Order Database Implementation on Linux using Oracle 19c

Hanif Lumsden

University of Maryland Global Campus

DBST 670 9040 Database Administration

Dr. Igwe

11/15/2022

Table of Contents

Introduction

Order management structures aid businesses in processing transactions made by consumers from various sales channels. In-store exchanges and e-commerce means such as online shopping comprise the transactions of the order database (Amula, 2020). Order database management systems track people, processes, and suppliers entailing the administration of business processes related to the company's order (Amula, 2020, para. 1). The oracle database administrator will locate for their client six preexisting order management tables, and create four relevant ones that will complete the order database. The database will undergo data definition and manipulation language,

Database Planning

Project Plan

The database administration project plan follows five phases: project creation and planning; database instance creation; capacity planning, tables, and tablespace creation; populate tables with SQL*Loader; and database monitoring. Below is the gantt chart that coincides with the project plan schedule. Copies of the project plan schedule can be accessed here and gantt chart accessed here.

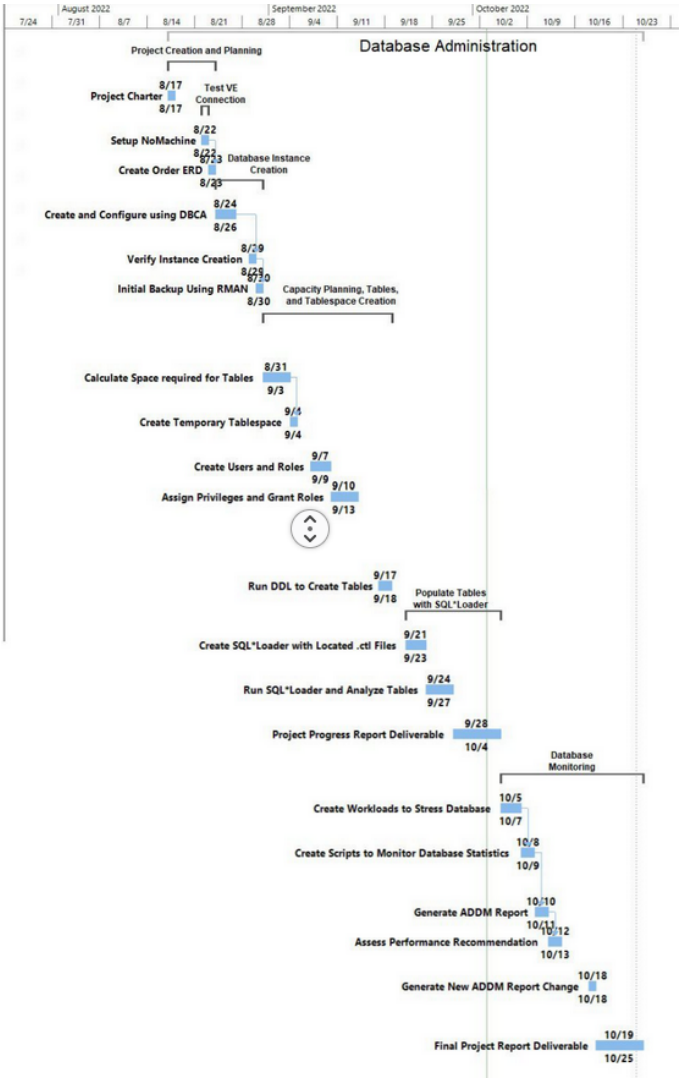
Figure 2

Project Chart

	Task Mode	Task Name	Duration	Start	Finish
0		**Database Administration**	**70 days**	**Wed 8/17/22 8:00 AM**	**Tue 10/25/22 5:00 PM**
1		**Project Creation and Planning**	**7 days**	**Wed 8/17/22 8:00 AM**	**Tue 8/23/22 5:00 PM**
2		Project Charter	1 day	Wed 8/17/22 8:00 AM	Wed 8/17/22 5:00 PM
3		**Test VE Connection**	**1 day**	**Mon 8/22/22 8:00 AM**	**Mon 8/22/22 5:00 PM**
4		Setup NoMachine	1 day	Mon 8/22/22 8:00 AM	Mon 8/22/22 5:00 PM
5		Create Order ERD	1 day	Tue 8/23/22 8:00 AM	Tue 8/23/22 5:00 PM
6		**Database Instance Creation**	**7 days**	**Wed 8/24/22 8:00 AM**	**Tue 8/30/22 5:00 PM**
7		Create and Configure using DBCA	3 days	Wed 8/24/22 8:00 AM	Fri 8/26/22 5:00 PM
8		Verify Instance Creation	1 day	Mon 8/29/22 8:00 AM	Mon 8/29/22 5:00 PM
9		Initial Backup Using RMAN	1 day	Tue 8/30/22 8:00 AM	Tue 8/30/22 5:00 PM
10		**Capacity Planning, Tables, and Tablespace Creation**	**19 days**	**Wed 8/31/22 8:00 AM**	**Sun 9/18/22 5:00 PM**
11		Calculate Space required for Tables	4 days	Wed 8/31/22 8:00 AM	Sat 9/3/22 5:00 PM
12		Create Temporary Tablespace	1 day	Sun 9/4/22 8:00 AM	Sun 9/4/22 5:00 PM
13		Create Users and Roles	3 days	Wed 9/7/22 8:00 AM	Fri 9/9/22 5:00 PM
14		Assign Privileges and Grant Roles	4 days	Sat 9/10/22 8:00 AM	Tue 9/13/22 5:00 PM
15		Design DDL for Ordinary and Partitioned Tables	3 days	Wed 9/14/22 8:00 AM	Fri 9/16/22 5:00 PM
16		Run DDL to Create Tables	2 days	Sat 9/17/22 8:00 AM	Sun 9/18/22 5:00 PM
17		**Populate Tables with SQL*Loader**	**14 days**	**Wed 9/21/22 8:00 AM**	**Tue 10/4/22 5:00 PM**
18		Create SQL*Loader with Located .ctl Files	3 days	Wed 9/21/22 8:00 AM	Fri 9/23/22 5:00 PM
19		Run SQL*Loader and Analyze Tables	4 days	Sat 9/24/22 8:00 AM	Tue 9/27/22 5:00 PM
20		Project Progress Report Deliverable	7 days	Wed 9/28/22 8:00 AM	Tue 10/4/22 5:00 PM
21		**Database Monitoring**	**21 days**	**Wed 10/5/22 8:00 AM**	**Tue 10/25/22 5:00 PM**
22		Create Workloads to Stress Database	3 days	Wed 10/5/22 8:00 AM	Fri 10/7/22 5:00 PM
23		Create Scripts to Monitor Database Statistics	2 days	Sat 10/8/22 8:00 AM	Sun 10/9/22 5:00 PM
24		Generate ADDM Report	2 days	Mon 10/10/22 8:00 AM	Tue 10/11/22 5:00 PM
25		Assess Performance Recommendation	2 days	Wed 10/12/22 8:00 AM	Thu 10/13/22 5:00 PM
26		Generate New ADDM Report Change	1 day	Tue 10/18/22 8:00 AM	Tue 10/18/22 5:00 PM
27		Final Project Report Deliverable	7 days	Wed 10/19/22 8:00 AM	Tue 10/25/22 5:00 PM

Figure 3

Project Gantt Chart



Scope

In-Scope

The following are within the scope of the project:

- Assessment of the linux virtual environment.
- Creation of logical and physical models.
- Instance creation and backup of the database using RMAN.
- Data definition and manipulation languages generated and said queries ran into the database.

- Creation of users, roles, indices and tablespaces
- Export and importing data.
- Using SQL*Loader to load the sample order data into the database.
- Verifying objects and analyzing tables using the COMPUTE STATISTICS clause.
- Monitoring database statistics.
- Performance tuning with the Automatic Database Diagnostic Monitor.

Out-of-Scope

The following are out of the scope:

- Populating the database system with actual data.
- Enacting role-based security measures and data redaction.

Hardware and Software Tools

Data Modeler:

- Oracle SQL*Loader

Database Management System

- Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version

  Oracle SQL Developer 19.3.0.0.0 64-bit running on Windows 10

Hardware and Software

- Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz - 2.81 GHz, 8.00 GB (7.88 GB usable), 64-bit operating system, x64-based processor, Windows 11 Home v. 21H2 OS build 22000.1098

Requirements Definition

Business Rules

a. A ORDERS can has one ORDER_ITEM.

b. A ORDER_ITEM contains one to many ORDERS.

c. A PRODUCTS provides one to many for the SUPPLIER.

d. A CUSTOMER places one to many ORDERS.

e. A ORDERS requires one to many PAYMENTS.

f. A EMPLOYEE receives one to many ORDERS.

g. A EMPLOYEE works for one or many STORES.

h. A CUSTOMER belongs to one LOCATION.

i. A STORE exists in one LOCATION.

j. A SUPPLIER operates in one LOCATION.

k. A MANAGER manages one STORE.

l. A EMPLOYEE reports to one to many MANAGER.

m. A MANAGER resides in one LOCATION.

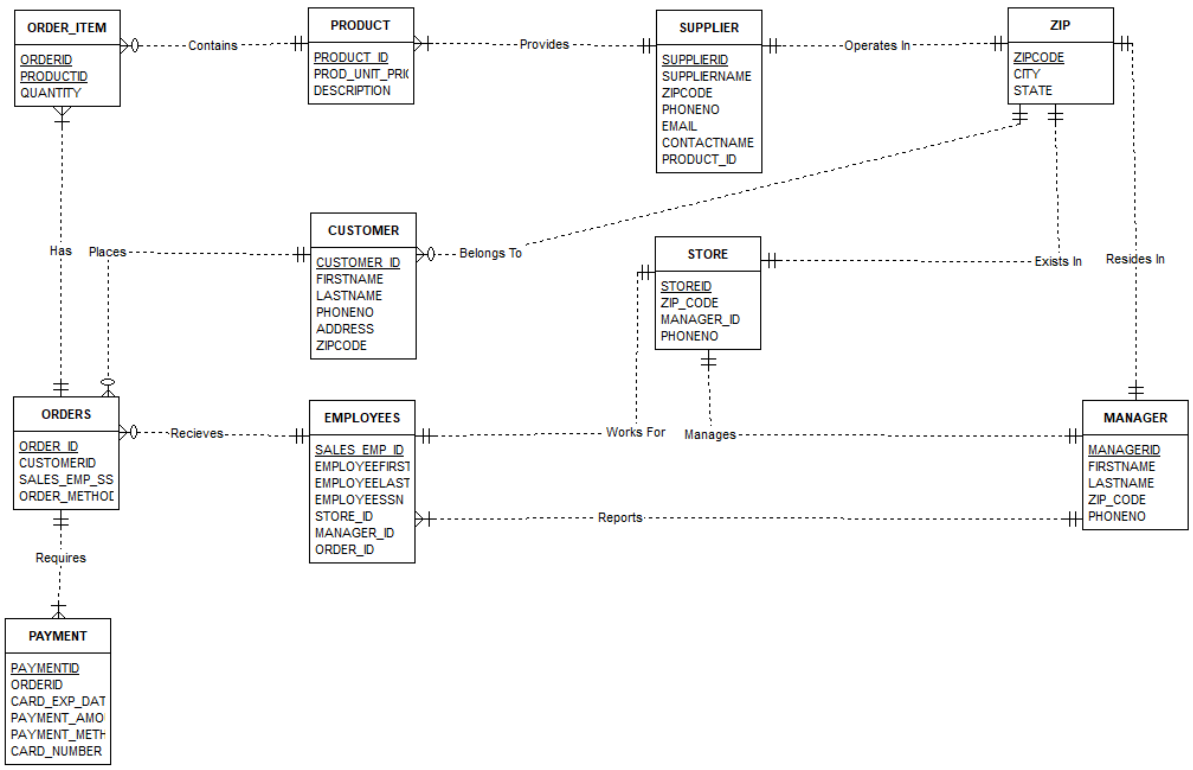
Data Tables

The database will have 10 tables: Orders, Payment, Order_Item, Products, Supplier, Location, Store, Manager, Employees,

Entity Relationship Diagram

Figure 1

Physical Model



Capacity Planning

This statement is referred for capacity calculations. overhead = fixed header + variable transaction header + table directory + row directory.

