**INTODUCTION:**

Now you have the Vm open you need to do the following:

1. Go to Applications/ Oracle Dev days/ Enterprise Manger/startup this may take a few minutes so be patient.
2. Go to Applications/ Oracle Dev days/ Enterprise Manger/open OEM this may take a few minutes so be patient.

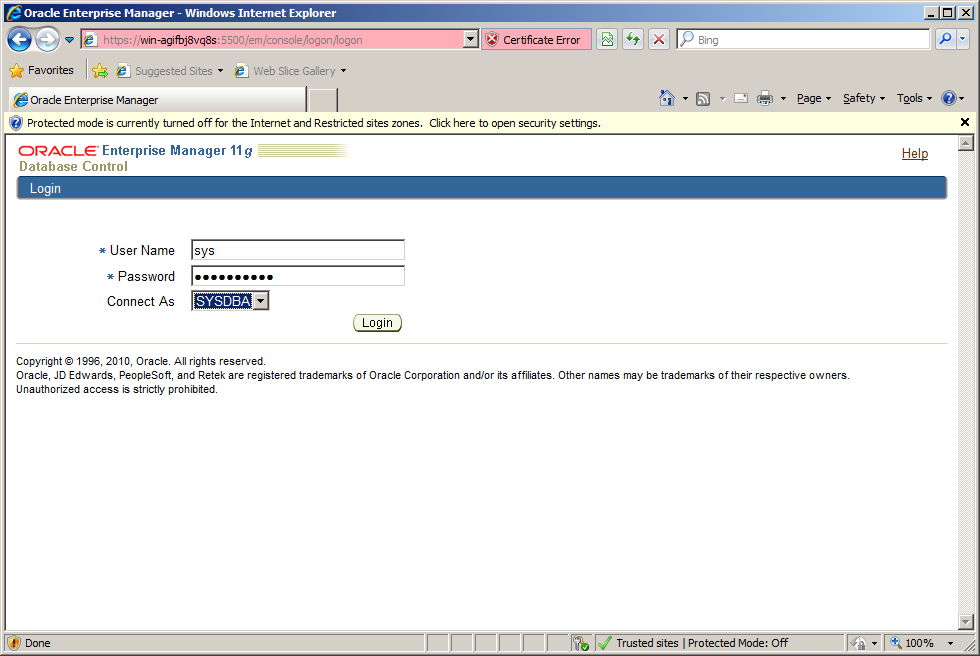
Username : sys

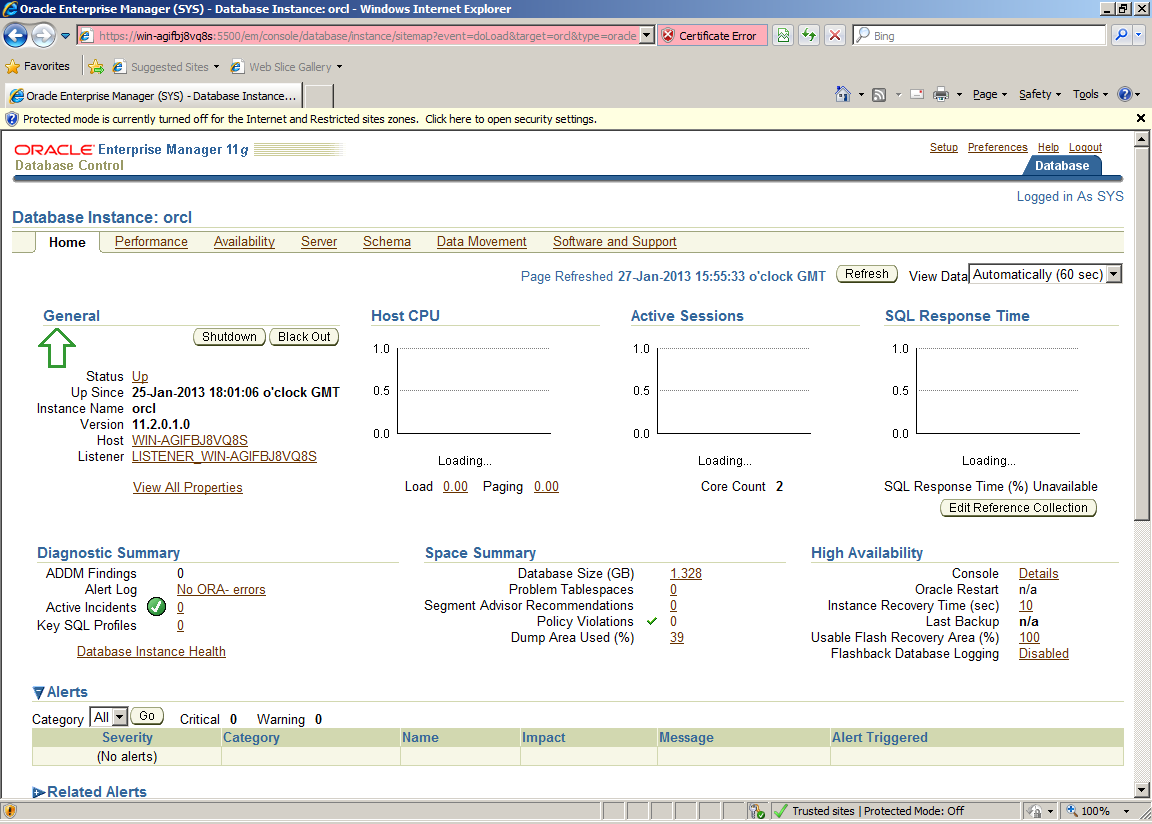
Password : oracle

Connect as: SYSDBA

Then click on login.

