**DBAS 3085 Security Audit Project**

* Logical and Physical Issues
* Data Application Security

# Memo

**To:** Bill Cunningham

**From:** Jamie Lu

## Introduction

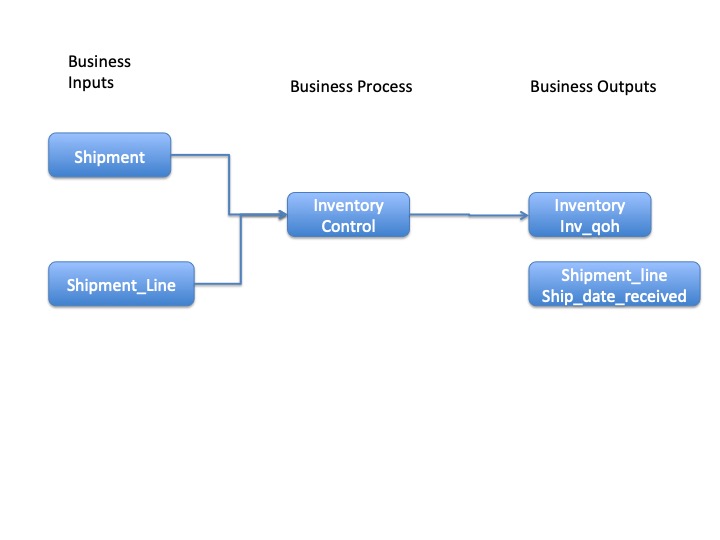
Data Security has become something that is crucial for any business. With technology being integrated more in everyone lives, the use of technology has been common among businesses. Since these applications use and store a lot of data, when it comes to the creation and usage of these applications, they must be constantly be monitored and tested to avoid any future breaches or security vulnerabilities.

With a lot of applications being available over networks or connected to cloud storage, data application security has become more crucial. It is important to do testing as it can reveal what weaknesses are at hand and address them to make things more secure.

Whether it is the physical, or logical security of data, or the data application security, everything must be tested, and we must ensure that these are secured. This protects not only the data, but the users of it as well.

## Existing System

Based on the scenario, this is the existing system of the e-business, Clearwater Traders:



## Statement of Requirements

* Formulate an analysis that will include sections for the topics below. Each section should consist of the observations and conclusions
  + Physical Security of Data
  + Logical Security of Data
  + Data Application Security
* Create a recommendations section that will consist of the following:
  + Implementation plan
    - Based on the conclusions made in the previous requirement
    - What the business should do
    - Should include pseudocode (if applicable)
    - Should include a testing plan for the Data Application Security recommendations, and should have the following headings:
      * Iteration
      * Test
      * Expected Result
      * Actual Result
      * Action Required

## Analysis

Physical:

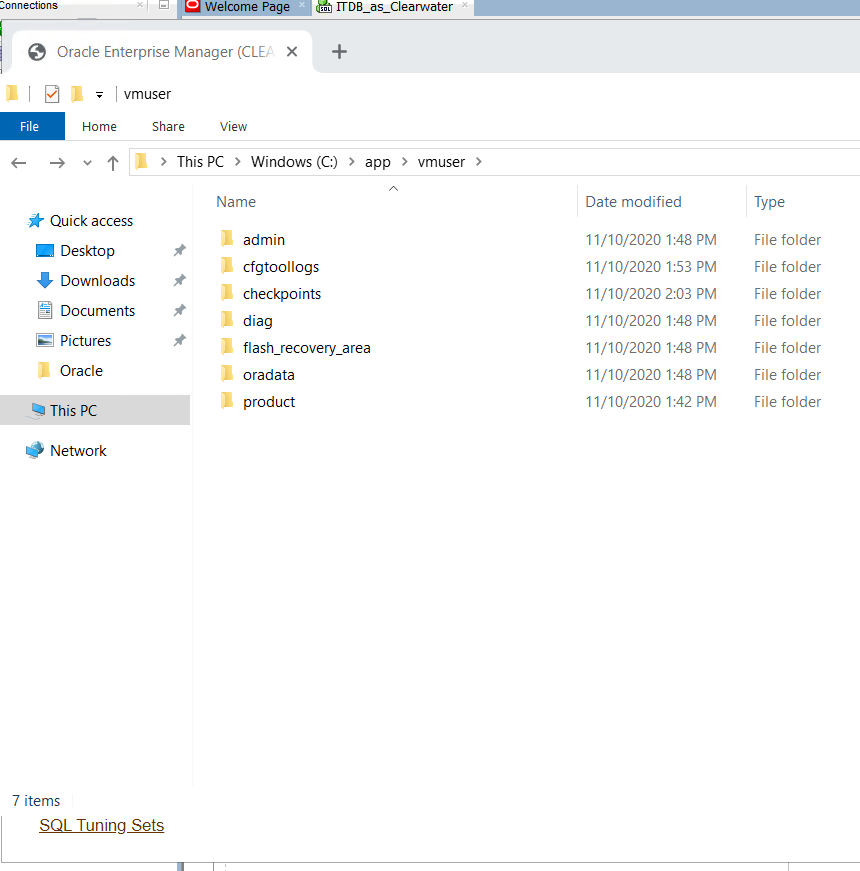
This has to do with the operating system and network.

Physical Issues:

Issue 1:

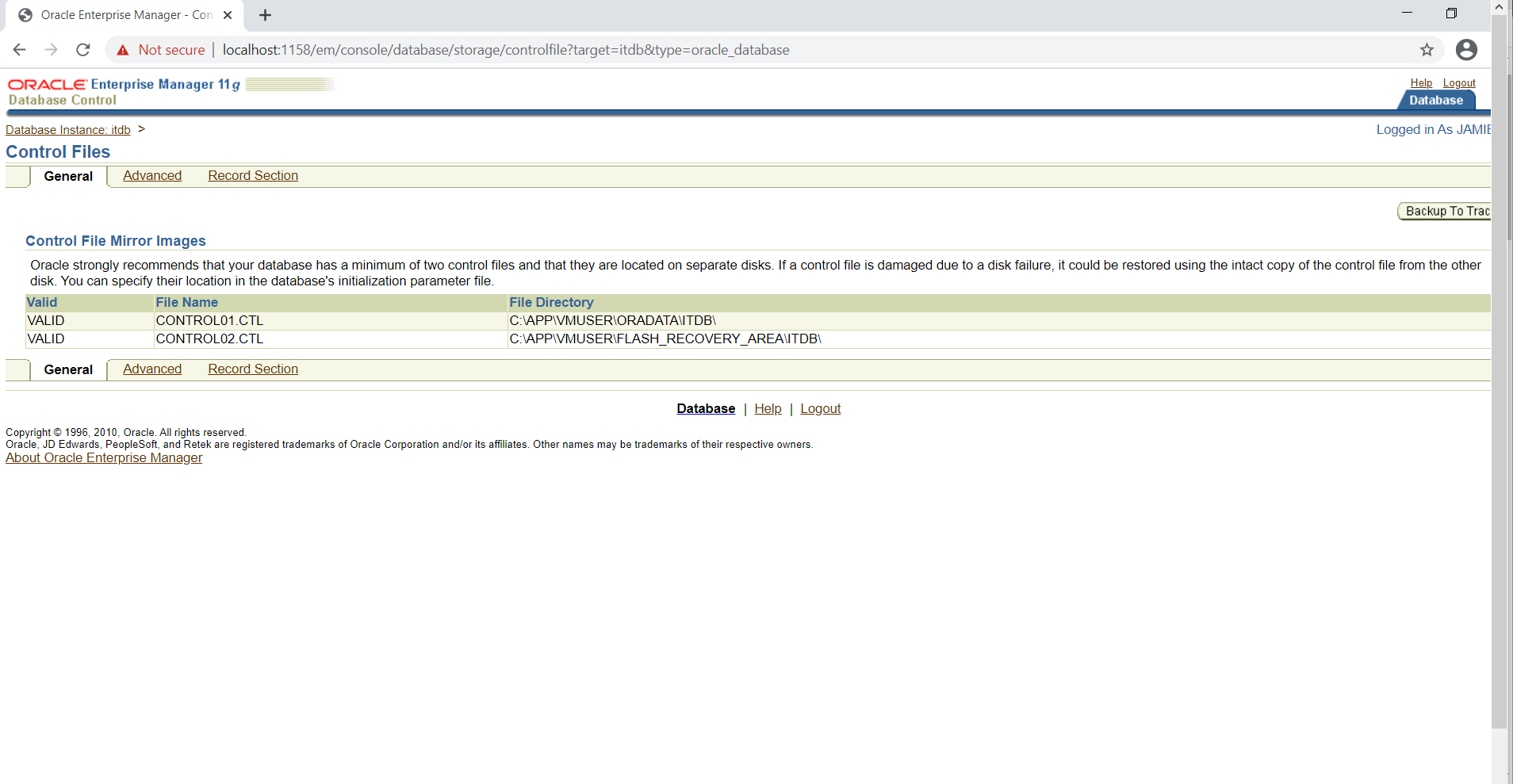
Some platforms like Microsoft will let you set up your Microsoft SQL Server on your Microsoft server as an operating system authentication, it is best not to do that because a hacker can simply access that one account and they would be able to access everything set up on the operating system authentication.

Issue 2:



All the database files are in the same root folder, root C. This is not good because if there is an issue with C or if it gets corrupted and blows up, it will affect all the files.

Issue 3:



This was a default installation and with this, it puts everything on the root of a C drive on a windows machine. This is risky because it if it were mapped to a single physical device then there would be a single point of failure. If the C drive gets corrupted or blows up, then everything is gone.

Control Files – there is only one recovery. This means that we can only bring it back up to the exact redundant copy. Both are also on the C drive which, as discussed earlier, is not advisable because they are both rooted in the same area.

Logical:

This has to do with what the database management system is doing.