Knowing that:

Block size is 8192 bytes

overhead = 8192 bytes

fixed header = 57 bytes

transaction header = 23 * initrans value for the table

table directory = 4 (for non-cluster databases)

row directory = 2*R

R = number of rows in the block

Therefore:

overhead = 57 bytes + 23*i + 4 bytes + 2R

overhead = 61 bytes + 23*(1) + 2R

8192 bytes = 84 bytes + 2R

8108 bytes = 2R

data space = available space - (available space * pctfree)

data space = (8108 bytes - 2R) - ((8108-2R))*1/10

data space = 8108 - 2R - 810.8 +.2R

data space = 7297.2 - 1.8R

We can calculate. Below are assumptions in terms of the columns and their sizes:

Products

ProductID-> 10 bytes

Prod_Unit_Price-> log(100) / log(2) / 8 = 1 byte

Description -> 100 bytes

Supplier_ID -> 10 bytes

(10 + 1 + 100 + 10) bytes = 122 bytes

R = (7297.2 - 1.8*R)/122

1.015R = 59.81

R = 58.93

Order_item

order_id -> 8 byte

productID ->10 byte

quantity -> log(1000) / log(2) / 8 = 1 byte

(8 + 10 + 1) bytes = 19 bytes

R = (7297.2 - 1.8*R)/19

1.09R = 384.06

R = 352.35

Payment

PaymentID -> 10 bytes

OrderID -> 10 byte

CardExpDate -> Date = 4 bytes

PaymentAmount -> log(100) / log(2) / 8 = 1

PaymentMethod -> 10 bytes

(10 + 10 + 4 + 1 + 10) bytes = 35 bytes

R = (7297.2 - 1.8*R)/35

1.051R = 208.49

R = 198.37

Orders

OrderID -> 8 byte

CustomerID -> 10 bytes

SalesEmpID -> 8 byte

OrderMethod -> 10 bytes

(8 + 10 + 8 + 10) bytes = 36 bytes

R = (7297.2 - 1.8*R)/36

1.05R = 202.7

R = 193.05

Customer

CustomerID -> 10 bytes

FirstName -> 13 bytes

LastName -> 13 bytes

PhoneNo -> 5 byte

LocationID -> 100 bytes

(10 + 13 + 13 + 5 + 100) bytes = 141 bytes

R = (7297.2 - 1.8*R)/141

1.02R = 86.87

R = 85.17

Zip

LocationID -> 10 bytes

ZipCode -> 1 byte

C -> 25 bytes

State -> 25 bytes

Address -> 50 bytes

(10 + 1 + 25 + 25 + 50) bytes = 110 bytes

R = (7297.2 - 1.8*R)/110

1.016R = 66.34

R = 65.3

Supplier

SupplierID -> 10 bytes

SupplierName -> 50 bytes

LocationID -> 10 bytes

PhoneNumber -> 5 byte

Email -> 50 bytes

ContactName - > 50 bytes

(10 + 50 + 10 + 5 + 50 + 50) bytes = 175 bytes

R = (7297.2 - 1.8*R)/175

1.01R = 41.70

R = 41.3

Store

StoreNo -> 1 byte

Address -> 40 bytes

ZipCode -> 1 byte

PhoneNo -> 1 byte

Email -> 30 bytes

ContactName -> 40 bytes

(1 + 40 + 1 + 1 + 30 + 40) bytes = 113

R = (7297.2 - 1.8*R)/113

1.0159R = 64.57

R = 63.56

Employees

SalesEmpID -> 8 bytes

EmployeeFirstName -> 25 bytes

EmployeeLastName -> 25 bytes

EmployeeSSN -> 5 byte

StoreID -> 10 bytes

ManagerID -> 10 bytes

(8 + 25 + 25 + 5 + 10 + 10) bytes = 83 bytes

R = (7297.2 - 1.8*R)/83

1.0217R = 87.92

R = 86.05

Manager

ManagerID -> 10 byte

FirstName -> 25 bytes

LastName -> 25 bytes

LocationID -> 10 bytes

PhoneNumber -> 5 bytes

ContactName -> 40 bytes

(10 + 25 + 25 + 10 + 5 + 40) bytes = 115 bytes

R = (7297.2 - 1.8*R)/115

1.0156R = 63.45

R = 62.47

Fetching the .dat files, we are able to see the number of row entries.

```
[oracle@NIXORA01 dir_6tables]$ cat customer.dat|wc -l
1006
[oracle@NIXORA01 dir_6tables]$ cat order_item.dat|wc -l
1102
[oracle@NIXORA01 dir_6tables]$ cat order.dat|wc -l
1103
```

```
[oracle@NIXORA01 dir_6tables]$ cat payment.dat |wc -l
1003
[oracle@NIXORA01 dir_6tables]$ cat product.dat |wc -l
14279
[oracle@NIXORA01 dir_6tables]$ cat hb_zip.dat| wc -l
50002
```

Block calculations are as follows. The manager, employee, store, and supplier tables are user constructed:

ProductBlock = ceiling(14279/58.93) = ceiling(242.30) = 243 blocks = 1990656 bytes

Order_itemBlock = ceiling(1102/352.35) = ceiling(3.127) = 4 blocks = 32768 bytes

PaymentBlock = ceiling(1003/198.37) = ceiling(5.06) = 6 blocks = 49152 bytes

OrdersBlock = ceiling(1103/193.05) = ceiling(5.71) = 6 blocks = 49152 bytes

CustomerBlock = ceiling(1006/85.17) = ceiling(11.81) = 12 blocks = 98304 bytes

ZipBlock = ceiling(50002/65.3) = ceiling(765.72) = 766 blocks = 6275072 bytes

SupplierBlock = ceiling(1000/41.3) = ceiling(24.21) = 24 blocks = 196608 bytes

StoreBlock = ceiling(1000/63.56) = ceiling(15.73) = 16 blocks = 131072 bytes

EmployeesBlock = ceiling(1000/86.05) = ceiling(11.62) = 12 blocks = 98304 bytes

ManagerBlock = ceiling(1000/62.47) = ceiling(16.01) = 16 blocks = 131072 bytes

With the block calculations done, it is compared with the actual table sizes.

Creating the Instance

Logging into the Linux os virtual machine, the user establishes a connection to 10.11.55.52 using the NoMachine software. The following is run in the terminal to add the non-network location connections to the access control list, then the database configuration assistant is started:

```
[StudentFirst@NIXORA01 ~]$ xhost +local:oracle
[StudentFirst@NIXORA01 ~]$ sudo -u oracle $ORACLE_HOME/bin/dbca
```

The user creates the database using the configuration assistant wizard and can be referenced in the index. Initial databses are deleted, and the user goes with advanced configuration. It is an transaction processing Oracle Single Instance Database with the global database name of DBST670. File system location is: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}. Fast recovery storage is 8256 megabytes in the /home area. Listener port 1521 in the Oracle home directory. SGA size is 3000 and PGA size is 1000 using the automatic shared memory management configuration. Processes sizing of 640. Character set is the universal unicode AL32UTF8. Control files are confirmed. Dedicated server mode initialized. RMAN is also initialized to backup the data.

```
[StudentFirst@NIXORA01 ~]$ rman target /
RMAN> backup database plus archivelog;
RMAN> backup archivelog all;
```

User, Tablespace and Data Definition Language Creation

The database is created for the user. Default tablespace is set to USERS and the defeault temporary tables to TEMP. CONNECT and RESOURCE roles are granted as well as the DBA.. An unprivileges user is granted the owner of the tables for the database with the user name ‘hanifdba’ password ‘oracle’..

```
[StudentFirst@NIXORA01 ~]$ sudo -s -u oracle
[oracle@NIXORA01 StudentFirst] . oraenv
ORACLE_SID = [NIXORA01] ? DBST670
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@NIXORA01 StudentFirst]$ sqlplus /nolog

SQL*Plus: Release 19.0.0.9.0 - Production on Thu Sep 15 11:15:03 2022
```

```
Version 19.3.0.0

Copyright © 1982, 2019 Oracle. All rights reserved.
```

```
SQL> startup
SQL> connect / as sysdba;
Connected.
SQL> create user hlumsden identified by oracle default tablespace
users temporary tablespace temp;
User created.
SQL> create user hanifdba identified by oracle default tablespace
users temporary tablespace temp quota unlimited on users;
SQL> grant connect, resource to hlumsden;
User reated.
SQL> grant hanifdba to dbauser;
Grant succeeded.
SQL> CREATE TABLESPACE "USER_DATA1" LOGGING DATAFILE
'/u01/app/oracle/oradata/DBST670/user_data101.dbf' SIZE 4286K REUSE
AUTOEXTEND ON NEXT 512K MAXSIZE 65536K EXTENT MANAGEMENT LOCAL;
Tablespace created.
SQL> CREATE TABLESPACE "USER_DATA2" LOGGING DATAFILE
'/u01/app/oracle/oradata/DBST670/user_data201.dbf' SIZE 4134K REUSE
AUTOEXTEND ON NEXT 512K MAXSIZE 65536K EXTENT MANAGEMENT LOCAL;
Tablespace created.
SQL> CREATE TABLESPACE "USER_INDX" LOGGING DATAFILE
'/u01/app/oracle/oradata/DBST670/user_indx01.dbf' SIZE 4210K REUSE
AUTOEXTEND ON NEXT 512K MAXSIZE 65536K EXTENT MANAGEMENT LOCAL;
Tablespace created.
SQL> CREATE TEMPORARY TABLESPACE "USER_TEMP" TEMPFILE
'/u01/app/oracle/oradata/DBST670/user_temp01.dbf' SIZE 16384K REUSE
AUTOEXTEND ON NEXT 512K MAXSIZE 65536K EXTENT MANAGEMENT LOCAL;
Tablespace created.
SQL> alter user hanif default tablespace user_data1 temporary
tablespace user_temp;
User altered.
SQL> SHOW USER
USER is "HANIF"
SQL> DROP TABLE customer CASCADE CONSTRAINTS PURGE;
```

```
Table dropped.
SQL> CREATE TABLE customer (customer_id INTEGER NOT NULL, firstname
VARCHAR2(15),
lastname VARCHAR2(15), phoneno CHAR(15), zip_code CHAR(5), address
CHAR(35)) TABLESPACE
USER_DATA1;
Table created.
SQL> ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY
(customer_id) USING
INDEX TABLESPACE USER_INDX
Table altered.
SQL>
SQL> DROP TABLE employees CASCADE CONSTRAINTS PURGE;
Table dropped.
SQL> CREATE TABLE employees (sales_emp_id INTEGER NOT NULL, store_id
INTEGER,
order_id INTEGER, manager_id INTEGER, employeessn VARCHAR2(9),
employeelastname
VARCHAR2(50), employeefirstname VARCHAR2(50)) TABLESPACE USER_DATA1;
Table created.
SQL> ALTER TABLE employees ADD CONSTRAINT employees_pk PRIMARY KEY (
sales_emp_id )
USING INDEX TABLESPACE USER_INDX;
Table altered.
SQL> DROP TABLE manager CASCADE CONSTRAINTS PURGE;
Table dropped.
SQL> CREATE TABLE manager (managerid INTEGER NOT NULL, phoneno
INTEGER, firstname
VARCHAR2(150), lastname VARCHAR2(150), zip_code CHAR(5)) TABLESPACE
USER_DATA1;
Table created.
SQL> ALTER TABLE manager ADD CONSTRAINT manager_pk PRIMARY KEY (
managerid ) USING
INDEX TABLESPACE USER_INDX;
Table altered.
SQL> DROP TABLE order_item CASCADE CONSTRAINTS PURGE;
Table dropped.
SQL> CREATE TABLE order_item (orderid INTEGER NOT NULL, productid
INTEGER NOT NULL,
```

```
quantity INTEGER) TABLESPACE USER_DATA1;
Table created.
SQL> ALTER TABLE order_item ADD CONSTRAINT order_item_pk PRIMARY KEY
(
orderid,productid ) USING INDEX TABLESPACE USER_INDX;
Table altered.
SQL> DROP TABLE orders CASCADE CONSTRAINTS PURGE;
Table dropped
SQL> CREATE TABLE orders (order_id INTEGER NOT NULL, customerid
INTEGER, order_method
CHAR(6), sales_emp_ssn INTEGER) TABLESPACE USER_DATA1;
Table created.
SQL> ALTER TABLE orders ADD CONSTRAINT orders_pk PRIMARY KEY (
order_id ) USING INDEX
TABLESPACE USER_INDX;
Table altered.
SQL> DROP TABLE PAYMENTS CASCADE CONSTRAINTS PURGE;
Table dropped.
SQL> CREATE TABLE payment ( paymentid INTEGER NOT NULL, orderid
INTEGER,
payment_amount INTEGER, payment_method VARCHAR2(5), card_number
INTEGER, card_exp_date
DATE) TABLESPACE USER_DATA2;
Table created.
SQL> ALTER TABLE payment ADD CONSTRAINT payment_pk PRIMARY KEY (
paymentid ) USING
INDEX TABLESPACE USER_INDX;
Table altered.
SQL> DROP TABLE products CASCADE CONSTRAINTS PURGE;
Table dropped.
SQL> CREATE TABLE product (product_id INTEGER NOT NULL,
prod_unit_price INTEGER,
description VARCHAR2(150)) TABLESPACE USER_DATA2;
Table created.
SQL> ALTER TABLE product ADD CONSTRAINT product_pk PRIMARY KEY (
product_id ) USING
INDEX TABLESPACE USER_INDX;
Table altered.
SQL> DROP TABLE store CASCADE CONSTRAINTS PURGE;
```

```
Table dropped.
SQL> CREATE TABLE store (storeid INTEGER NOT NULL, manager_id
INTEGER, phoneno
CHAR(15), zip_code CHAR(5)) TABLESPACE USER_DATA2;
Table created.
SQL> ALTER TABLE store ADD CONSTRAINT store_pk PRIMARY KEY ( storeid
) USING INDEX
TABLESPACE USER_INDX;
Table altered.
SQL> DROP TABLE supplier CASCADE CONSTRAINTS PURGE;
Table dropped.
SQL> CREATE TABLE supplier (supplierid INTEGER NOT NULL, suppliername
VARCHAR2(150),
zipcode CHAR(5), product_id INTEGER, phoneno INTEGER, email
VARCHAR2(100), contactname
VARCHAR2(150)) TABLESPACE USER_DATA2;
Table created.
SQL> ALTER TABLE supplier ADD CONSTRAINT supplier_pk PRIMARY KEY (
supplierid ) USING
INDEX TABLESPACE USER_INDX;
Table altered.
SQL> DROP TABLE zip CASCADE CONSTRAINTS PURGE;
Table dropped.
SQL> CREATE TABLE zip (zipcode CHAR(5) NOT NULL, city CHAR(25), state
CHAR(2))
TABLESPACE USER_DATA2;
Table created.
SQL> ALTER TABLE zip ADD CONSTRAINT zip_pk PRIMARY KEY (zipcode)
USING INDEX TABLESPACE
USER_INDX;
Table altered.
```

