**B) Migration Strategy**

**Q1:** 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.

**Tools:** There are multiple tools available for migration to PostgreSQL like:

* Sqlserver2pgsql
* AWS SCT Tool
* AWS DMS Tool
* Ora2pg

1. As Database size is huge so first, we have to find-out some SQL server specific features and syntax and this activity should perform in non-business hours because Larger databases may require more time for schema and data migration.
2. Rollback plan
   1. Develop a comprehensive rollback plan in case unexpected issues arise during or after migration.
   2. Regularly back up the PostgreSQL database during the transition
3. (Optional) we can add more space temporarily to perform this activity (Optional but good to have)

**Q2:** What can be the issues being faced and possible mitigation plan?

Below points should consider

1. Mismatch in data types between SQL Server and PostgreSQL.
2. We should carefully analyse the database schema and address syntax differences during the migration.
3. Modify stored procedures, triggers, and functions to align with PostgreSQL syntax.
4. Update all non-supported data types (Example: DATETIME to TIMESTAMP)
5. Remove non-supported optional keywords (Example: WITH RECOMPILE)
6. Remove square brackets (Use double quotes instead)
7. Case sensitivity
8. Replace T-SQL batch terminator "GO" with PostgreSQL ";"
9. Nested Stored Procedures
10. SQL Server Jobs should replace with CRON Jobs
11. Special features like WAIT FOR DELAY needs to be replace with pg\_sleep()
12. To use SSIS packages we should create ODBC connection and using file deployment we can use them
13. SQL Server Stored Procedure returns dynamic result sets based on input like 1 or 2 or any number of columns. (Need to use cursor in Procedure in PostgreSQL)
14. Optimize queries and indexes in the PostgreSQL environment.

**Q3:** What will be the roadmap for the transition and what factors will determine the timelines of such a migration?

Below points we should consider

1. We Must schedule the migration during a low-traffic period, considering the database workload.
2. Generate the assessment report using AWS SCT tool to check out the Action Item and according plan the roadmap for the transition.
3. After analysing assessment and complexity level of database object we can determine the timeline for the migration.
4. First, we should consider independent database object in roadmap to make process fast. Further that we need to consider the parent database object which is referred from other objects for example parent table/view/function etc.
5. Based on the database object complexity we can assign set of objects to database developer for the manual conversion.
6. We should prepare a tracker sheet to Keep track of changes made during the process.