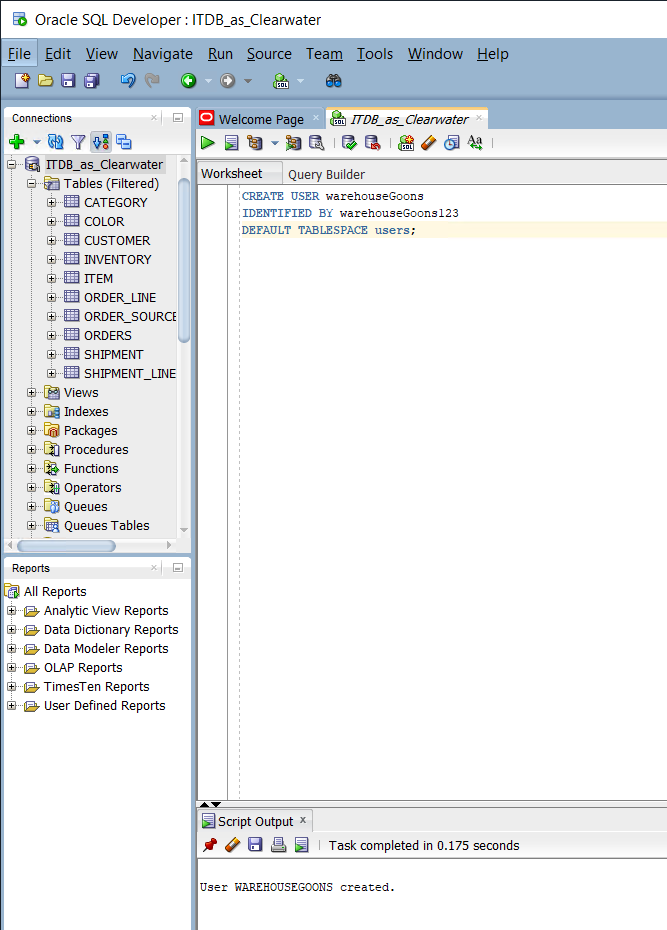
Schemas

Users

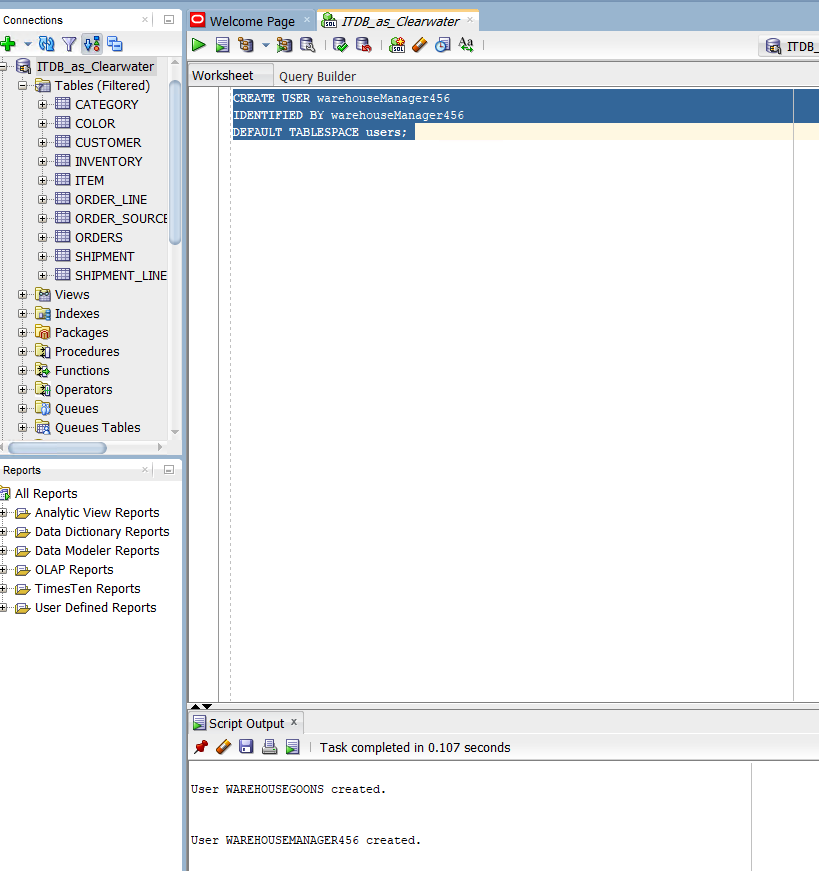
Clearwater role is default. (Check if this is secure role, google)

There are demo users which could be risky as hackers are quite aware of the usernames and passwords.

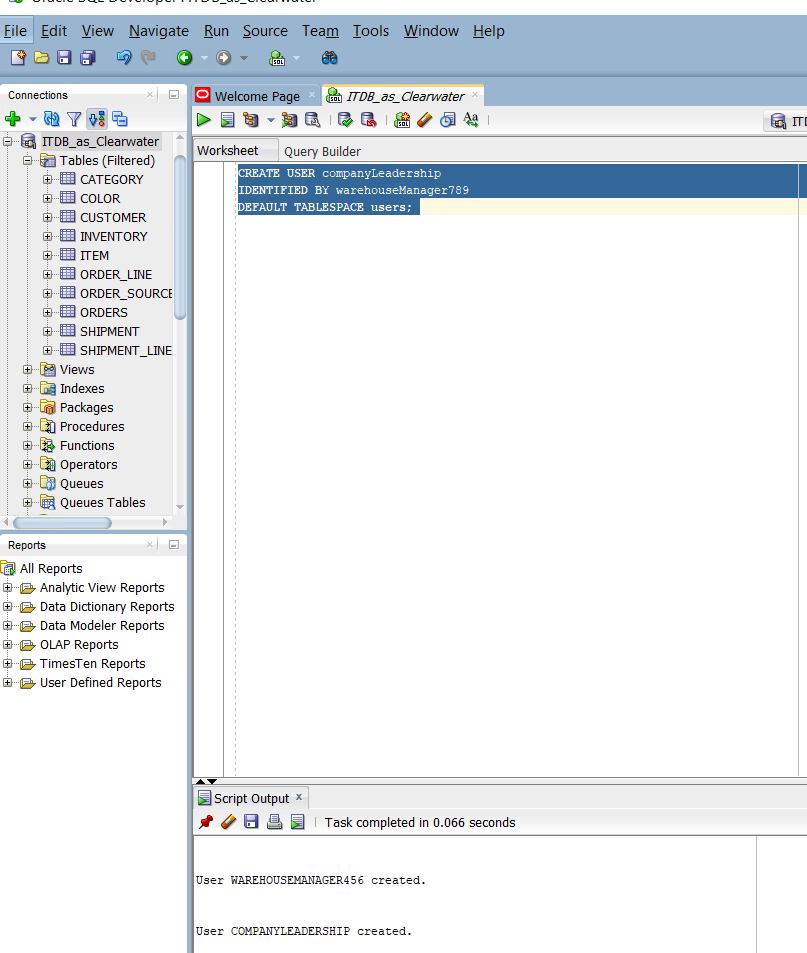
Creating Users:



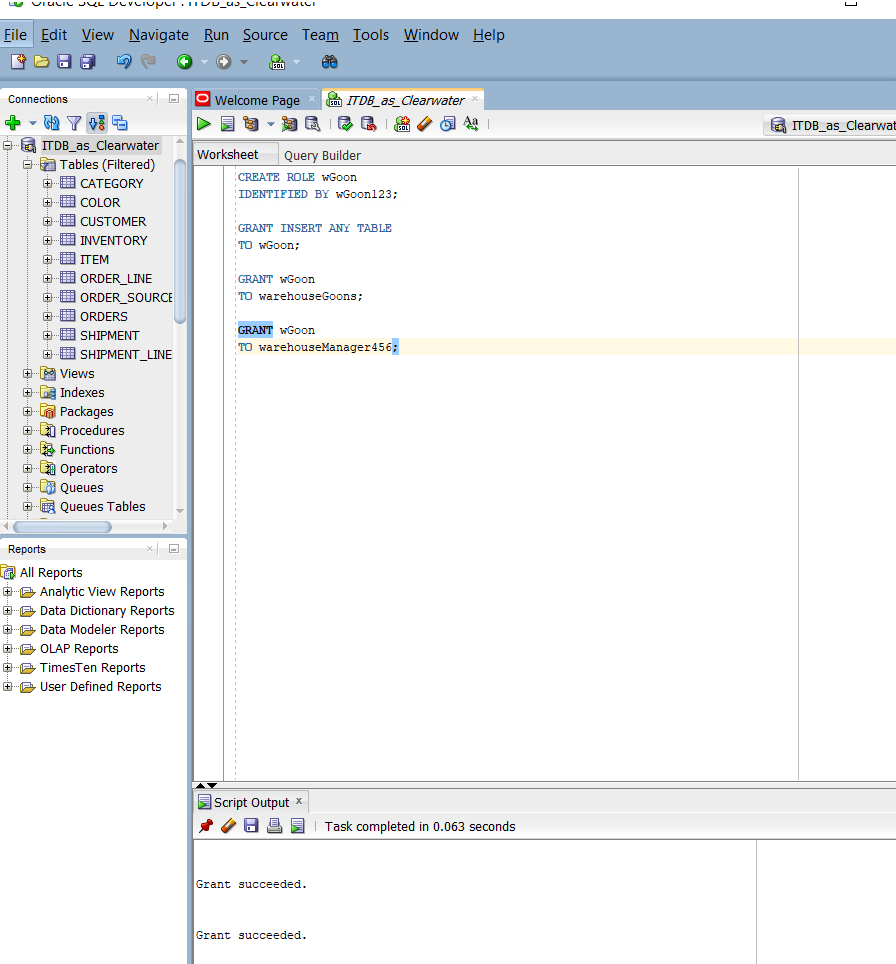
Creating the warehouse goons user.