Foreign Keys

```
SQL> ALTER TABLE customer ADD CONSTRAINT customer_zip_fk FOREIGN KEY
(zip_code)
REFERENCES zip ( zipcode );
Table altered.
SQL> ALTER TABLE employees ADD CONSTRAINT employees_manager_fk
FOREIGN KEY (
```

```
manager_id ) REFERENCES manager ( managerid );
Table altered.
SQL> ALTER TABLE employees ADD CONSTRAINT employees_store_fk FOREIGN
KEY ( store_id )
REFERENCES store ( storeid );
Table altered.
SQL> ALTER TABLE manager ADD CONSTRAINT manager_zip_fk FOREIGN KEY (
zip_code )
REFERENCES zip ( zipcode );
Table altered.
SQL> ALTER TABLE order_item ADD CONSTRAINT order_item_orders_fk
FOREIGN KEY (
orderid ) REFERENCES orders ( order_id );
Table altered.
SQL> ALTER TABLE order_item ADD CONSTRAINT order_item_product_fk
FOREIGN KEY (
productid ) REFERENCES product ( product_id );
Table altered.
SQL> ALTER TABLE orders ADD CONSTRAINT orders_customer_fk FOREIGN KEY
(
customerid ) REFERENCES customer ( customer_id );
Table altered.
SQL> ALTER TABLE employees ADD CONSTRAINT employees_orders_fk FOREIGN
KEY ( order_id )
REFERENCES orders (order_id);
Table altered.
SQL> ALTER TABLE payment ADD CONSTRAINT payment_orders_fk FOREIGN KEY
( orderid )
REFERENCES orders ( order_id );
Table altered.
SQL> ALTER TABLE supplier ADD CONSTRAINT supplier_product_fk FOREIGN
KEY ( product_id )
REFERENCES product ( product_id );
Table altered.
SQL> ALTER TABLE store ADD CONSTRAINT store_manager_fk FOREIGN KEY (
manager_id )
REFERENCES manager ( managerid );
Table altered.
SQL> ALTER TABLE store ADD CONSTRAINT store_zip_fk FOREIGN KEY (
```

```
zip_code ) REFERENCES
zip ( zipcode );
Table altered.
SQL> ALTER TABLE supplier ADD CONSTRAINT supplier_zip_fk FOREIGN KEY
( zipcode )
REFERENCES zip ( zipcode );
Table altered.
```

Creating Partitioned Table

```
SQL> drop table ZIP cascade constraints purge;
Table dropped.
SQL> create table ZIP (ZIPCODE char(5) NOT NULL,CITY char(25),STATE
char(2),
constraint pk_zip primary key (ZIPCODE) using index storage (initial
25k next 12k)
tablespace USER_INDX) partition by range (state) (partition zip_part1
values less than ('N')
pctfree 5 pctused 90 tablespace USER_DATA1 storage (initial 1500k
next 200k pctincrease 0 maxextents 5), partition zip_part2 values
less than (MAXVALUE) pctfree 5 pctused 90
tablespace USER_DATA2 storage (initial 1500k next 200k pctincrease 0
maxextents 5));
Table created.
SQL> drop table CUSTOMER cascade constraints purge;
Table dropped.
SQL> create table CUSTOMER (ACCOUNT_NUMBER number(10) NOT NULL, SSN
number(9) NOT
NULL,FIRST_NAME char(10), MIDDLE_INITL char(1), LAST_NAME char(20)
NOT NULL,FIRST_LINE_AD
char(20), FK_ZIPCODE char(5), constraint pk_customer primary key
(ACCOUNT_NUMBER) using
index storage (initial 25k next 12k) tablespace USER_INDX, constraint
fk_zip foreign key
(FK_ZIPCODE) references ZIP(ZIPCODE)) pctfree 0 pctused 90 storage
(initial 50k next 10k
pctincrease 0 maxextents 5) tablespace USER_DATA1;
Table created.
SQL> create index cust_zip_idx on CUSTOMER(FK_ZIPCODE)
storage(initial 10k next 10k)
```

```
tablespace USER_INDX;
Index created.
```

Synonyms are created

```
CREATE OR REPLACE PUBLIC SYNONYM CUSTOMER FOR HANIF.CUSTOMER;
CREATE OR REPLACE PUBLIC SYNONYM PRODUCT FOR HANIF.PRODUCT;
CREATE OR REPLACE PUBLIC SYNONYM ORDERS FOR HANIF.ORDERS;
CREATE OR REPLACE PUBLIC SYNONYM ORDER_ITEM FOR HANIF.ORDER_ITEM;
CREATE OR REPLACE PUBLIC SYNONYM ZIP FOR HANIF.ZIP;
CREATE OR REPLACE PUBLIC SYNONYM PAYMENT FOR HANIF.PAYMENT;
CREATE OR REPLACE PUBLIC SYNONYM SUPPLIER FOR HANIF.SUPPLIER;
CREATE OR REPLACE PUBLIC SYNONYM EMPLOYEES FOR HANIF.EMPLOYEES;
CREATE OR REPLACE PUBLIC SYNONYM STORE FOR HANIF.STORE;
CREATE OR REPLACE PUBLIC SYNONYM MANAGER FOR HANIF.MANAGER;
```

To take notes of the users and their activity, the DBA will show what a user is eecuting and their details:

```
COL USERNAME FOR A10
COL MACHINE FOR A10
COL SCHEMANAME FOR A10
COL OSUSER FOR A12
COL PROGRAM FOR A30
COL MODULE FOR A20
COL EVENT FOR A30
COL STATUS FOR A8
COL SERVICE_NAME FOR A12
COL STATE FOR A8
SQL> SELECT USERNAME, STATUS,SCHEMANAME,OSUSER,MACHINE, PROGRAM,
MODULE, LOGON_TIME, LAST_CALL_ET,
STATE, SERVICE_NAME, EVENT, SQL_ID FROM V$SESSION WHERE
USERNAME='&USERNAME';
Enter value for username: HANIF
USERNAME STATUS SCHEMANAME OSUSER MACHINE PROGRAM MODULE
LOGON_TIM LAST_CALL_ET STATE SERVICE_NAME EVENT SQL_ID
---------- -------- ---------- ------------ ----------
------------------------------ ------------------
-- --------- ------------ -------- ------------
------------------------------ -------------
```

```
HANIF INACTIVE HANIF oracle NIXORA01 sqlplus@NIXORA01 (TNS V1-V3)
SQL*Plus
30-OCT-22 0 WAITING SYS$USERS SQL*Net message from client
3ws0hu2bdjgt0
1 row selected.
SQL>
```

The event column will inform the DBA or system administrator what the user is doing. The “db file sequential read” entails reading from a block sequentially and is an I/O operation. “SQL*net message from client” means waiting for the next query to be fired and to complete its execution. SQL_ID will allow the user to know what other users are doing in terms of SQL execution.

Export and Import

The database administrator user creates test table to test the export and import function to demonstrate capability.

```
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.


Username: hanifdba
Password:

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Produc
tion
Version 19.3.0.0.0
Export done in US7ASCII character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)

About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user TESTUSER
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user TESTUSER
About to export TESTUSER's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export TESTUSER's tables via Conventional Path ...
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully without warnings.
[oracle@NIXORA01 StudentFirst]$
```

RMAN is most efficient for backup and recovery to the database though it cannot backup a single table or schema like the export-import functions can. For schema level the DBA will use export-import, otherwise RMAN should be utilized.

Loading .ctl and .dat files using SQL*Loader

The SQL*Loader utility will load the respective .ctl and .dat files for the order tables. SQL*Loader acts as a utility for the .ctl file parameters to be pass through resulting in said utility

opening the .ctl file and loading the data. Store, manager, employee, and supplier table data is randomly generate to produce a sample data using the Mockaroo website (Moomin, 2022). Data loading is limited to 1000 tables max and 50000 max according to the load files in the index. The loading is done through Linux, fetching the tables in the table directory [oracle@NIXORA01 dir_6tables]. The process is shown below:

```
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/zip.ctl
data=/u01/app/oracle/repository/dir_6tables/hb_zip.dat
bad=/u01/app/oracle/repository/dir_6tables/zip.bad
discard=/u01/app/oracle/repository/dir_6tables/zip.dsc
log=/u01/app/oracle/repository/dir_6tables/zip.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production on Wed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=50000
Load completed - logical record count 50000.
Table ZIP:
      50000 Rows successfully loaded.
Check the log file:
      /u01/app/oracle/repository/dir_6tables/store.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/orders.ctl
data=/u01/app/oracle/repository/dir_6tables/order.dat
bad=/u01/app/oracle/repository/dir_6tables/orders.bad
discard=/u01/app/oracle/repository/dir_6tables/orders.dsc
log=/u01/app/oracle/repository/dir_6tables/orders.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production on Wed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table ORDERS:
      1000 Rows successfully loaded.
Check the log file:
      /u01/app/oracle/repository/dir_6tables/store.log
```

```
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif//oracle
control=/u01/app/oracle/repository/dir_6tables/product.ctl
data=/u01/app/oracle/repository/dir_6tables/product.dat
bad=/u01/app/oracle/repository/dir_6tables/product.bad
discard=/u01/app/oracle/repository/dir_6tables/product.dsc
log=/u01/app/oracle/repository/dir_6tables/product.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production on Wed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table PRODUCT:
      1000 Rows successfully loaded.
Check the log file:
      /u01/app/oracle/repository/dir_6tables/store.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/order_item.ctl
data=/u01/app/oracle/repository/dir_6tables/order_item.dat
bad=/u01/app/oracle/repository/dir_6tables/order_item.bad
discard=/u01/app/oracle/repository/dir_6tables/order_item.dsc
log=/u01/app/oracle/repository/dir_6tables/order_item.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production onWed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table ORDER_ITEM:
      1000 Rows successfully loaded.
Check the log file:
      /u01/app/oracle/repository/dir_6tables/order_item.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/payment.ctl
data=/u01/app/oracle/repository/dir_6tables/payment.dat
bad=/u01/app/oracle/repository/dir_6tables/payment.bad
discard=/u01/app/oracle/repository/dir_6tables/payment.dsc
```

```
log=/u01/app/oracle/repository/dir_6tables/payment.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production onWed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table PAYMENT:
     1000 Rows successfully loaded.
Check the log file:
     /u01/app/oracle/repository/dir_6tables/store.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/customer.ctl
data=/u01/app/oracle/repository/dir_6tables/customer.dat
bad=/u01/app/oracle/repository/dir_6tables/customer.bad
discard=/u01/app/oracle/repository/dir_6tables/customer.dsc
log=/u01/app/oracle/repository/dir_6tables/customer.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production onWed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table CUSTOMER:
     1000 Rows successfully loaded.
Check the log file:
     /u01/app/oracle/repository/dir_6tables/store.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/employee.ctl
data=/u01/app/oracle/repository/dir_6tables/employee.dat
bad=/u01/app/oracle/repository/dir_6tables/employee.bad
discard=/u01/app/oracle/repository/dir_6tables/employee.dsc
log=/u01/app/oracle/repository/dir_6tables/employee.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production on Wed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table EMPLOYEE:
     1000 Rows successfully loaded.
```

```
Check the log file:
      /u01/app/oracle/repository/dir_6tables/store.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/manager.ctl
data=/u01/app/oracle/repository/dir_6tables/manager.dat
bad=/u01/app/oracle/repository/dir_6tables/manager.bad
discard=/u01/app/oracle/repository/dir_6tables/manager.dsc
log=/u01/app/oracle/repository/dir_6tables/manager.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production onWed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table MANAGER:
      1000 Rows successfully loaded.
Check the log file:
      /u01/app/oracle/repository/dir_6tables/store.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/store.ctl
data=/u01/app/oracle/repository/dir_6tables/store.dat
bad=/u01/app/oracle/repository/dir_6tables/store.bad
discard=/u01/app/oracle/repository/dir_6tables/store.dsc
log=/u01/app/oracle/repository/dir_6tables/store.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production onWed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table STORE:
      1000 Rows successfully loaded.
Check the log file:
      /u01/app/oracle/repository/dir_6tables/store.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
sqlldr userid=hanif/oracle
control=/u01/app/oracle/repository/dir_6tables/supplier.ctl
data=/u01/app/oracle/repository/dir_6tables/supplier.dat
```

```
bad=/u01/app/oracle/repository/dir_6tables/supplier.bad
discard=/u01/app/oracle/repository/dir_6tables/supplier.dsc
log=/u01/app/oracle/repository/dir_6tables/supplier.log direct=y
SQL*Loader: Release 19.0.0.0.0 - Production onWed Oct 12 19:43:28
2022 Version 19.3.0.0.
Path used: Direct, LOAD=1000
Load completed - logical record count 1000.
Table SUPPLIER:
     998 Rows successfully loaded.
Check the log file:
     /u01/app/oracle/repository/dir_6tables/store.log
for more information about the load.
[oracle@NIXORA01 dir_6 tables]$
```

Table Analyze

The user will run clauses to analyze the tables. ANALYZE TABLE command is used along with the DBMS_STATS including verifying that objects are valid. First, the tables are analyzed so that oracle can collect statistics.

```
SQL> analyze table ZIP compute statistics;
Table analyzed.
SQL> analyze table ORDER_ITEM compute statistics;
Table analyzed.
SQL> analyze table ORDERS compute statistics;
Table analyzed.
SQL> analyze table PAYMENT compute statistics;
Table analyzed.
SQL> analyze table PRODUCT compute statistics;
Table analyzed.
SQL> analyze table CUSTOMER compute statistics;
Table analyzed.
SQL> analyze table EMPLOYEES compute statistics;
Table analyzed.
SQL> analyze table SUPPLIER compute statistics;
Table analyzed.
SQL> analyze table STORE compute statistics;
Table analyzed.
```

```
SQL> analyze table MANAGER compute statistics;
Table analyzed.
```

The DBA can then use the LAST_ANALYZE, STALE_STATS and NUM_ROWS clauses as examples of user statistics.

```
SQL>SET LINES 350
SQL>SET PAGES 5000
SQL>col table_name for a25
SQL>select table_name, last_analyzed, stale_stats from
user_tab_statistics where
table_name in ('ZIP', 'EMPLOYEES', 'SUPPLIER', 'ORDER_ITEM',
'ORDERS', 'PAYMENT', 'PRODUCT', 'STORE', 'CUSTOMER', ' MANAGER')
ORDER BY 2 DESC;
```

