### Problem statement - Select, Design and Plan High Availability of database for test platform

**Group - 10**

**Scope and purpose**

* **Objectives** – The main objective in designing any High Availability strategy is Business Continuity.

Each business has its own level of tolerance for system failures and outages and depending upon that tolerance, a suitable strategy can be planned and implemented.

High Availability systems in the organizations become more of a requirement from the service standpoint than from the perspective of cost associated with downtime.

Availability of the systems should be seen from the end user's perspective. Any time a user cannot connect to the system is considered as downtime but this does not necessarily mean the main computer system is going down because, in many cases, a poorly performing system is also considered an Unavailable System.

* **Goals** – Any high availability network should have the following technical aspects that includes:

1. Continuous operation - Continuous operations are activities within a business or organization that are ongoing and sustained in the event of a business disruption.
2. Recoverability - This is done in order to ensure that an operation can be recovered in the event of an untimely application or system failure.
3. Timely error detection - Fast detection is another essential component in recovering from a possible unexpected failure.

* **Existing resources –**Decided to work on everything from scratch, from developing our own database to duplicating the data present in it.
* **Tools and management –** Used PostgreSQL for database creating, data duplication, data manipulation, importing the data. Referred already developed projects based on high availability of database to gain more knowledge about the problem statement.

**Structure and design**

**Table of contents**

* **Scope and purpose**

1. Objectives
2. Goals

* Continuous operation
* Recoverability
* Timely error detection

3. Existing resources

4. Tools and management

* **Creating content**
* Research
* **Deliver and test**

**Creating content**

* **Research**

Carried out a research in which we came across the use of copy database wizard and how it works. Studied about its limitations and ways to overcome any obstacle that we would have to face.

Transfer of logins and passwords between instances of SQL server for when a server collapses can also be helpful.

Use case diagrams, sequence diagrams and state diagrams helped us visualize and understand the problem statement better.

Tested codes for transferring and copying servers for a user as well as for duplication of the data on the same and different servers.

**Deliver and test**

Developed a database and imported the data through a CSV file into the schema that was already defined in the database.

Duplicated the existing data present in the table into “dupe\_tables”.

After successful duplication, created all possible test cases to test the system and all test cases were positive.

**Installing PostgreSQL on your local development computer**

For early development, testing, and proof of concepts, installing PostgreSQL on your local development machine can provide reliable, easy to manage access to your databases.

| **Hosting Option** | **Local development machine** |
| --- | --- |
| **Project stage** | Development |
| **Cost** | No additional costs |
| **Performance** | Low |
| **Scalability** | None |
| **Management complexity** | Low |
| **Additional notes** | Does not require network configuration. Good for local development. |

**Cost**

Setting up PostgreSQL on your development machine is free of cost. You are running the database from a computer that would already be active as you develop. You only need to consider the amount of resources that PostgreSQL will consume when up and running.

**Performance**

Installing PostgreSQL on your development machine is a low performance option.

Your database will not be easily or reliably available to other users. Your own usage of the database will be limited by your hardware and the amount of resources you can spare for PostgreSQL. These concerns are not typically an issue when testing or developing locally, but are entirely inadequate for anything more complex.

**Scalability**

Hosting on your development machine gives you almost no scalability. You can change the amount of resources allocated to PostgreSQL, but not much more. You can upgrade your development machine, but that is neither practical or particularly useful in the long term.

**Management complexity**

In terms of complexity, hosting PostgreSQL on your local machine is often fairly straightforward. The installation process for most operating systems is well thought out and the resulting database can be started or stopped easily. Configuring your local PostgreSQL instance for external access is usually not worth the effort however given the resource limitations and consumer network instability.

Though setting up PostgreSQL locally is not complex, you will still be required to manage your database and perform any upgrades as necessary. These might be required for security patching occasionally and it will be your responsibility to keep track of these instances if you are concerned about your data.

**Localhost a PostgreSQL server**

The PostgreSQL database service is available on localhost at the default port 5432. A default user (hosting-db) and database (postgres) exist so you can quickly test your connection and perform management tasks.

Default connection settings:

Host: localhost

Port: 5432

User: hosting-db

Password: <none>

Database: postgres

The URL version looks like this:

tcp://hosting-db@localhost/postgres