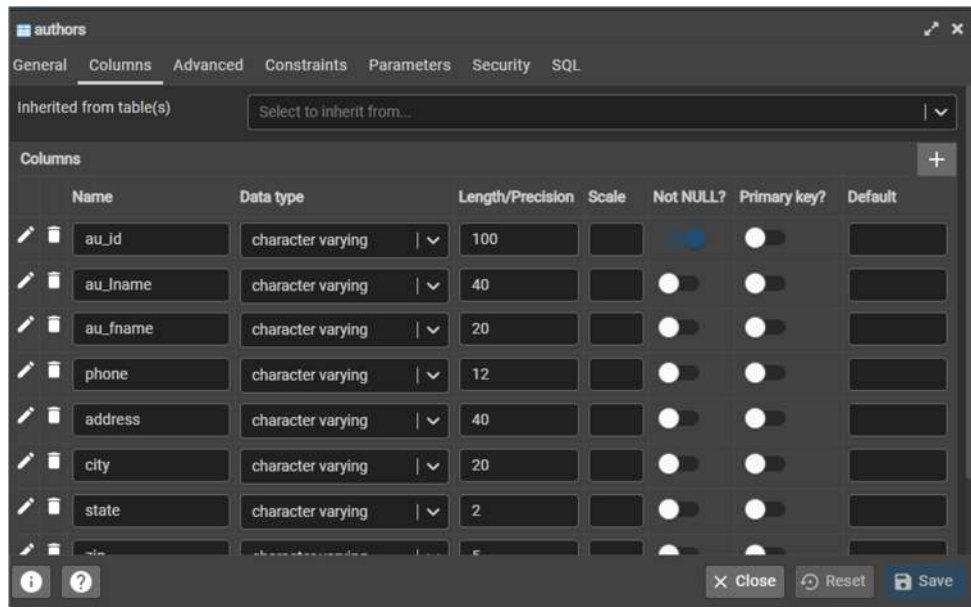


2- Before going for the migration, as this was a cross platform migration between a different database, I did not had any third party tool to facilitate the migration so I decided to go for the export/ import method. I have to create equivalent database in Postgresql.