Result:

```
SQL> SET PAGES 5000
SQL> SET LINES 350
SQL> COL TABLE_NAME FOR A25
SQL> SELECT TABLE_NAME, LAST_ANALYZED, STALE_STATS, NUM_ROWS FROM USER_TAB_STATI
STICS WHERE TABLE_NAME IN ('ZIP', 'EMPLOYEES', 'SUPPLIER', 'ORDER_ITEM', 'ORDERS
', 'PAYMENT', 'PRODUCT', 'STORE', 'CUSTOMER', 'MANAGER') ORDER BY 2 DESC;

TABLE_NAME                LAST_ANAL STALE_S   NUM_ROWS
------------------------- --------- ------- ----------
ZIP                       20-OCT-22 NO           50000
ZIP                       20-OCT-22 NO           24377
ZIP                       20-OCT-22 NO           25623
CUSTOMER                  12-OCT-22 NO            1000
PAYMENT                   12-OCT-22 NO            1000
ORDER_ITEM                12-OCT-22 NO            1000
ORDERS                    12-OCT-22 NO            1000
PRODUCT                   12-OCT-22 NO            1000
MANAGER                   12-OCT-22 NO            1000
STORE                     12-OCT-22 NO            1000
SUPPLIER                  12-OCT-22 NO             998
EMPLOYEES                 12-OCT-22 NO            1000

12 rows selected.

SQL>
```

The statistics that were gathered from the database is done using “DBMS_STATS”

```
SQL>
BEGIN
```

```
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'ZIP', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'PAYMENT', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'ORDERS', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'PRODUCT', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
```

```
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'ORDER_ITEM', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'CUSTOMER', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'EMPLOYEE', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'MANAGER', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
```

```
PL/SQL procedure successfully completed.
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'STORE', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
SQL>
DBMS_STATS.GATHER_TABLE_STATS
(
OWNNAME =>'HANIF', TABNAME => 'SUPPLIER', ESTIMATE_PERCENT =>
DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL COLUMNS SIZE
AUTO', CASCADE => TRUE, GRANULARITY => 'ALL', NO_INVALIDATE => FALSE,
DEGREE => 2);
END;
/
PL/SQL procedure successfully completed.
```

Table Sizes

The table sizes are calculated and compared with the capacity planning. The product table capacity is 384 kb to the 1990 kb. This can be attributed to the only 1000 rows that comprise the product table over the 14,000 estimated. Order_item, customer, product and orders are among the calculations that were undervalued. The rest are overcalculated. There isn't substantial accuracy based on assumptions, but its a good practice to undergo capacity planning for the database project plan. Project managers must be prepare for over or underestimates. Below is the actual table size:

```
SQL> select SEGMENT_NAME as "OBJECT_NAME", SEGMENT_TYPE as
"OBJECT_TYPE", sum(bytes)/(1024) as "TABLE_SIZE(KB)" from
dba_segments where segment_name IN('EMPLOYEES', 'SUPPLIER',
'ORDER_ITEM', 'ORDERS', 'PAYMENT', 'PRODUCT', 'STORE', 'CUSTOMER',
```

```
'MANAGER', 'ZIP') group by SEGMENT_NAME, SEGMENT_TYPE;
OBJECT OBJECT
NAME            TYPE       TABLE_SIZE(KB)
------------------------------   --------------  --------------
ZIP TABLE             PARTITION  5120
PAYMENT               TABLE      64
STORE                 TABLE      64
SUPPLIER              TABLE      128
PRODUCT               TABLE      384
ORDER_ITEM            TABLE      64
EMPLOYEES          TABLE    128
CUSTOMER              TABLE      128
ORDERS                TABLE      64
MANAGER               TABLE      64
10 rows selected.
```

Monitoring the Database Statistics

V$SYSSTAT is used to provide access to various database statistics collected. It is important to note that numbers aren't guaranteed to remain constant from one reading to another, so the DBA should take into consideration the date and time these statistic monitoring sessions occur. The DBA must log in to the SYS user to monitor the database. Physical, cache, and

```
SQL> select name,value from v$sysstat where name like 'physical%'
order by value desc;
NAME
VALUE
----------------------------------------------------------------
------ ----------
physical read total bytes
765639680
physical read bytes
191111168
physical write total bytes
121629237
physical write bytes 24207360
physical read total IO requests
42962
```

```
physical reads 23329
physical reads cache 13355
physical reads direct (lob) 9973
physical reads direct 9973
physical read IO requests
8268
physical write total IO requests
7984
physical reads cache prefetch 5383
physical writes
2955
physical writes from cache
2792
physical writes non checkpoint
2075
physical write IO requests
1386
physical read total multi block requests 215
physical writes direct (lob) 163
physical writes direct
163
physical write total multi block requests 52
physical reads prefetch warmup
6
physical reads direct for securefile flashback block new 0
physical reads cache for securefile flashback block new 0
physical reads for flashback new
0
physical reads retry corrupt 0
physical writes direct temporary tablespace 0
physical reads direct temporary tablespace 0
physical reads for data transfer
0
physical write snap IO requests new allocations 0
physical read snap bytes copy 0
physical read snap bytes base 0
physical read snap IO requests no data
0
physical read snap IO requests copy 0
```

```
physical read snap IO requests base 0
physical write total bytes optimized 0
physical write requests optimized 0
physical read partial requests
0
physical read total bytes optimized 0
physical read requests optimized
0
physical read flash cache hits
0
40 rows selected.
Elapsed: 00:00:00.01
```

```
SQL> select name,value from v$sysstat where name like '%cache%' order
by value desc;
NAME VALUE
---------------------------------------------------------------------
------ ----------
logical read bytes from cache 5093851136
consistent gets from cache 555010
session cursor cache hits 102638
db block gets from cache 66798
db block gets from cache (fastpath) 54339
physical reads cache 16248
physical reads cache prefetch 7121
physical writes from cache 5766
session cursor cache count 2582
LOB table id lookup cache misses 3
Commit SCN cached 1
cell writes to flash cache 0
cell overwrites in flash cache 0
cell partial writes in flash cache 0
cell writes to flash cache for temp IO 0
physical read flash cache hits 0
flash cache inserts 0
flash cache eviction: invalidated 0
flash cache eviction: buffer pinned 0
flash cache eviction: aged out 0
```

```
flash cache insert skip: not current 0
flash cache insert skip: DBWR overloaded 0
flash cache insert skip: exists 0
flash cache insert skip: not useful 0
flash cache insert skip: modification 0
flash cache insert skip: corrupt 0
recovery block gets from cache 0
flashback cache read optimizations for block new 0
flashback securefile cache read optimizations for block new 0
physical reads cache for securefile flashback block new 0
gc cluster flash cache reads served 0
gc cluster flash cache reads received 0
gc cluster flash cache received read time 0
gc cluster flash cache reads failure 0
gc flash cache reads served 0
gc flash cache served read time 0
BA count - cache miss 0
BA count - cache denied 0
cell physical IO bytes saved by columnar cache 0
cell smart IO session cache lookups 0
cell smart IO session cache hits 0
cell smart IO session cache soft misses 0
cell smart IO session cache hard misses 0
cell smart IO session cache hwm 0
txn cache local read hits 0
txn cache local read misses 0
txn cache local loscn read misses 0
txn cache remote loscn read misses 0
txn cache local writes 0
txn cache local usn hash write misses 0
txn cache local usn hash read misses 0
txn cache remote usn hash read misses 0
txn cache remote read hits 0
txn cache remote read misses 0
txn cache remote writes 0
txn cache remote active read misses 0
txn cache local ac read misses 0
txn cache redo sync reads 0
txn cache remote copy hits 0
```

```
txn cache remote copy misses 0
txn cache remote ts/inst mismatch 0
txn cache remote proc misses 0
txn cache local proc misses 0
txn cache upper bound misses 0
txn cache local preset misses 0
txn cache local seq num misses 0
txn cache remote seq num misses 0
txn cache local empty slot misses 0
txn cache remote empty slot misses 0
txn cache local sync commit scn misses 0
txn cache remote sync commit scn misses 0
txn cache remote read msg sent 0
txn cache remote requested xid count 0
txn cache remote fetch double pass 0
txn cache lookup usn exceeds max 0
txn cache lookup slot exceeds max 0
txn cache lookup fail read even version 0
txn cache lookup conflicting write exact 0
txn cache lookup conflicting write uppper 0
txn cache init usn slot mismatch 0
txn cache init usn exceeds max 0
txn cache write usn exceeds max 0
txn cache write slot mismatch 0
IM populate cache read time (ms) 0
IM scan CUs CR cache hit 0
IM scan CUs CR cache miss 0
table scans (cache partitions) 0
cell blocks processed by cache layer 0
cell commit cache queries 0
cell transactions found in commit cache 0
cell blocks helped by commit cache 0
securefile dedup wapp cache miss 0
IM scan rows cache delta exists 0
IM scan rows cache populate 0
IM scan rows cache no delta 0
IM scan blocks cache 0
key vector CUs processed using cached integer join keys 0
cell flash cache read hits 0
```

```
cell flash cache read hits for controlfile reads 0
cell flash cache read hits for smart IO 0
cell flash cache read hits for temp IO 0
cell ram cache read hits 0
cell pmem cache read hits 0
cell pmem cache writes 0
104 rows selected.
Elapsed: 00:00:00.01
SQL>
```

Statistic Interpretation

Some statistics stick out: the CPU used by the session, sorts, session pga memory, file i/o wait time, user i/o wait time, redo size, redo wastage, enqueue timeouts and waits, db block gets and from the cache as well. Five of these statistics are further assessed: CPU, sorts, user i/o wait time, redo wastage, and enqueue timeouts.

```
SQL> select name,value from v$sysstat where name in ('CPU used by
this session', 'sorts
(memory)', 'user I/O wait time', 'redo wastage','enqueue timeouts')
order by value desc;
NAME                                             VALUE
-------------------------------------------------- ----------
redo wastage                                      37124
sorts (memory)                                    8324
CPU used by this session                          1764
user I/O wait time                                1360
enqueue timeouts                                  5
5 rows selected.
Elapsed: 00:00:00.01
```

To test the effects of workload, the workload query that runs for 1:00:00 is ran and its effect on the statistics are noted.

```
SQL> ed
Wrote file afiedt.buf
1 DECLARE
2 P_CUSTOMER_ID HANIF.CUSTOMER.CUSTOMER_ID%TYPE;
```

```
3 P_FIRSTNAME HANIF.CUSTOMER.FIRSTNAME%TYPE;
4 P_LASTNAME HANIF.CUSTOMER.LASTNAME%TYPE;
5 P_ADDRESS HANIF.CUSTOMER.ADDRESS%TYPE;
6 P_ORDER_ID HANIF.ORDERS.ORDER_ID%TYPE;
7 P_ORDER_METHOD HANIF.ORDERS.ORDER_METHOD%TYPE;
8 P_CUSTOMERID HANIF.ORDERS.CUSTOMERID%TYPE;
9 P_PRODUCT_ID HANIF.PRODUCT.PRODUCT_ID%TYPE;
10 P_DESCRIPTION HANIF.PRODUCT.DESCRIPTION%TYPE;
11 P_QUANTITY HANIF.ORDER_ITEM.QUANTITY%TYPE;
12 i NUMBER;
13 BEGIN
14 i :=1;
15 FOR i IN 1..100000000 LOOP
16 SELECT C.CUSTOMER_ID, C.FIRSTNAME, C.LASTNAME, C.ADDRESS,
O.ORDER_ID, O.ORDER_METHOD
INTO P_CUSTOMER_ID, P_FIRSTNAME, P_LASTNAME, P_ADDRESS, P_ORDER_ID,
P_ORDER_METHOD FROM
HANIF.CUSTOMER C INNER JOIN HANIF.ORDERS O ON C.CUSTOMER_ID =
O.CUSTOMERID AND ROWNUM=1;
17 SELECT O.ORDER_ID, O.CUSTOMERID, O.ORDER_METHOD, P.PRODUCT_ID,
P.DESCRIPTION,
OT.QUANTITY INTO P_ORDER_ID, P_CUSTOMERID, P_ORDER_METHOD,
P_PRODUCT_ID, P_DESCRIPTION,
P_QUANTITY FROM HANIF.PRODUCT P INNER JOIN HANIF.ORDER_ITEM OT ON
P.PRODUCT_ID =
OT.PRODUCTID INNER JOIN HANIF.ORDERS O ON OT.ORDERID = O.ORDER_ID AND
ROWNUM=1;
18 END LOOP;
19* END;
SQL> /
PL/SQL procedure successfully completed.
Elapsed: 01:01:27.78
```

The statistics collection is queried again:

```
SQL> select name,value from v$sysstat where name in ('CPU used by
this session', 'sorts
(memory)', 'user I/O wait time', 'redo wastage','enqueue timeouts')
order by value desc;
NAME                              VALUE
```

```
---------------------------------------------------- ----------
CPU used by this session                382931
redo wastage                            177508
sorts (memory)                           22709
user I/O wait time                       1528
enqueue timeouts                         6
5 rows selected.
```

The intensive query had an effect to the degree that the CPU used multiplied by 10.3, redo wastage by 21.56, sorts by 12.87, user i/o wait time increase by 168 and enqueue timeouts increased by 1. Redo wastage benefits sequential i/o pattern but the high redo wastage indicates a very active log writer (Burleson, n.d.b). Sorts memory metric indicates how much the sort is performed completely, no sorts is most beneficial on the workloads (Burleson, n.d.c). CPU used by the session multiplied by 10 times. This is a considerable amount of CPU usage and SQL has optimizer cost tools to counter this (Burlreson, n.d.a). The changes reflect what the DBA would expect from a workload of only reads. Reads don't create locks, therefore enqueue changes by one. The database is restarted and the write workload is ran:

82

Note: The database was restarted after Read workload execution.