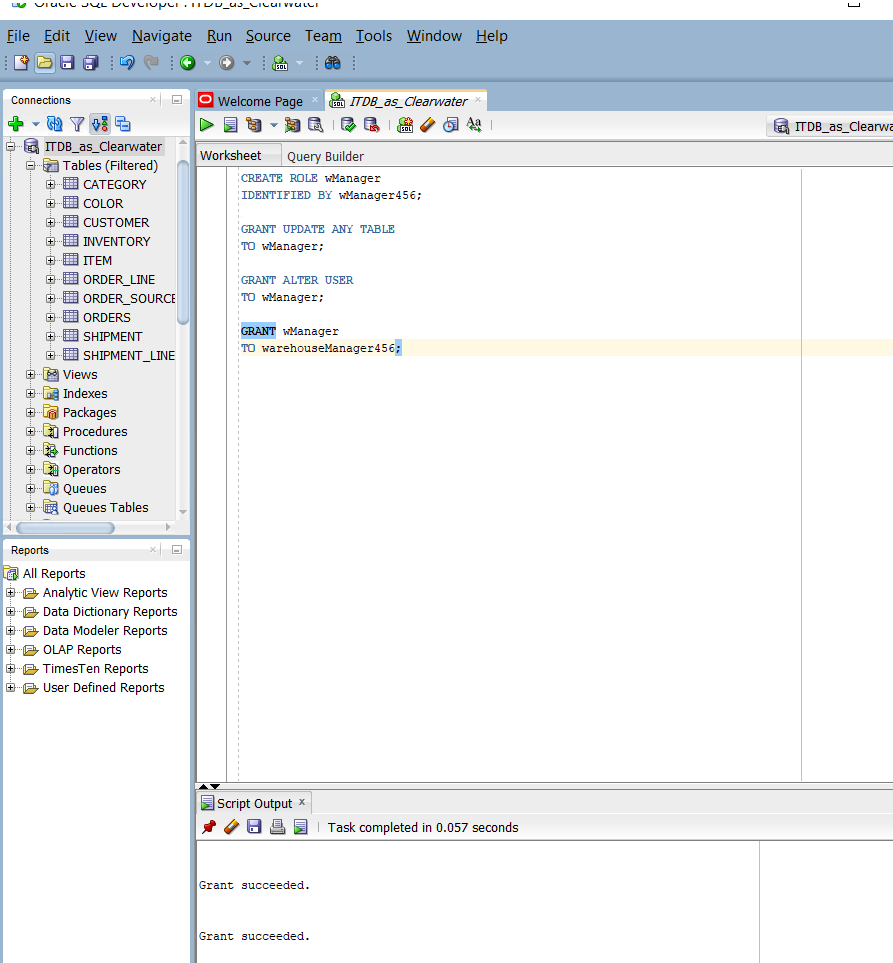
Creating the warehouse manager user.



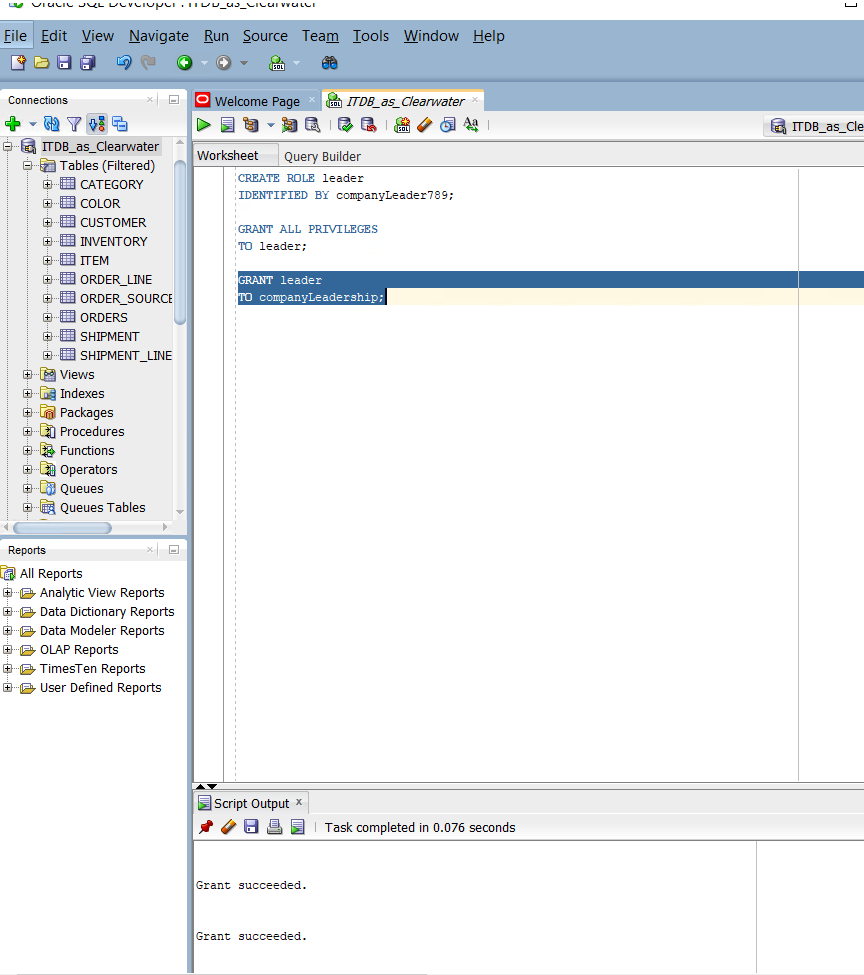
Creating the company leadership user.



Creating wGoon role which has the basic role/s for the warehouse goon, that is why it is being granted to the warehouse goon. It is also granted to the warehouse manager because they have the privileges that the warehouse goons have.



Creating wManager role, granting it privileges, and granting it to the warehouse manager.



Creating the leader role and assigning it all privileges and granting it to the companyLeadership user.