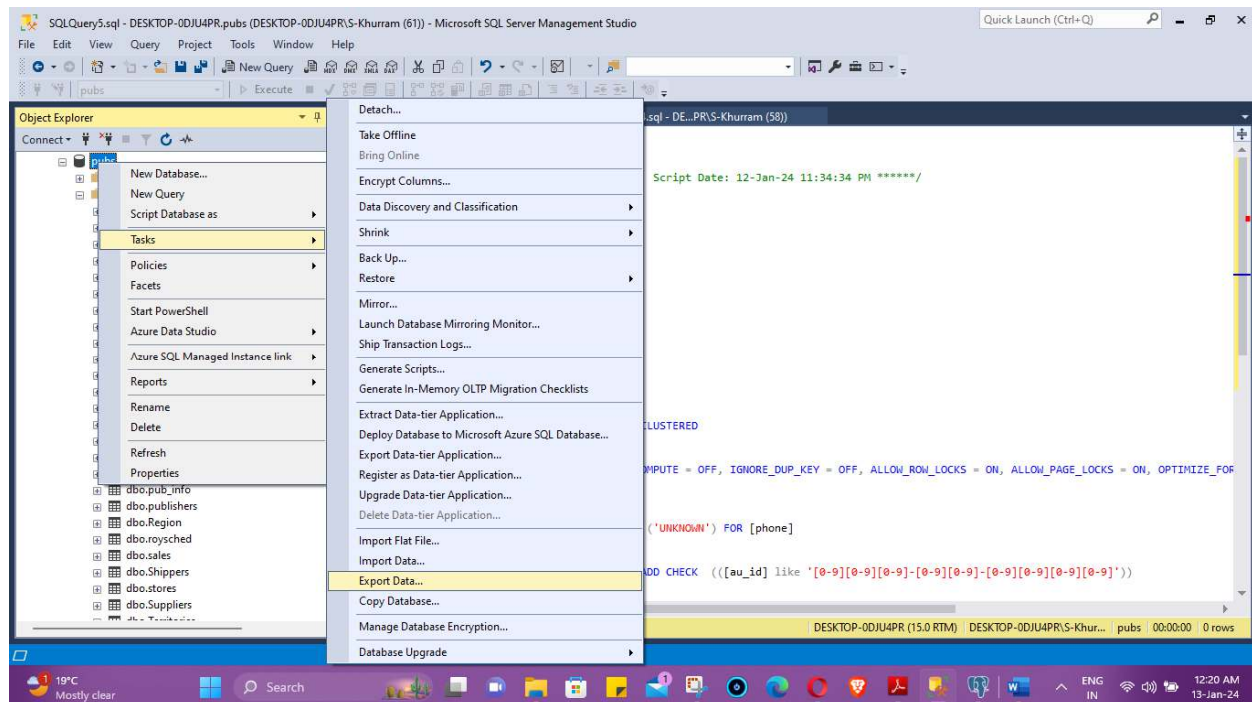
Created the database there



Created equivalent table using equivalent datatype.

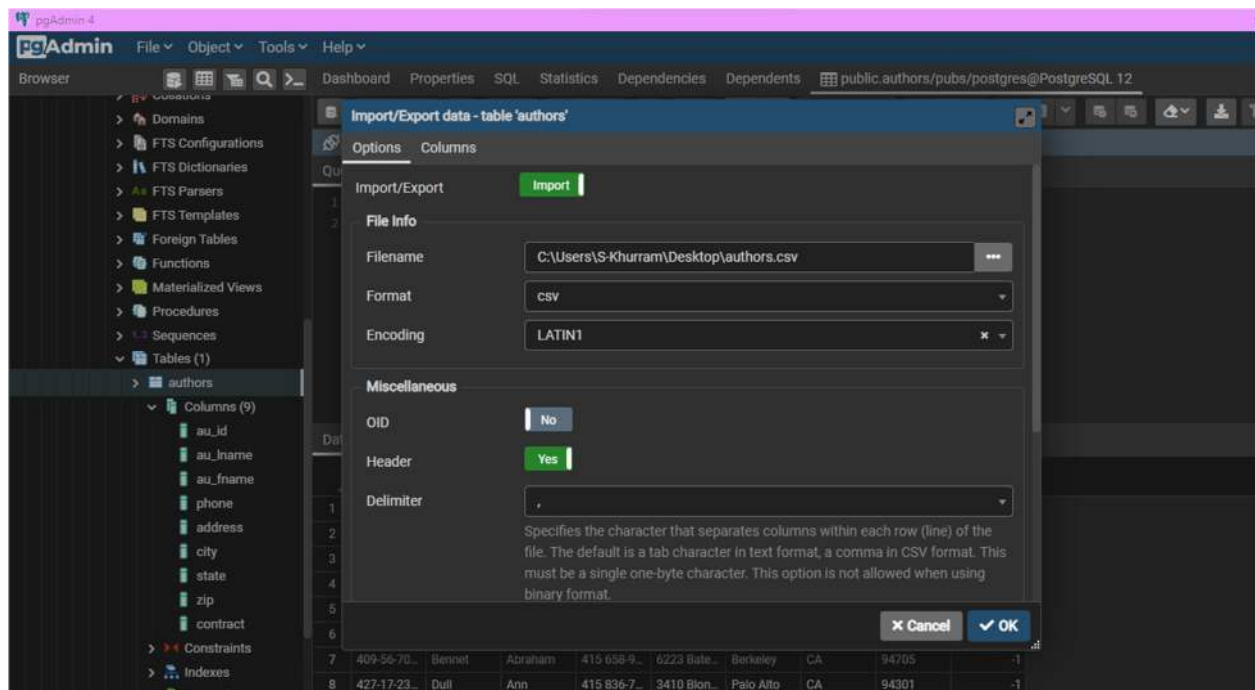


After that I exported the table author from the sql server using the export wizard to a csv excel file.