Statistics collection before write workload execution:

```
SQL> select name,value from v$sysstat where name in ('CPU used by
this session', 'sorts
(memory)', 'user I/O wait time', 'redo wastage','enqueue timeouts')
order by value desc;
NAME                  VALUE
---------------------------------------------------------------
----------
redo wastage                        29628
sorts (memory)                16467
CPU used by this session                   1247
user I/O wait time                  1175
```

```
enqueue timeouts                 5
5 rows selected.
Elapsed: 00:00:00.02
```

```
SQL> DROP TABLE HANIF.ORDER_ITEM_1 CASCADE CONSTRAINTS PURGE;
Table dropped.
Elapsed: 00:00:00.60
SQL> CREATE TABLE HANIF.ORDER_ITEM_1 AS SELECT * FROM
HANIF.ORDER_ITEM WHERE 1=2;
Table created.
Elapsed: 00:00:00.22
SQL> TRUNCATE TABLE HANIF.ORDER_ITEM_1;
Table truncated.
Elapsed: 00:00:00.09
SQL> SET SERVEROUTPUT ON
SQL> BEGIN
INSERT INTO HANIF.ORDER_ITEM_1 SELECT * FROM HANIF.ORDER_ITEM CONNECT
BY LEVEL <= 2;
DBMS_OUTPUT.PUT_LINE('ROWS INSERTED = '||SQL%ROWCOUNT);
COMMIT;
END;
/
ROWS INSERTED = 1211100
PL/SQL procedure successfully completed.
Elapsed: 00:01:48.10
SYS>
```

The statistics collection is queried again.

```
SQL> select name,value from v$sysstat where name in ('CPU used by
this session', 'sorts
(memory)', 'user I/O wait time', 'redo wastage','enqueue timeouts')
order by value desc;
NAME                                                  VALUE
---------------------------------------------------- ----------
redo wastage                                          204672
sorts (memory)                                        25777
CPU used by this session                              13769
user I/O wait time                                    1572
```

```
enqueue timeouts                 5
5 rows selected.
Elapsed: 00:00:00.03
```

The intensive query had an effect to the degree that the CPU used multiplied by 6.93, redo wastage by 1.57, sorts by 7.8, user i/o wait time increase by 325 and enqueue timeouts increased by 1. User i/o wait time increased two times more than it did with the read workload and enqueue remained the same, outside of that the read workload made for a more intensive query than the write workload. Regardless their values increased. No lock is created so the enqueue timeouts remained static.

Performance Tuning with Automatic Database Diagnostic Monitor (ADDM)

Since Oracle 10g, the Automatic Database Diagnostic Monitor (ADDM) is a utility introduced in Oracle 10g showcasing overview of the most prominent performance bottlenecks while offering suggestions for improvement (Oracle, 2019). This feature is utilized for the order management system:

```
SQL> set long 1000000
SQL> set pagesize 50000
SQL> column get_clob format a80
SQL> select dbms_advisor.get_task_report(task_name, 'TEXT', 'ALL')
as ADDM_report from dba_advisor_tasks where task_id=(select
max(t.task_id) from dba_advisor_tasks t,dba_advisor_log l where
t.task_id = l.task_id and t.advisor_name='ADDM'and l.status=
'COMPLETED');
ADDM_REPORT
--------------------------------------------------------------------------
-----------
ADDM Report for Task 'ADDM:992067541_1_95'
------------------------------------------
Analysis Period
---------------
AWR snapshot range from 94 to 95.
Time period starts at 27-OCT-22 02.00.04 AM
```

```
Time period ends at 27-OCT-22 02.04.18 AM
Analysis Target
---------------
Database 'DBST670' with DB ID 992067541.
Database version 19.0.0.0.0.
ADDM performed an analysis of instance DBST670, numbered 1 and hosted
at
NIXORA01.
Activity During the Analysis Period
---------------------------------
Total database time was 0 seconds.
The average number of active sessions was 0.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~There are no findings to report.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~
Additional Information
----------------------
Miscellaneous Information
-------------------------
There was no significant database activity to run the ADDM.
1 row selected.
Elapsed: 00:00:00.13
```

The read workload is ran, and the ADDM is subsequently initialized.

```
ADDM_REPORT
--------------------------------------------------------------------------
-----------
ADDM Report for Task 'ADDM:992067541_1_96'
-----------------------------------------
Analysis Period
---------------
AWR snapshot range from 95 to 96.
Time period starts at 27-OCT-22 02.04.18 AM
Time period ends at 27-OCT-22 03.00.11 AM
```

```
Analysis Target
---------------
Database 'DBST670' with DB ID 992067541.
Database version 19.0.0.0.0.
ADDM performed an analysis of instance DBST670, numbered 1 and hosted
at
NIXORA01.
Activity During the Analysis Period
-----------------------------------
Total database time was 1459 seconds.
The average number of active sessions was .44.
Summary of Findings
-------------------
Description Active Sessions Recommendations
Percent of Activity
------------------ ------------------- ---------------
1 Top SQL Statements .4 | 91.22 5
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~
Findings and Recommendations
---------------------------
Finding 1: Top SQL Statements
Impact is .4 active sessions, 91.22% of total activity.
-------------------------------------------------------
SQL statements consuming significant database time were found. These
statements offer a good opportunity for performance improvement.
Recommendation 1: SQL Tuning
Estimated benefit is .18 active sessions, 40.54% of total activity.
-------------------------------------------------------------------
Action
Run SQL Tuning Advisor on the SELECT statement with SQL_ID
"8h4k7hum8jgz4".
Related Object
SQL statement with SQL_ID 8h4k7hum8jgz4.
SELECT O.ORDER_ID, O.CUSTOMERID, O.ORDER_METHOD, P.PRODUCT_ID,
P.DESCRIPTION, OT.QUANTITY FROM HANIF.PRODUCT P, HANIF.ORDER_ITEM
OT, HANIF.ORDERS O WHERE ROWNUM=1
```

```
Rationale
The SQL spent 100% of its database time on CPU, I/O and Cluster
waits.
This part of database time may be improved by the SQL Tuning Advisor.
Rationale
Database time for this SQL was divided as follows: 100% for SQL
execution, 0% for parsing, 0% for PL/SQL execution and 0% for Java
execution.
Rationale
SQL statement with SQL_ID "8h4k7hum8jgz4" was executed 30000000 times
and had an average elapsed time of 0.000017 seconds.
Rationale
Top level calls to execute the PL/SQL statement with SQL_ID
"ba9c6wwn4rm1d" are responsible for 63% of the database time spent on
the SELECT statement with SQL_ID "8h4k7hum8jgz4".
Related Object
SQL statement with SQL_ID ba9c6wwn4rm1d.
DECLARE
P_CUSTOMER_ID
HANIF.CUSTOMER.CUSTOMER_ID%TYPE;
P_FIRSTNAME
HANIF.CUSTOMER.FIRSTNAME%TYPE;
P_LASTNAME
HANIF.CUSTOMER.LASTNAME%TYPE;
P_ADDRESS
HANIF.CUSTOMER.ADDRESS%TYPE;
P_ORDER_ID
HANIF.ORDERS.ORDER_ID%TYPE;
P_ORDER_METHOD
HANIF.ORDERS.ORDER_METHOD%TYPE;
P_CUSTOMERID
HANIF.ORDERS.CUSTOMERID%TYPE;
P_PRODUCT_ID
HANIF.PRODUCT.PRODUCT_ID%TYPE;
P_DESCRIPTION
HANIF.PRODUCT.DESCRIPTION%TYPE;
P_QUANTITY
HANIF.ORDER_ITEM.QUANTITY%TYPE;
i NUMBER;
```

```
BEGIN
i :=1;
FOR i IN 1..20000000 LOOP
SELECT
C.CUSTOMER_ID, C.FIRSTNAME, C.LASTNAME, C.ADDRESS, O.ORDER_ID,
O.ORDER_METHOD INTO P_CUSTOMER_ID, P_FIRSTNAME, P_LASTNAME,
P_ADDRESS, P_ORDER_ID, P_ORDER_METHOD FROM HANIF.CUSTOMER C,
HANIF.ORDERS O WHERE ROWNUM=1;
SELECT O.ORDER_ID, O.CUSTOMERID, O.ORDER_METHOD, P.PRODUCT_ID,
P.DESCRIPTION, OT.QUANTITY INTO P_ORDER_ID, P_CUSTOMERID,
P_ORDER_METHOD, P_PRODUCT_ID, P_DESCRIPTION, P_QUANTITY FROM
HANIF.PRODUCT P, HANIF.ORDER_ITEM OT, HANIF.ORDERS O WHERE
ROWNUM=1;
END LOOP;
END;
Recommendation 2: SQL Tuning
Estimated benefit is .1 active sessions, 22.97% of total activity.
-------------------------------------------------------------------
Action
Run SQL Tuning Advisor on the SELECT statement with SQL_ID
"gmqm0a80cd8br".
Related Object
SQL statement with SQL_ID gmqm0a80cd8br.
SELECT C.CUSTOMER_ID, C.FIRSTNAME, C.LASTNAME, C.ADDRESS, O.ORDER_ID,
O.ORDER_METHOD FROM HANIF.CUSTOMER C, HANIF.ORDERS O WHERE ROWNUM=1
Rationale
The SQL spent 100% of its database time on CPU, I/O and Cluster
waits.
This part of database time may be improved by the SQL Tuning Advisor.
Rationale
Database time for this SQL was divided as follows: 100% for SQL
execution, 0% for parsing, 0% for PL/SQL execution and 0% for Java
execution.
Rationale
SQL statement with SQL_ID "gmqm0a80cd8br" was executed 30000000 times
and had an average elapsed time of 0.000009 seconds.
Rationale
Top level calls to execute the PL/SQL statement with SQL_ID
"ba9c6wwn4rm1d" are responsible for 67% of the database time spent on
```

```
the SELECT statement with SQL_ID "gmqm0a80cd8br".
Related Object
SQL statement with SQL_ID ba9c6wwn4rm1d.
DECLARE
P_CUSTOMER_ID
HANIF.CUSTOMER.CUSTOMER_ID%TYPE;
P_FIRSTNAME
HANIF.CUSTOMER.FIRSTNAME%TYPE;
P_LASTNAME
HANIF.CUSTOMER.LASTNAME%TYPE;
P_ADDRESS
HANIF.CUSTOMER.ADDRESS%TYPE;
P_ORDER_ID
HANIF.ORDERS.ORDER_ID%TYPE;
P_ORDER_METHOD
HANIF.ORDERS.ORDER_METHOD%TYPE;
P_CUSTOMERID
HANIF.ORDERS.CUSTOMERID%TYPE;
P_PRODUCT_ID
HANIF.PRODUCT.PRODUCT_ID%TYPE;
P_DESCRIPTION
HANIF.PRODUCT.DESCRIPTION%TYPE;
P_QUANTITY
HANIF.ORDER_ITEM.QUANTITY%TYPE;
i NUMBER;
BEGIN
i :=1;
FOR i IN 1..20000000 LOOP
SELECT
C.CUSTOMER_ID, C.FIRSTNAME, C.LASTNAME, C.ADDRESS, O.ORDER_ID,
O.ORDER_METHOD INTO P_CUSTOMER_ID, P_FIRSTNAME, P_LASTNAME,
P_ADDRESS, P_ORDER_ID, P_ORDER_METHOD FROM HANIF.CUSTOMER C,
HANIF.ORDERS O WHERE ROWNUM=1;
SELECT O.ORDER_ID, O.CUSTOMERID, O.ORDER_METHOD, P.PRODUCT_ID,
P.DESCRIPTION, OT.QUANTITY INTO P_ORDER_ID, P_CUSTOMERID,
P_ORDER_METHOD, P_PRODUCT_ID, P_DESCRIPTION, P_QUANTITY FROM
HANIF.PRODUCT P, HANIF.ORDER_ITEM OT, HANIF.ORDERS O WHERE
ROWNUM=1;
END LOOP;
```

```
END;
Recommendation 3: SQL Tuning
Estimated benefit is .06 active sessions, 13.51% of total activity.
-------------------------------------------------------------------------
Action
Run SQL Tuning Advisor on the SELECT statement with SQL_ID
"bcg8x196y0n41".
Related Object
SQL statement with SQL_ID bcg8x196y0n41.
SELECT C.CUSTOMER_ID, C.FIRSTNAME, C.LASTNAME, C.ADDRESS, O.ORDER_ID,
O.ORDER_METHOD FROM HANIF.CUSTOMER C INNER JOIN HANIF.ORDERS O ON
C.CUSTOMER_ID = O.CUSTOMERID AND ROWNUM=1
Rationale
The SQL spent 100% of its database time on CPU, I/O and Cluster
waits.
This part of database time may be improved by the SQL Tuning Advisor.
Rationale
Database time for this SQL was divided as follows: 100% for SQL
execution, 0% for parsing, 0% for PL/SQL execution and 0% for Java
execution.
Rationale
Top level calls to execute the PL/SQL statement with SQL_ID
"57gx14s4sjsah" are responsible for 100% of the database time spent
on
the SELECT statement with SQL_ID "bcg8x196y0n41".
Related Object
SQL statement with SQL_ID 57gx14s4sjsah.
Recommendation 4: SQL Tuning
Estimated benefit is .03 active sessions, 7.43% of total activity.
-------------------------------------------------------------------------
Action
Investigate the PL/SQL statement with SQL_ID "ba9c6wwn4rm1d" for
possible performance improvements. You can supplement the information
given here with an ASH report for this SQL_ID.
Related Object
SQL statement with SQL_ID ba9c6wwn4rm1d.
DECLARE
P_CUSTOMER_ID
HANIF.CUSTOMER.CUSTOMER_ID%TYPE;
```

```
P_FIRSTNAME
HANIF.CUSTOMER.FIRSTNAME%TYPE;
P_LASTNAME
HANIF.CUSTOMER.LASTNAME%TYPE;
P_ADDRESS
HANIF.CUSTOMER.ADDRESS%TYPE;
P_ORDER_ID
HANIF.ORDERS.ORDER_ID%TYPE;
101
P_ORDER_METHOD
HANIF.ORDERS.ORDER_METHOD%TYPE;
P_CUSTOMERID
HANIF.ORDERS.CUSTOMERID%TYPE;
P_PRODUCT_ID
HANIF.PRODUCT.PRODUCT_ID%TYPE;
P_DESCRIPTION
HANIF.PRODUCT.DESCRIPTION%TYPE;
P_QUANTITY
HANIF.ORDER_ITEM.QUANTITY%TYPE;
i NUMBER;
BEGIN
i :=1;
FOR i IN 1..20000000 LOOP
SELECT
C.CUSTOMER_ID, C.FIRSTNAME, C.LASTNAME, C.ADDRESS, O.ORDER_ID,
O.ORDER_METHOD INTO P_CUSTOMER_ID, P_FIRSTNAME, P_LASTNAME,
P_ADDRESS, P_ORDER_ID, P_ORDER_METHOD FROM HANIF.CUSTOMER C,
HANIF.ORDERS O WHERE ROWNUM=1;
SELECT O.ORDER_ID, O.CUSTOMERID, O.ORDER_METHOD, P.PRODUCT_ID,
P.DESCRIPTION, OT.QUANTITY INTO P_ORDER_ID, P_CUSTOMERID,
P_ORDER_METHOD, P_PRODUCT_ID, P_DESCRIPTION, P_QUANTITY FROM
HANIF.PRODUCT P, HANIF.ORDER_ITEM OT, HANIF.ORDERS O WHERE
ROWNUM=1;
END LOOP;
END;
Rationale
The SQL Tuning Advisor cannot operate on PL/SQL statements.
Rationale
Database time for this SQL was divided as follows: 37% for SQL
```

```
execution, 0% for parsing, 63% for PL/SQL execution and 0% for Java
execution.
Rationale
SQL statement with SQL_ID "ba9c6wwn4rm1d" was executed 1 times and
had
an average elapsed time of 755 seconds.
Recommendation 5: SQL Tuning
Estimated benefit is .03 active sessions, 6.76% of total activity.
---------------------------------------------------------------
Action
Run SQL Tuning Advisor on the SELECT statement with SQL_ID
"3nmpa03ayt7av".
Related Object
SQL statement with SQL_ID 3nmpa03ayt7av.
SELECT O.ORDER_ID, O.CUSTOMERID, O.ORDER_METHOD, P.PRODUCT_ID,
P.DESCRIPTION, OT.QUANTITY FROM HANIF.PRODUCT P INNER JOIN
HANIF.ORDER_ITEM OT ON P.PRODUCT_ID = OT.PRODUCTID INNER JOIN
HANIF.ORDERS O ON OT.ORDERID = O.ORDER_ID AND ROWNUM=1
Rationale
The SQL spent 100% of its database time on CPU, I/O and Cluster
waits.
This part of database time may be improved by the SQL Tuning Advisor.
Rationale
Database time for this SQL was divided as follows: 100% for SQL
execution, 0% for parsing, 0% for PL/SQL execution and 0% for Java
execution.
Rationale
Top level calls to execute the PL/SQL statement with SQL_ID
"57gx14s4sjsah" are responsible for 100% of the database time spent
on
the SELECT statement with SQL_ID "3nmpa03ayt7av".
Related Object
SQL statement with SQL_ID 57gx14s4sjsah.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~
Additional Information
----------------------
```

```
Miscellaneous Information
-------------------------
Wait class "Application" was not consuming significant database time.
Wait class "Commit" was not consuming significant database time.
Wait class "Concurrency" was not consuming significant database time.
Wait class "Configuration" was not consuming significant database
time.
CPU was not a bottleneck for the instance.
Wait class "Network" was not consuming significant database time.
Wait class "User I/O" was not consuming significant database time.
Session connect and disconnect calls were not consuming significant
database
time.
Hard parsing of SQL statements was not consuming significant database
time.
1 row selected.
Elapsed: 00:00:00.16
```