Logical Issues:

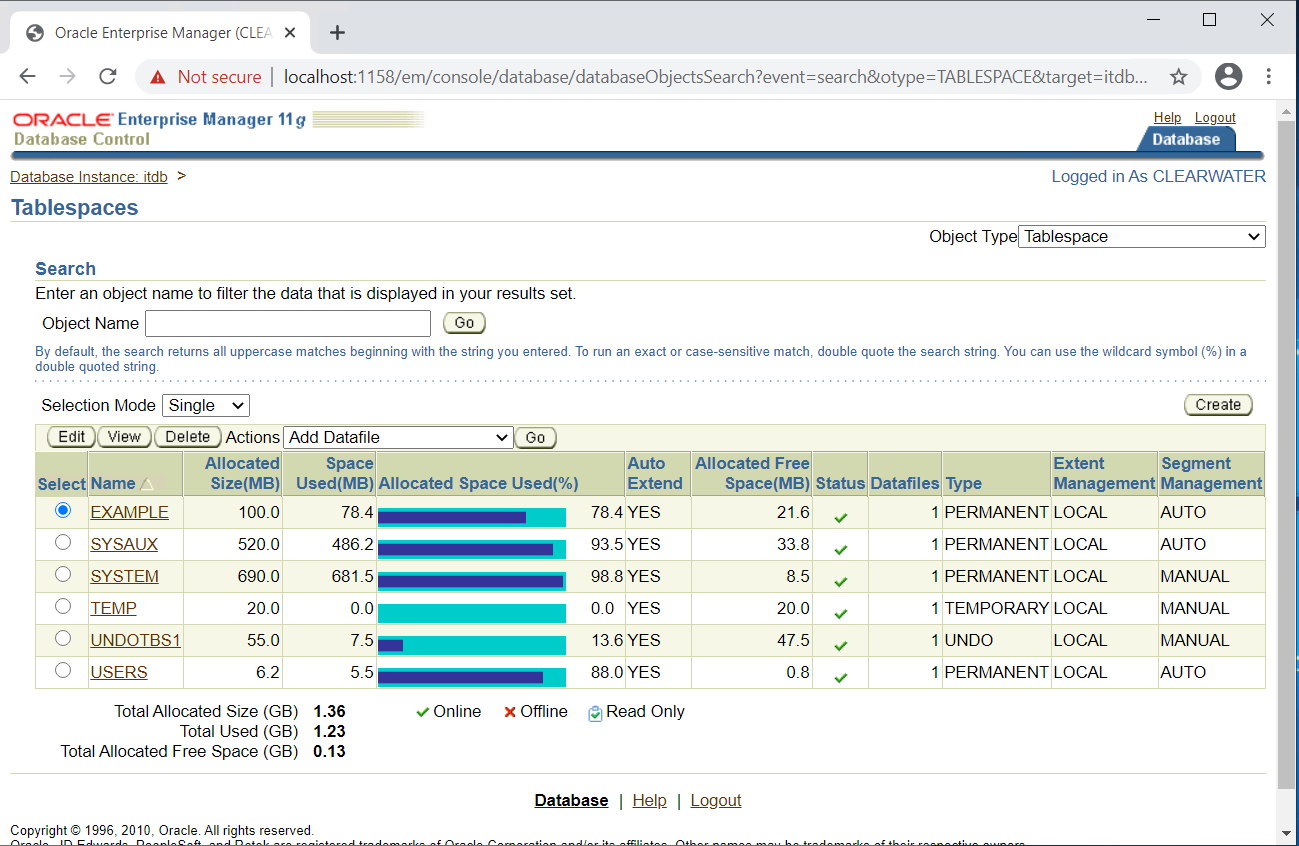
Issue 4:

If demo users were created and installed, there is a chance that hackers are familiar with them and know what the default usernames and passwords are. That is why it is best to check for the demo users and remove them if any. That is also why it is important to chance the SYS password to something that cannot be guessed on the first try.

Tablespaces

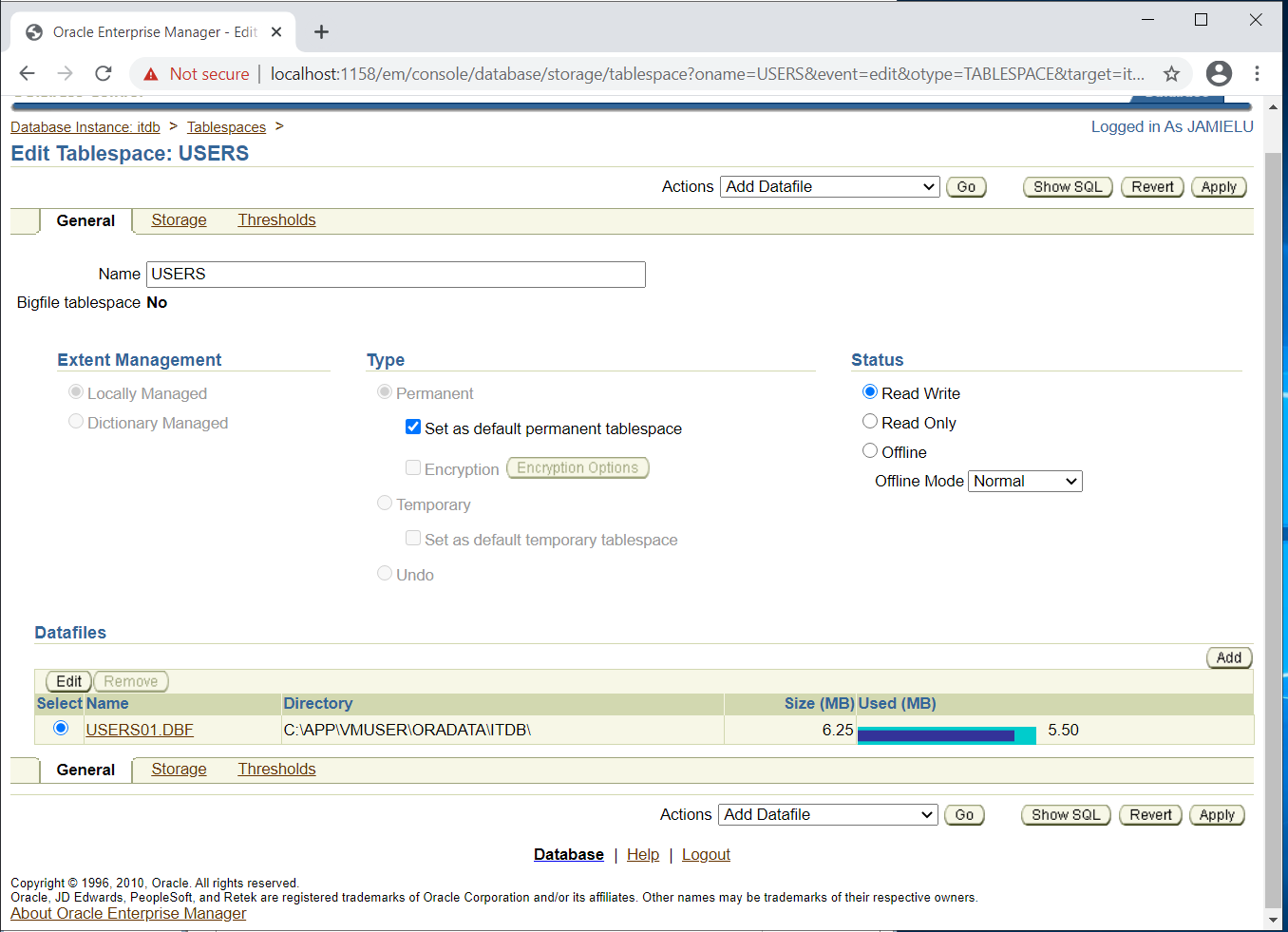
Area in server-side RAM or memory, Logical space that Corresponds to a data file in the physical world.

