Question 3 (10 points)

You have been tasked to setup a test environment for the API. Part of the task is to

setup the database, so its contents mimic the existing production database. Refer

to the attached ER-diagram. The tables have the following number of records:

• housing – 1.5 million rows

• housing\_type – 10 rows

• ownership – 1.75 million rows

• owners – 1.25 million rows

The test environment will be hosted on OpenShift. A namespace has already been

provisioned. The production database is also deployed in OpenShift, although in a

different namespace. It uses the Crunchy Postgres Operator version 5.4.3 and is

running Postgres version 15.

Describe how you would approach this task

Assumptions

1. I would have admin access to all the environments.
2. There are shared drives accessible between the environments.
3. There is not an existing database backup process taking place.

Step Taken

1. Backup the Production database to shared storage.
   1. The backup process for the database should be planned when the load on the database is low e.g. off peak hours or after hours to not affect service.
   2. The existing storage space should be verified to make sure the database backup won’t have any issues.
   3. The backup should be transferred to shared storage when complete.
2. Restore Production database backup from shared storage to test database.
   1. From the shared space the database should be restored to the test env location.
3. Sanitize the database.
   1. This would be a process or running scripts to remove all sensitive information e.g., phone email, address from the test database.
   2. The scripts could be simple replacement of dummy numbers to more complex scripts changing data in batches with different values.
4. Backup the test database to be used later.
   1. Once the test database has been sanitized and verified a backup should be made to be used if another test environment would need to be setup or if the test database would get corrupted so it can be setup quickly.
5. Put a process in place to test out any schema changes on the test database also.
   1. If in the future any database schema changes which would get applied to production would naturally be tested before going to production.
   2. This step would help ensure that the database in the test environment would be up to date.
6. Have a production database backup process in place for weekly and daily backups depending on the RPO policies.
   1. As an extra step I would discuss having a disaster recovery plan for the database to be in place. This would in the future ensure that the database is backed-up regularly. Saving steps and time if any need arises for the database to be recreated from production.