The SQL statements consumed a significant amount of database time and CPU. SQL tuning is recommending to tune the read query causing such a strain. For comparative studies, the write workload is ran. The ADDM report is below.

```
ADDM_REPORT
--------------------------------------------------------------------------------
----------------------
ADDM Report for Task 'ADDM:992067541_1_101'
-------------------------------------------
Analysis Period
---------------
AWR snapshot range from 100 to 101.
Time period starts at 27-OCT-22 10.06.06 AM
Time period ends at 27-OCT-22 10.18.27 AM
Analysis Target
---------------
Database 'DBST670' with DB ID 992067541.
Database version 19.0.0.0.0.
ADDM performed an analysis of instance DBST670, numbered 1 and hosted
at
```

```
NIXORA01.
Activity During the Analysis Period
-----------------------------------
Total database time was 303 seconds.
The average number of active sessions was .41.
Summary of Findings
-------------------
Description Active Sessions Recommendations
Percent of Activity
------------------------------ ------------------ --------------
-
1 Top SQL Statements .41 | 100 1
2 Session Connect and Disconnect .01 | 2.6 1
3 Hard Parse .01 | 2.35 0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~
Findings and Recommendations
---------------------------
Finding 1: Top SQL Statements
Impact is .41 active sessions, 100% of total activity.
-----------------------------------------------------
SQL statements consuming significant database time were found. These
statements offer a good opportunity for performance improvement.
Recommendation 1: SQL Tuning
Estimated benefit is .41 active sessions, 100% of total activity.
-----------------------------------------------------------------
Action
Run SQL Tuning Advisor on the INSERT statement with SQL_ID
"7ucjjhygs7abw".
Related Object
SQL statement with SQL_ID 7ucjjhygs7abw.
Rationale
The SQL spent 100% of its database time on CPU, I/O and Cluster
waits.
This part of database time may be improved by the SQL Tuning
Advisor.
Rationale
```

```
Database time for this SQL was divided as follows: 100% for SQL
execution, 0% for parsing, 0% for PL/SQL execution and 0% for
Java
execution.
Rationale
Top level calls to execute the PL/SQL statement with SQL_ID
"ar6t9h4a8sdah" are responsible for 100% of the database time
spent on
the INSERT statement with SQL_ID "7ucjjhygs7abw".
Related Object
SQL statement with SQL_ID ar6t9h4a8sdah.
Finding 2: Session Connect and Disconnect
Impact is .01 active sessions, 2.6% of total activity.
-----------------------------------------------------------
Session connect and disconnect calls were consuming significant
database time.
Recommendation 1: Application Analysis
Estimated benefit is .01 active sessions, 2.6% of total activity.
-----------------------------------------------------------------
Action
Investigate application logic for possible reduction of connect
and
disconnect calls. For example, you might use a connection pool
scheme in
the middle tier.
Finding 3: Hard Parse
Impact is .01 active sessions, 2.35% of total activity.
-----------------------------------------------------------
Hard parsing of SQL statements was consuming significant database
time.
Hard parses due to cursor environment mismatch were not consuming
significant
database time.
Hard parsing SQL statements that encountered parse errors was not
consuming
significant database time.
Hard parses due to literal usage and cursor invalidation were not
consuming
significant database time.
```

```
The SGA was adequately sized.
No recommendations are available.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~
Additional Information
----------------------
Miscellaneous Information
-------------------------
Wait class "Application" was not consuming significant database time.
Wait class "Commit" was not consuming significant database time.
Wait class "Concurrency" was not consuming significant database time.
Wait class "Configuration" was not consuming significant database
time.
CPU was not a bottleneck for the instance.
Wait class "Network" was not consuming significant database time.
Wait class "User I/O" was not consuming significant database time.
1 row selected.
```

The SQL tuning advisor is ran per the recommendations.

```
SQL> VARIABLE t VARCHAR2(20);
BEGIN
:t := DBMS_SQLTUNE.create_tuning_task(sql_id => '&SQL_ID');
DBMS_SQLTUNE.execute_tuning_task(:t);
END;
/SQL> SQL> 2 3 4 5 6 7 8 9
Enter value for sql_id: 8h4k7hum8jgz4
old 3: :t := DBMS_SQLTUNE.create_tuning_task(sql_id => '&SQL_ID');
new 3: :t := DBMS_SQLTUNE.create_tuning_task(sql_id =>
'8h4k7hum8jgz4');
PL/SQL procedure successfully completed.
Elapsed: 00:00:00.67
SQL>
SQL> SET LINES 32767 PAGES 0 TRIMSPOOL ON VERIFY OFF LONG 1000000
LONGC 1000000
SELECT DBMS_SQLTUNE.report_tuning_task (task_name => :t, TYPE =>
'TEXT') FROM DUAL;SQL>
SQL>
```

```
GENERAL INFORMATION SECTION
-------------------------------------------------------------------------------
Tuning Task Name : TASK_394
Tuning Task Owner : SYS
Workload Type : Single SQL Statement
Scope : COMPREHENSIVE
Time Limit(seconds): 1800
Completion Status : COMPLETED
Started at : 10/27/2022 03:09:33
Completed at : 10/27/2022 03:09:34
-------------------------------------------------------------------------------
Schema Name: SYS
SQL ID : 8h4k7hum8jgz4
SQL Text : SELECT O.ORDER_ID, O.CUSTOMERID, O.ORDER_METHOD,
P.PRODUCT_ID,
P.DESCRIPTION, OT.QUANTITY FROM HANIF.PRODUCT P,
HANIF.ORDER_ITEM OT, HANIF.ORDERS O WHERE ROWNUM=1
-------------------------------------------------------------------------------
There are no recommendations to improve the statement.
```

As stated by Oracle, the SQL Tuning Advisor does not provide recommendations to improve the query. This is good to report, since the DBA can assume that their query is the most efficient it can be.

Performance Testing Matrix

DB Name	Datetime of Check	No. long running queries	No. long running RMAN Job	No. Blocking Sessions	DB Size, free space, and used space	Free space for tablespace	% multipass execution and PGA status	Shared pool buffer cache status	Outstanding Alerts	Total users and their type	Topmost wait events, DB time, processing time, and wait
DBST670	10/30/2022 5:00 PM	1	1	1	3gb, 1gb free, 2gb used	All tables have space	0%, PGA is ok	Sufficient	No	33 users 30 background, 3 foreground	SQL*Net Message from client DB Time 942 wait time 935, processing time 7

Conclusion

The order data both stored in the virtual machine and imported undergo integration into the Oracle instance implemented on to the Linux machine. Order data is some of the most popular data in the world resulting in order management system popularity. The database administrator is tasked with many steps in order database implementation including planning, importing data, backup, recovery, statistical analysis, optimization, and maintenance. Oracle 19c on Linux is used to host the database instance. Database performance is confirmed and optimized.

References

Amulya, M. P. (2020). Progression of Order Management System for Digital Marketing. *International Journal of Advanced Research in Computer and Communication Engineering*, *9*(4), 109-116. https://doi.org/10.17148/IJARCCE.2020.9418

Oracle. (n.d.a). Oracle CPU used by this session. Oracle Consulting, Oracle Support and Oracle Training byBC Oracle Consulting. https://www.dba-oracle.com/m_cpu_used_by_this_session.htm

Burleson. (n.d.b). Oracle redo wastage. Oracle Consulting, Oracle Support and Oracle Training by BC Oracle Consulting. https://www.dba-oracle.com/m_redo_wastage.htm

Burleson, D. (n.d.c). Oracle sorts memory. Oracle Consulting, Oracle Support and Oracle Training by BC Oracle Consulting. https://www.dba-oracle.com/m_sorts_memory.htm

Mockaroo. (2022). *Random Data Generator and API Mocking Tool*. https://www.mockaroo.com/

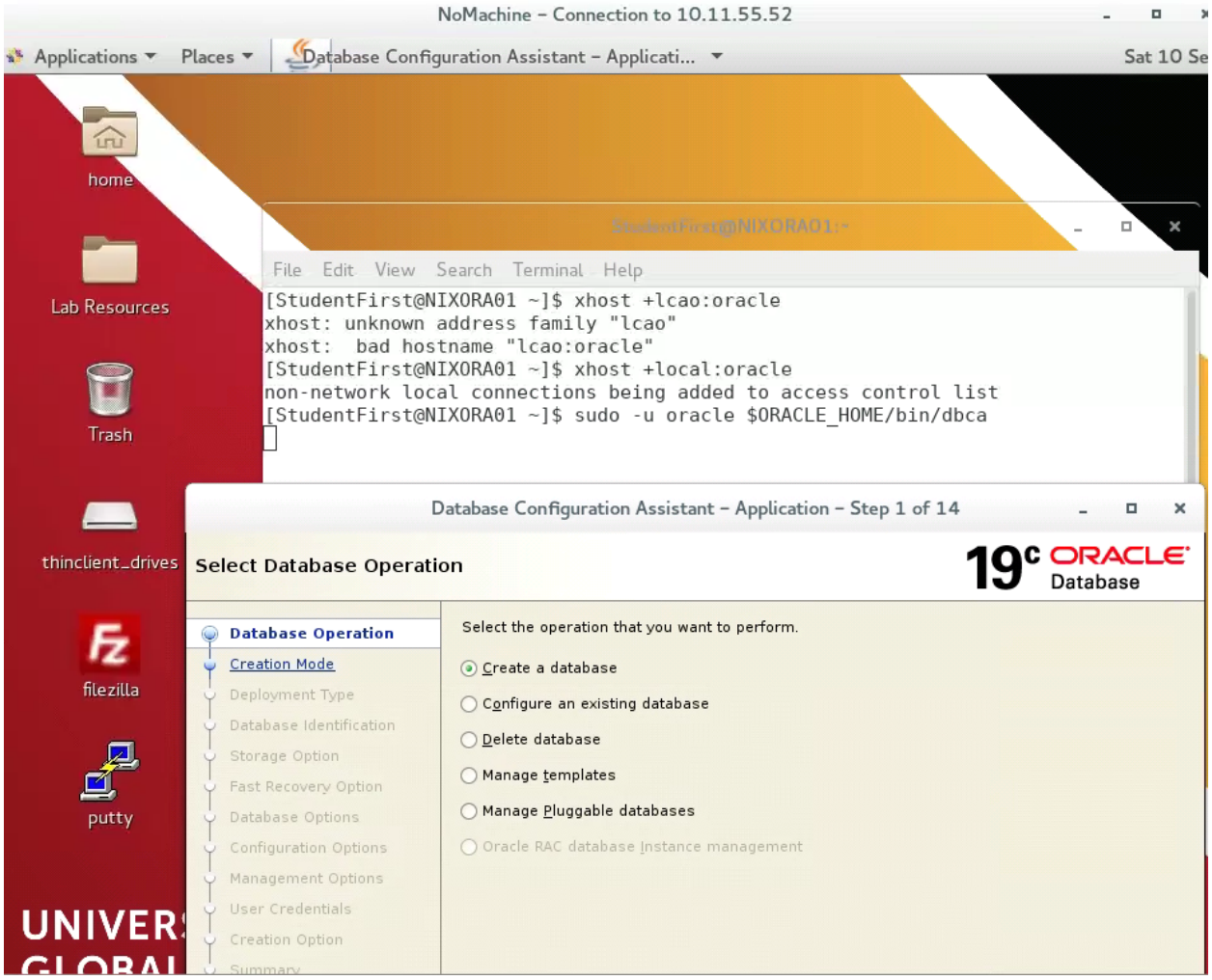
Oracle. (2019, July 31). Automatic database diagnostic monitor (ADDM) in Oracle database 10g. oracle-base.com. https://oracle-base.com/articles/10g/automatic-database-diagnostic-monitor-10g