Issue 5:



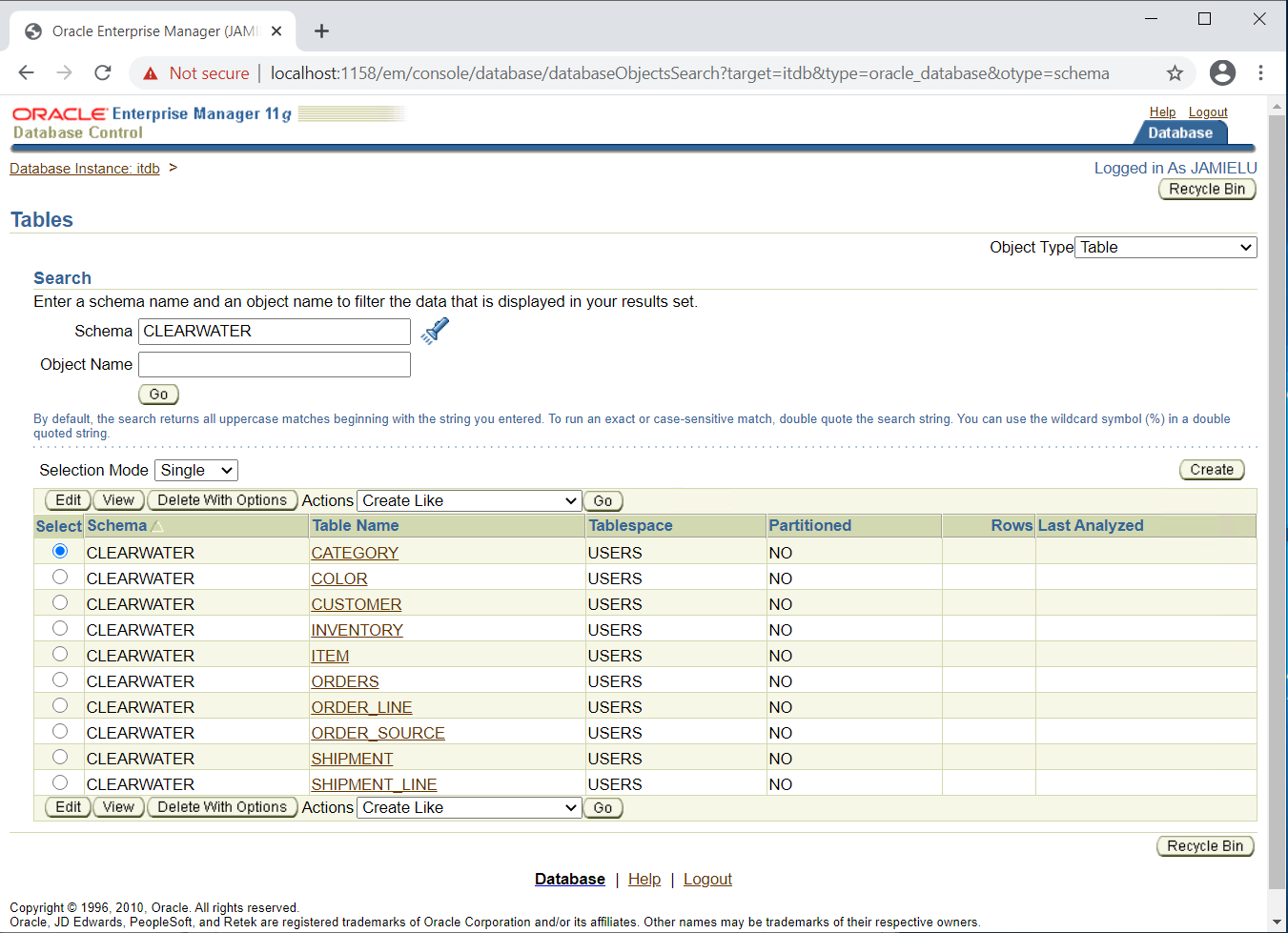
We have to check for tablespaces that are allocating space but are useless. The screenshot above shows that the example tablespace which is taking up about 78.4MB. This is just an example tablespace, so it is useless. The sysaux and system are taking up a lot of space, even one being almost full. However, they are useful as they have to do with the running of the DBMS, this is where the metadata is stored. Temp is for sorting. Undotbs1 is used to store the changes such as inserts, updates, and deletes, so that it could be instantaneously be looked at, or gotten rid of, or transferred over to the permanent data files by the user that put it there entering a commit. The users tablespace is dedicated to the production data.

Issue 6:



It is important to notice that only the users tablespace is dedicated to the data, while all the rest are concerned with the running of the system. This is a problem because the clearwater schema has a predictable password and it is in the same logical tablespace as the other stuff, like the ones used for training, etc. By putting everything in the same logical tablespace, we are taking away server-side resources from the operation of our production objects, and this is not what we want to do.

In the screenshot above, it can be seen that the users tablespace is mapped to just a single data file. This would mean that not only are those other users taking away server-side resources (like mentioned before) like memory from the production objects, but they are also sharing the same physical space, the same single data file. This is problematic because if they picked up a virus then it is going to affect the production data or users tablespace because they are in the same physical data file.



This screenshot shows the tables in the clearwater schema.