Refresh your browser and make sure to connect as SYSDBA as below:





When you see the screen above you are in Enterprise Manager the Database Administration environment.

**Lab: Managing Database Storage Structures**

**Background:** You need to create a new tablespace for an INVENTORY application.

When finished, please upload your answers to moodle.

1. Using Enterprise Manager, Login as **sys** *User* *Name* and **oracle** as Password, and connect as SYSDBA.
2. Using Enterprise Manager, view information about the EXAMPLE tablespace by selecting **Server** > **Tablespaces**.

Answer the following questions about it:

a) Click the EXAMPLE tablespace name.

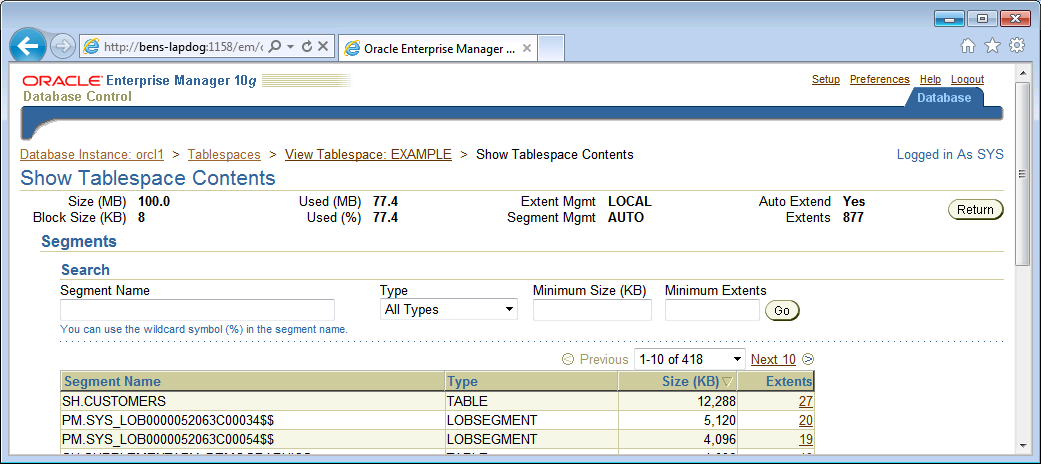
|  |
| --- |
| ***Question 1:***What percentage of free space can be used up before the Warning threshold is reached?  **85%**  ***Question 2:***What percentage of free space can be used up before the Critical threshold is reached?  **97%** |

1. From the **Actions** drop-down list, select **Show Tablespace Contents**, and then click **Go**.

Text

Description automatically generated

1. The Show Tablespace Contents page appears.



|  |
| --- |
| ***Question 2:***  How large is the tablespace?  **82 MB**  How many segments are there in the EXAMPLE tablespace? List ALL seven of the segment types.  **355 Segments**  **Table, LobSegement, Table partition, Index, index partition, Lob Index, nested table,**  How many extents does the SH.CUSTOMERS Table (data) segment contain?  **27 Extents**  How many Oracle blocks make up this Table segment and what is the overall size in KB?  **Size = 12,288**  **Blocks = 12,288 /8 (size of block) =1536 blocks**  How many extents are there in the tablespace?  **789 Extents** |

1. Select INDEX from the **Type** drop-down list in the Search region, and then click **Go**.

Graphical user interface, table

Description automatically generated

|  |
| --- |
| ***Question 3***  Which index in the EXAMPLE tablespace takes up the most space, ensure you move through all the pages?  **SH.Customers\_PK with a size of 1024KB**  How many extents are used to store the index you have identified?  **16 Extents**  How many blocks are used for an extent ( click on the number of extents for the segement)?  **8 Blocks used for an Extent.** |

1. Scroll to the bottom of the page, and then click the plus icon to the left of the Extent Map label. After several seconds, the extent map appears. Note that the map legend indicates that pink is the tablespace header.

Table

Description automatically generated

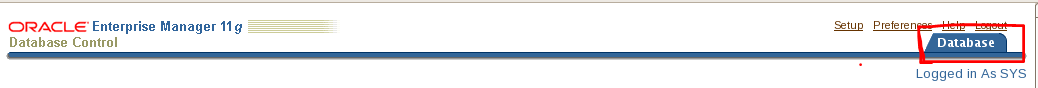
1. Scroll back to the top of the page, select **All Types** from the **Type** drop-down list, and

then click **Go**.