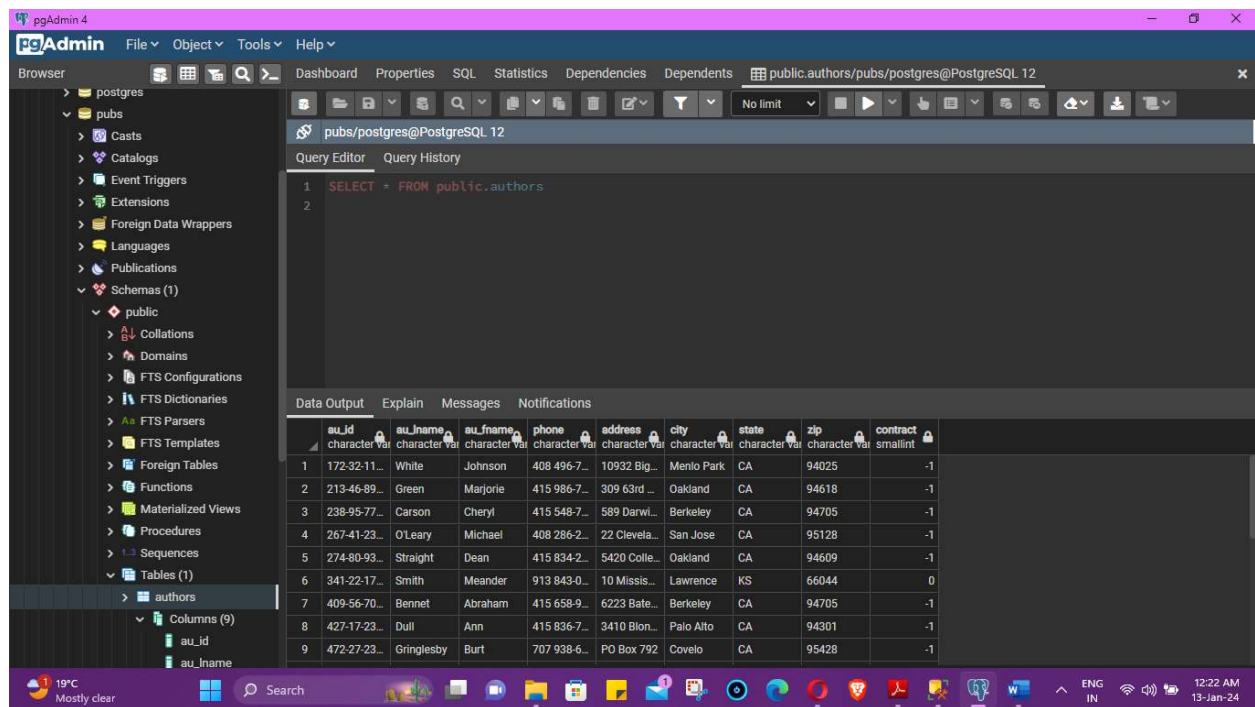


Saved the xls file to csv format. Validated that everything is correct.

Started the import using the import wizard in pgadmin



After that import is successful.



Using the above method each table can be migrated to PostgreSQL server.

3. Any incompatibility needs to be noted and an approach identified for fixing that out.

Answer: Equivalent datatype need to be maintained while creating the equivalent table in the destination postgresql server

Data need to be validated after the migration.

**B) Migration Strategy** 1. A client has an SQL Server database that has SSIS jobs, and a Service Broker configured on it. The database is approximately 10TB in size and grows about 10GB monthly. It currently uses 24-core VCPU and 256GB of RAM under SQL Enterprise Edition 2016. This is a transactional database that has a max downtime limit of 4 hours on special update events.