## Recommendations:

Recommendations for each issue stated in the previous sections:

Issue 1:

Do not save the password so that whoever is trying to connect needs to input the password.

Issue 2:

It might be best to have several mount points, distribute things across several physical storage devices. One thing that can be done is implement RAID 5. It is good to use the operating system to physically distribute the database files across the network while maintaining the flexible architecture using logical mount points.

Issue 3:

To mitigate the issue of having only one point of failure, we can periodically carry out backups to trace. Although it is important to note that this is not a perfect solution to the issue, but it can certainly help.

Issue 4:

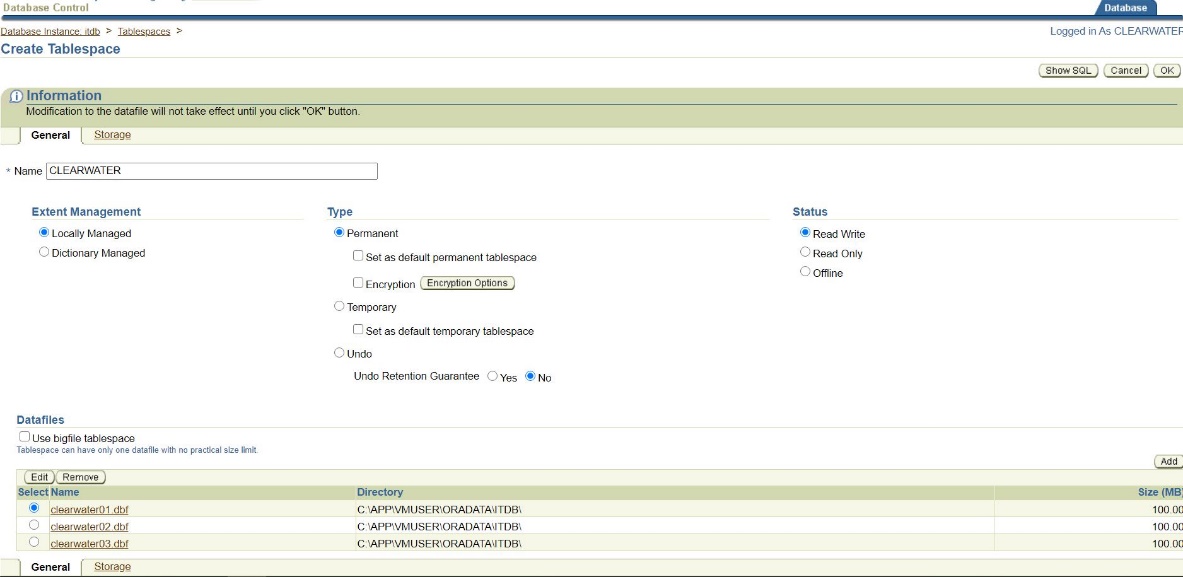
When installing production database, install it on a fixed or static IP address rather than something generated by DHCP.

Issue 5:

It might be best to get rid of the useless tablespaces so that it doesn’t take up the space.

Issue 6:

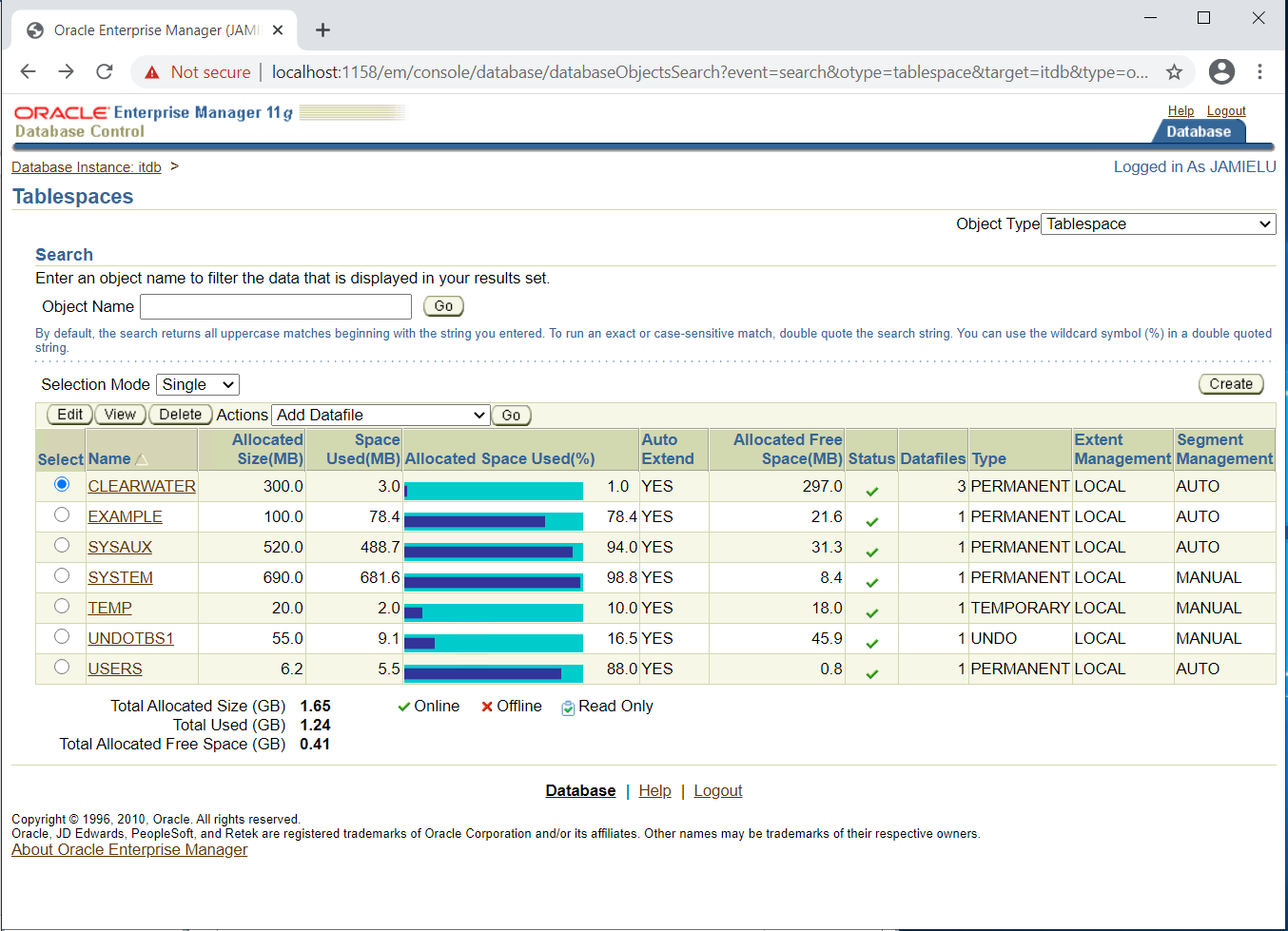
To mitigate this, we can create a new tablespace.