1. Go back down to the Extent map and click the first used extent just to the right of the tablespace header extent on the map.
2. Scroll to the top of the page again, and note the segment that is being pointed to:

|  |
| --- |
| ***Question 4:***  Which segment (and segment type) is stored first in the tablespace? That is, which one is stored right after the tablespace header? Include an image here.  **PM.SYS\_LOB0000073986C00042$$ - Lob segement**  How many extents does it take up?  **7 Extents** |

1. Now let us create a new, locally managed tablespace (LMT) called INVENTORY of size 5 MB. Go Back to the main page by clicking on the Database Tab on the top right hand side of the window.



1. In Enterprise Manager, select **Server** > **Tablespaces**
2. Click **Create**. This is over to the far right, if you cannot see it maximise your screen size.
3. Enter INVENTORY as the tablespace name, and verify that **Extent Management** is **Locally Managed**, **Type** is **Permanent**, **Status** is **Read Write**, and that **Use Bigfile tablespace** is **not** selected.
4. Click **Add** in the Datafiles region. Again to the very right so maximise screen if needed.
5. On the Add Datafile page, enter inventory01.dbf for **File Name**, and 5 MB as **File Size**.
6. Click **Continue**.
7. Click the **Storage** tab, and verify that **Extent Allocation** is **Automatic**, **Segment Space Management** is **Automatic**, and **Logging** is enabled.
8. Click the **General** tab.
9. Click **Show SQL** to see the SQL that will be run, and then click **Return**.

|  |
| --- |
| ***Question 5:***  What SQL is generated?  **CREATE SMALLFILE TABLESPACE "INVENTORY" DATAFILE '/home/oracle/app/oracle/oradata/orcl/inventory01.dbf' SIZE 5M LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO**  Insert a screen shot of the successful update message. |

1. Click **OK**, and a successful Update Message appears.
2. Follow the instructions below to, as the SYS user, run the lab\_script1.sql script to create and populate a table (called X) in the INVENTORY tablespace. What error do you eventually see?
3. Download the folder labscripts from moodle.
4. Open a SQL DEVLOPER from main desktop in the VM.
5. Login as **SYS** (user name) with the **ORACLE** password and make sure you connect AS **SYSDBA** role and connection type needs to be **LOCAL/BEQUETH**. Connection Name can be anything so use **LAB1.**

Graphical user interface, text, application, email

Description automatically generated

1. Click the test button and then connect if test is successful.
2. Open the labscripts file on your machine and copy the code. Paste the script code into the worksheet window and execute the script ) use the second run button with the page behind it)
3. Read and make sure you understand this SQL script such that you could write it yourself.

Describe how this SQL works and what it is doing in here. Give a detailed explanation. One line will not suffice.

This code creates a table called x with one character column called A. This table is created inside the tablespace Inventory that we created earlier in the lab. The value A is inserted into the table. Then all the data from the table is copied and re-inserted and this is repeated several times thus doubling the data in the table each time. The inserts are then committed to the table.

T

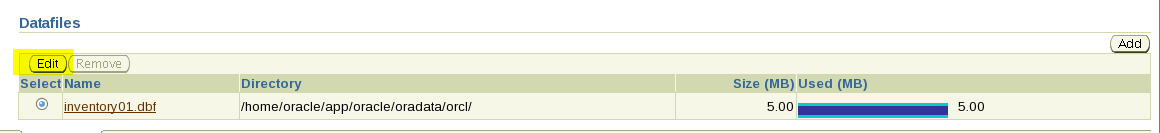
1. Note that there is eventually an error ORA-01653 stating that the table cannot be extended. There is not enough space to accommodate all of the rows to be inserted.

Insert a screen shot of the error you got from SQL.

Text, letter

Description automatically generated

1. Go back to the Enterprise Manager window and define space for 50 MB in the INVENTORY tablespace instead of 5 MB, while keeping the same single data file in the tablespace, follow the instructions below.
2. Select **Server** **> Tablespaces**.
3. Select the INVENTORY tablespace, and then click **Edit**.
4. In the **Datafiles** region, click **Edit**.



1. Change **File Size** from 5 MB to 50 MB.
2. Click **Continue** to return to the General tabbed page.
3. Click **Show SQL** to see the SQL that will be run. Note he SQL below and then click **Return** and then click **APPLY.**

|  |
| --- |
| ***Question* 6:** What is the ALTER statement that is executed to make this change?  ALTER DATABASE DATAFILE '/home/oracle/app/oracle/oradata/orcl/inventory01.dbf' RESIZE 50M |

1. Now in the SQL Developer window execute the following **drop table x purge**; (note purge clause results in the space being released back to the tablespace for use by other objects and the space continues to count towards the user’s space quota).
2. Run the previous script again lab\_script1.sql and note the result.
3. Examine the INVENTORY tablespace.

|  |
| --- |
| ***Question 7*:**  How many extents does the X table use? (Remember show tablespace contents to see this)  **20 Extents**  How many blocks are used?  **Size = 5,120 KB**  **Blocks = 5120/8 = 640 blocks**  What do you notice about some of the extents and the no. of blocks they contain?  **They increase in size to 128 for the last 4 blocks due to the large amount of data being added.** |