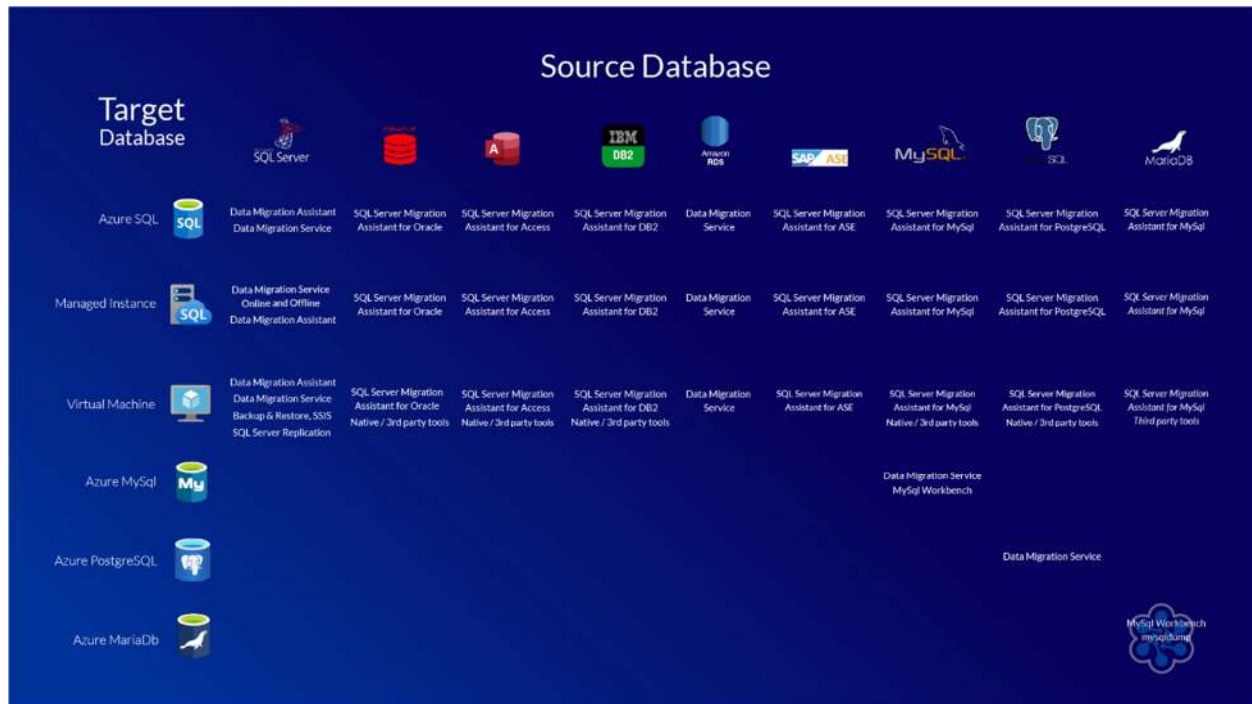
2. What will be the strategy to migrate such a database to PostGRE considering the size and transactional volume? Mention any tooling (open-source or proprietary) that can ease out this process.

Answer: As this is a cross platform migration, following tool can be utilized

- [AWS Schema Conversion Tool](#)
- [AWS SCT Action Code Index](#)
- [AWS Database Migration Service](#)
- [Amazon RDS on Outposts](#)
- [Amazon RDS Proxy](#)
- [Amazon Aurora Serverless v1](#)
- [Amazon Aurora Backtrack](#)

Continue on next page

But in case if it is not a cross platform migration and we are migration from sql to sql, postgresql to postgresql and mysql to mysql then we can utilize below tool.



What can be the issues being faced and possible mitigation plan?

- Answer : size of the database and provided downtime will play critical role in the above scenario. Based on the mentioned factors migration plan will be chosen. Here [AWS Database Migration Service](#) will suggested migration plan.

What will be the roadmap for the transition and what factors will determine the timelines of such a migration?

Answer: key factors will be size of the database : 10TB and provided downtime 4 hours. We should complete all the pre-requisite before heading to migration, do the troubleshooting if any error occurs. Do the validation post migration before handing over to the user.