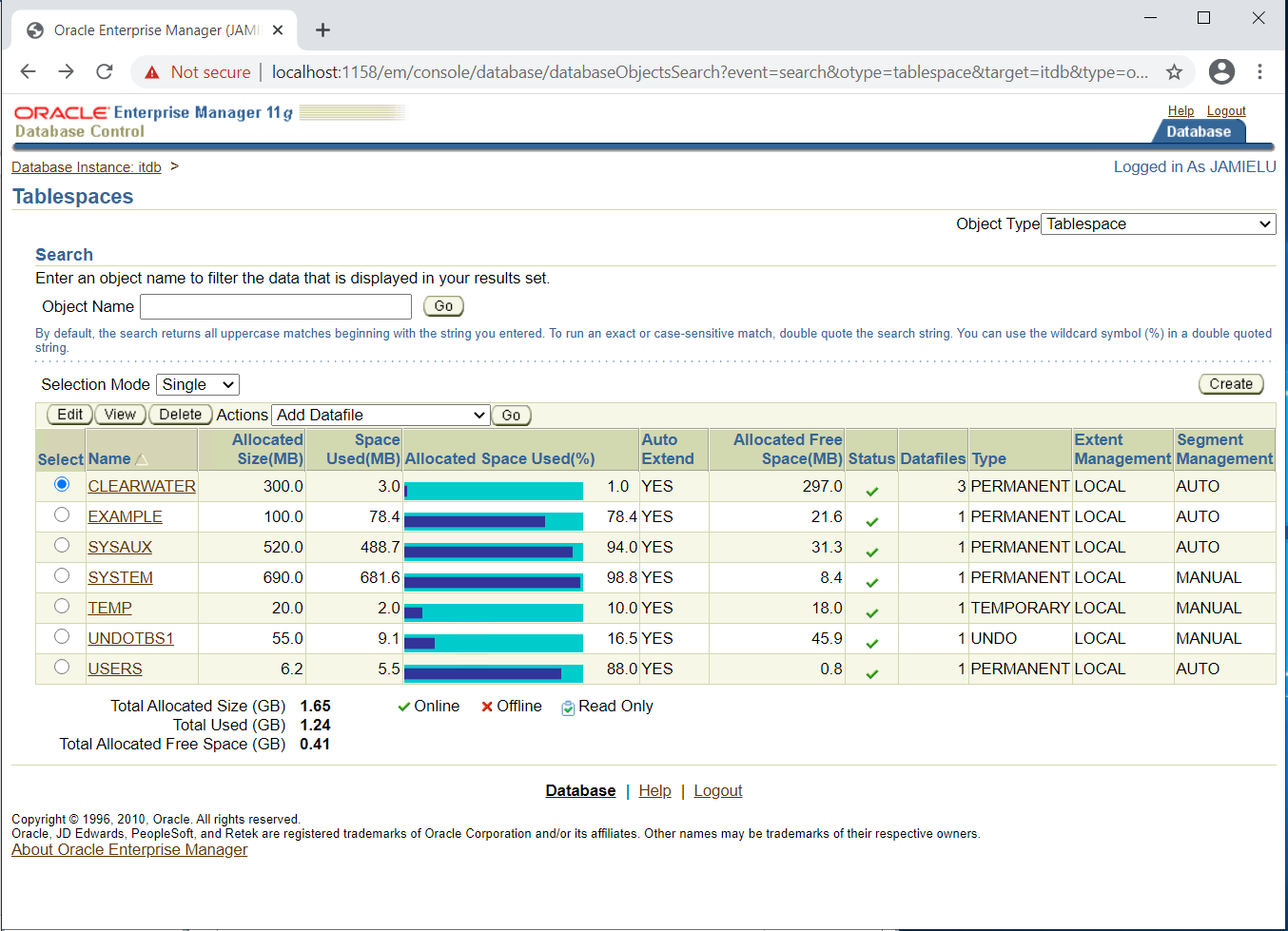
By doing this, we can create something similar to RAID 5. One benefit of this is that we will have a faster read and write time, which is good to use with transaction processing databases.

Below is the pseudo code for the table space I want to create for our production objects so that we can physically and logically segregate our production objects from the thing that normally exists in the public access users tablespace.

**CREATE** SMALLFILE **TABLESPACE** "CLEARWATER" **DATAFILE** 'C:\APP\VMUSER\ORADATA\ITDB\clearwater01.dbf' **SIZE** 100M **AUTOEXTEND** **ON** **NEXT** 25M **MAXSIZE** **UNLIMITED** , 'C:\APP\VMUSER\ORADATA\ITDB\clearwater02.dbf' **SIZE** 100M **AUTOEXTEND** **ON** **NEXT** 10M **MAXSIZE** **UNLIMITED** , 'C:\APP\VMUSER\ORADATA\ITDB\clearwater03.dbf' **SIZE** 100M **AUTOEXTEND** **ON** **NEXT** 10M **MAXSIZE** **UNLIMITED** **LOGGING** **EXTENT** **MANAGEMENT** **LOCAL** **SEGMENT** **SPACE** **MANAGEMENT** AUTO



The screenshot above shows that the clearwater tablespace has been created with 300 allocated size.



The screenshot above shows the details of the clearwater tablespace.

## Conclusion

Security is not a joke, most especially if it involves data as data nowadays is considered more valuable than even oil or gold. Data is knowledge and knowledge will always be power. This is why it is essential to familiarize and if possible, even become well-versed with physical, logical, and application data security.

It is important to do a check up of a company’s existing system. This also must be done multiple times to continually ensure a system’s security. Multiple tests must be done as when checking the system to ensure that during execution, it is still secured. Once this is done, the issues found must be resolved as much as they can, and an implementation plan must be made and executed.

# References

Fearn, N. (2018, June 05). *Application security more important than ever*. Retrieved November 20, 2020, from Computer Weekly: https://www.computerweekly.com/feature/Application-security-more-important-than-ever#:~:text=To%20protect%20users%20and%20data,fix%20and%20prevent%20security%20vulnerabilities.