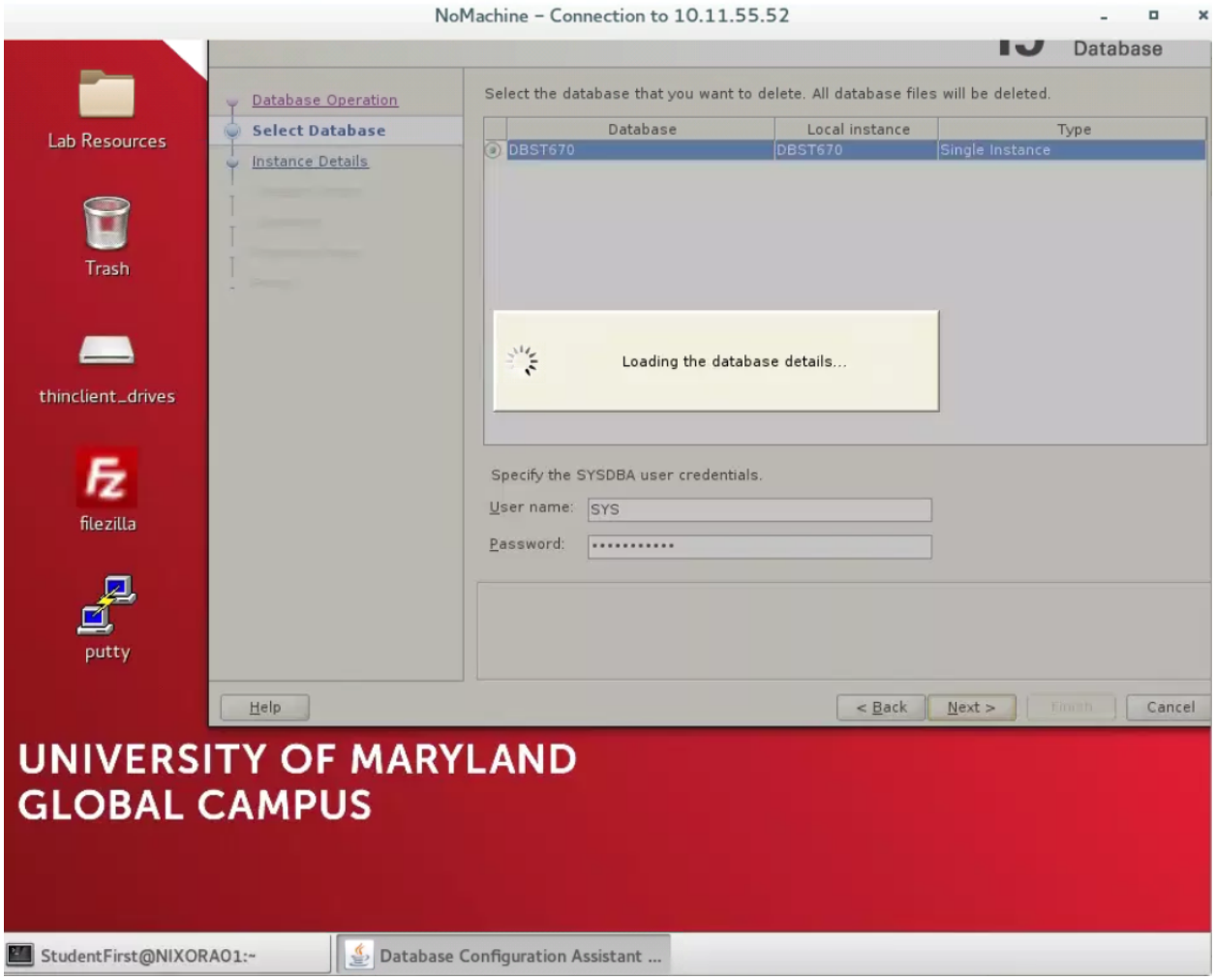
Oracle. (n.d.). V$sysstat. https://docs.oracle.com/cd/E18283_01/server.112/e17110/dynviews_3093.htm

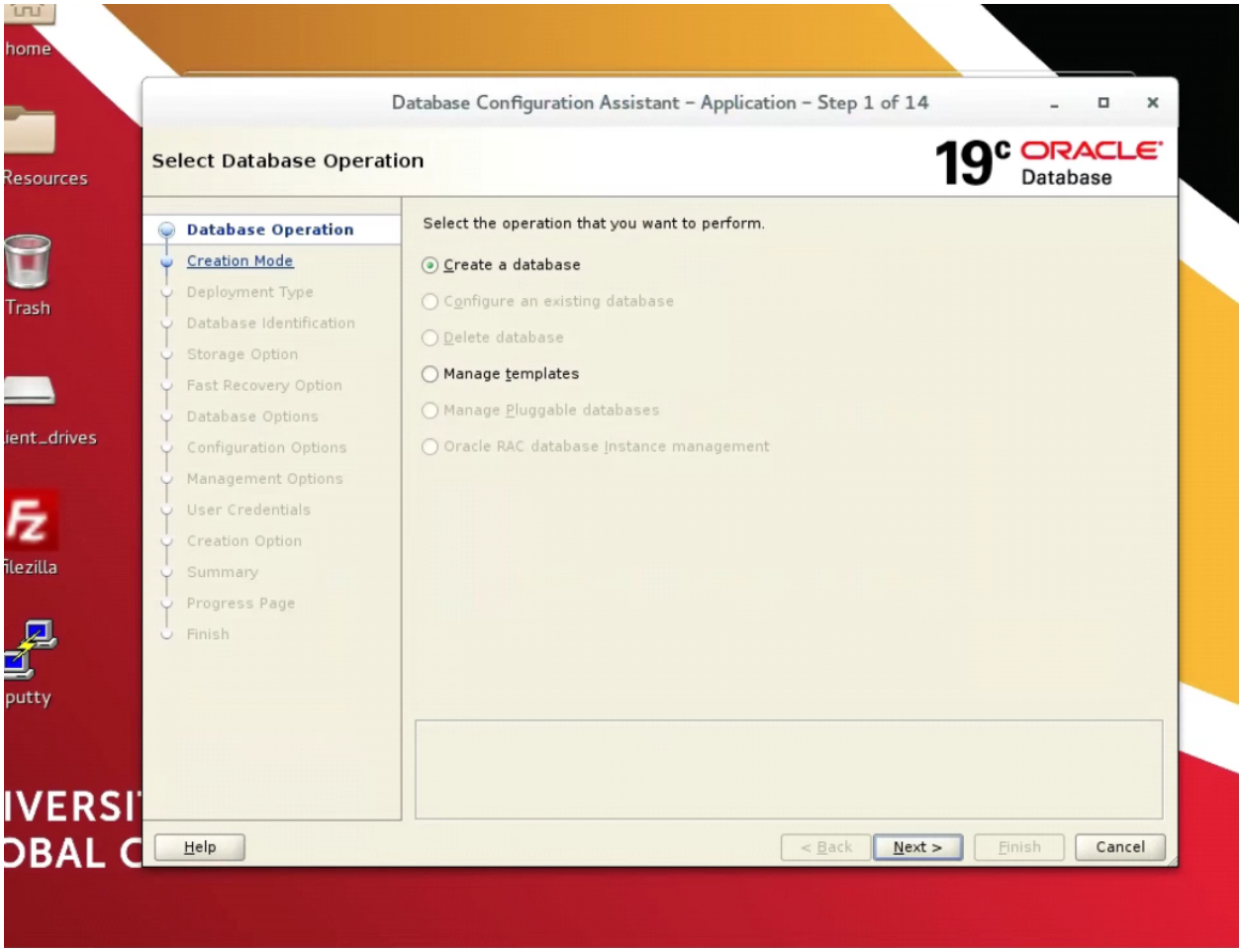
Index

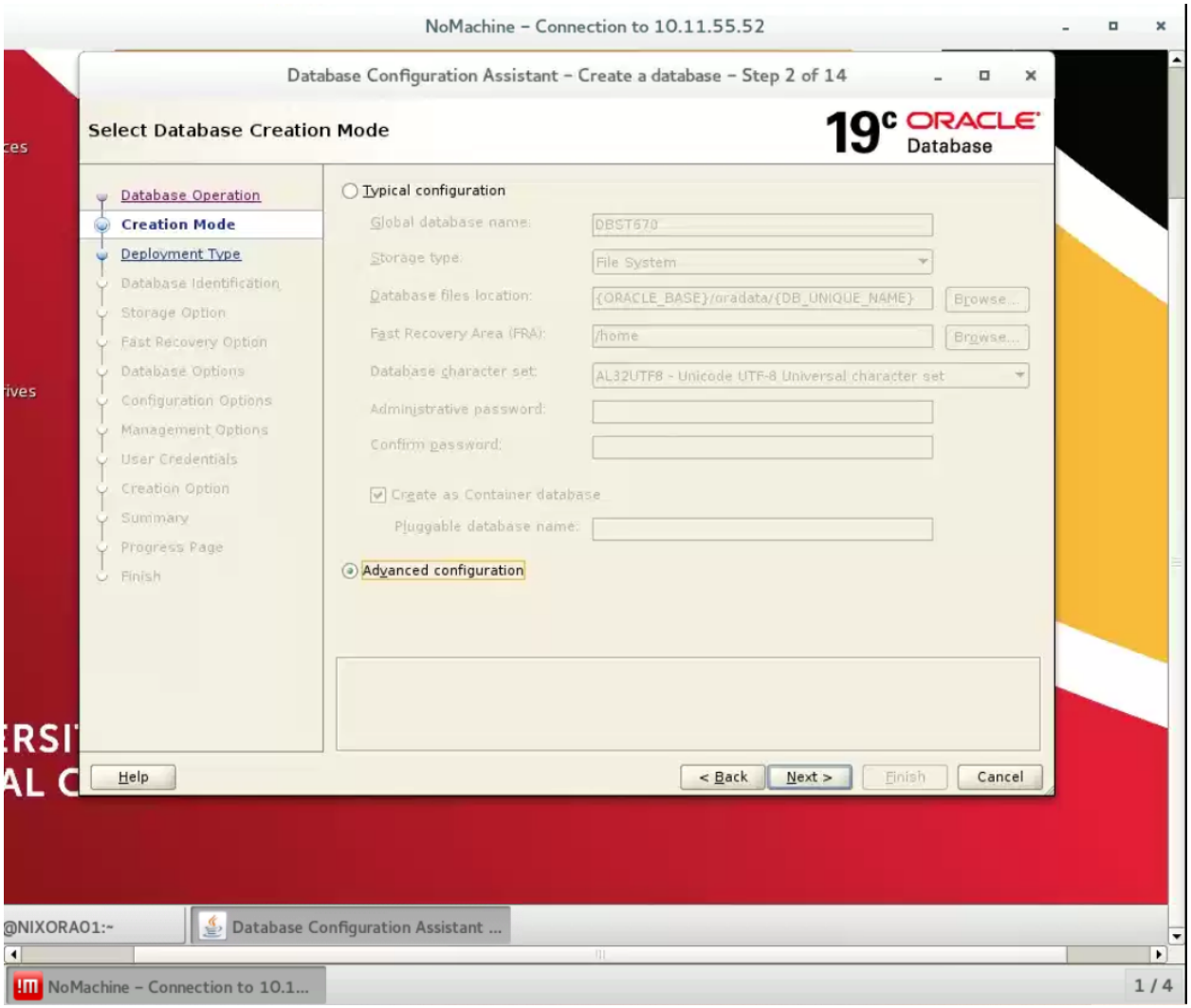
The following is the DDL script generated using Oracle Data Modeler:

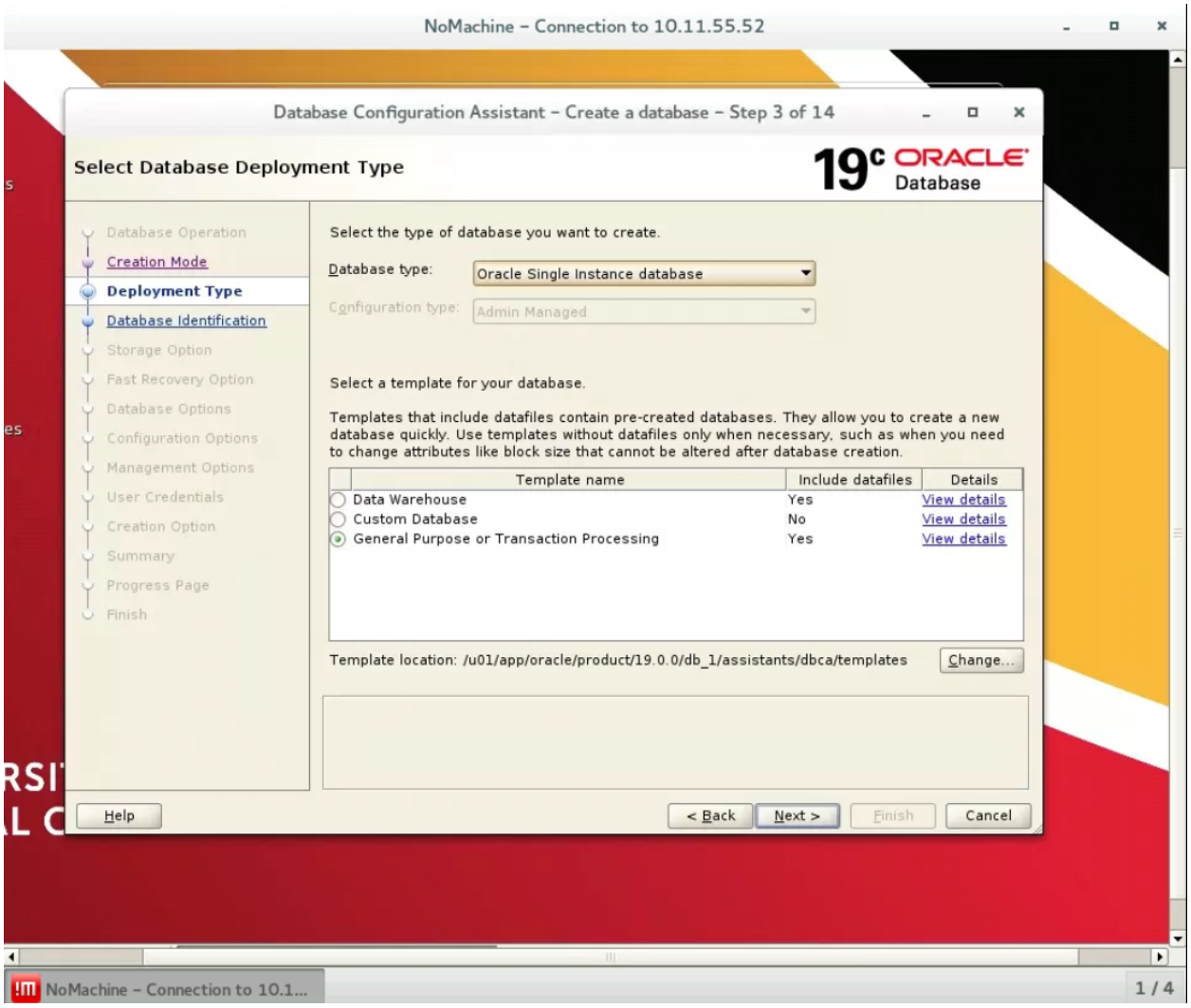
Database Configuration Assistant Wizard Steps

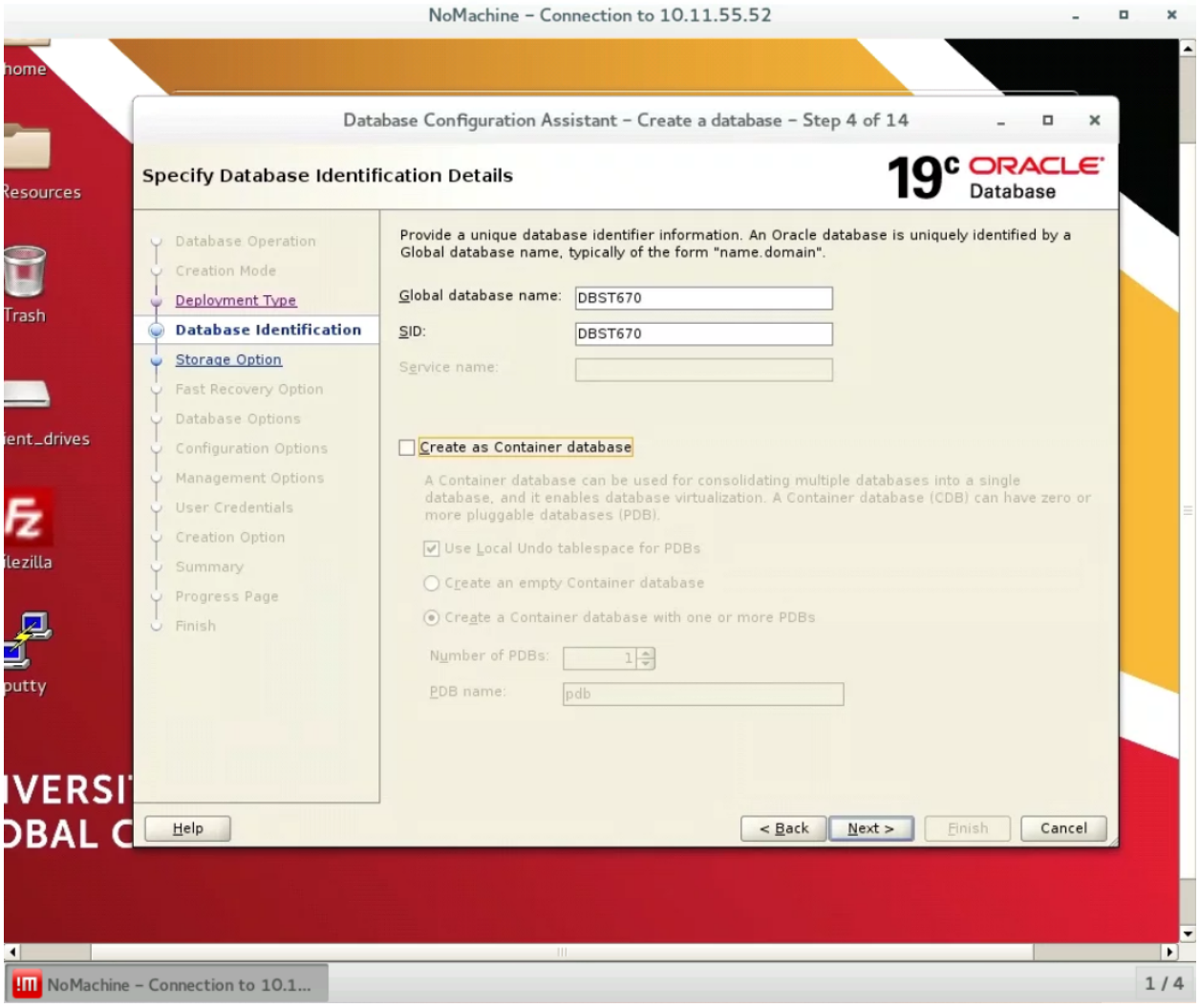


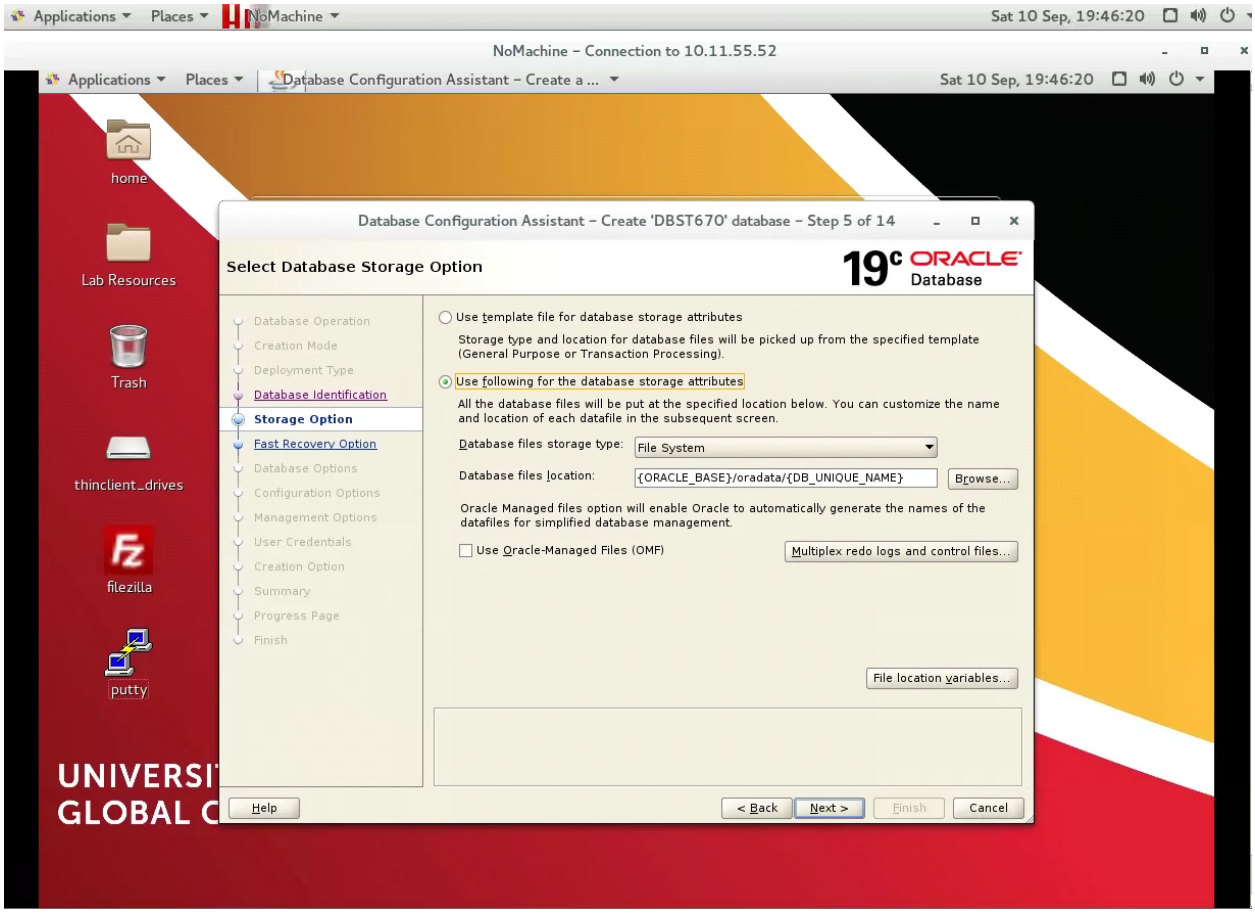

NoMachine – Connection to 10.11.55.52
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database
Database Operation
Select Database
Instance Details
Select the database that you want to delete. All database files will be deleted.
Database
Local instance
Type
DBST670
DBST670
Single Instance
Loading the database details...
Specify the SYSDBA user credentials.
User name:
SYS
Password:
Help
< Back
Next >
Cancel
UNIVERSITY OF MARYLAND
GLOBAL CAMPUS
StudentFirst@NIXORA01:~
Database Configuration Assistant ...

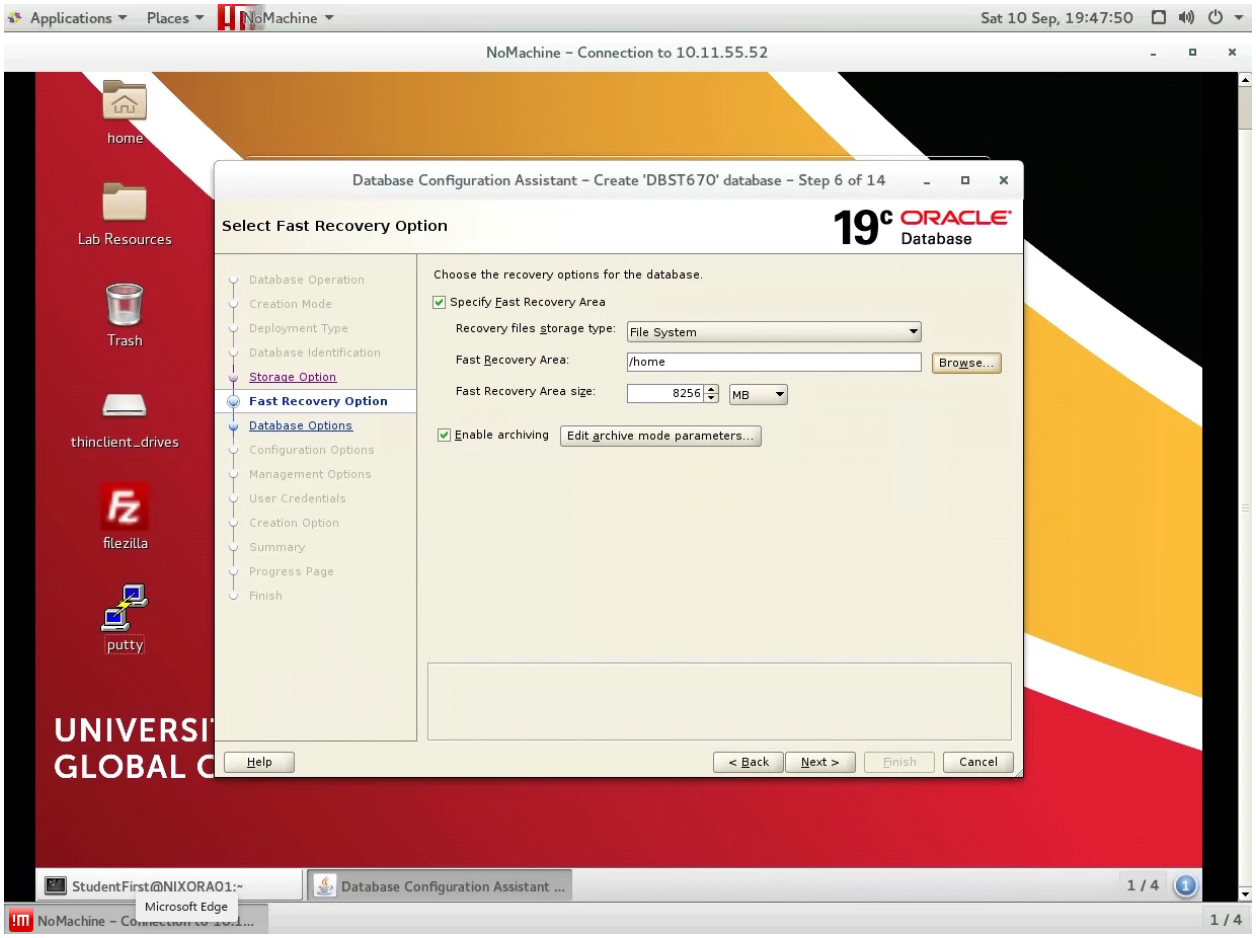

home
Resources
Trash
ient_drives
filezilla
putty
IVERSI
OBAL C
Database Configuration Assistant – Application – Step 1 of 14
Select Database Operation
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Database Options
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Select the operation that you want to perform.
Create a database
Configure an existing database
Delete database
Manage templates
Manage Pluggable databases
Oracle RAC database Instance management
Help
< Back
Next >
Finish
Cancel

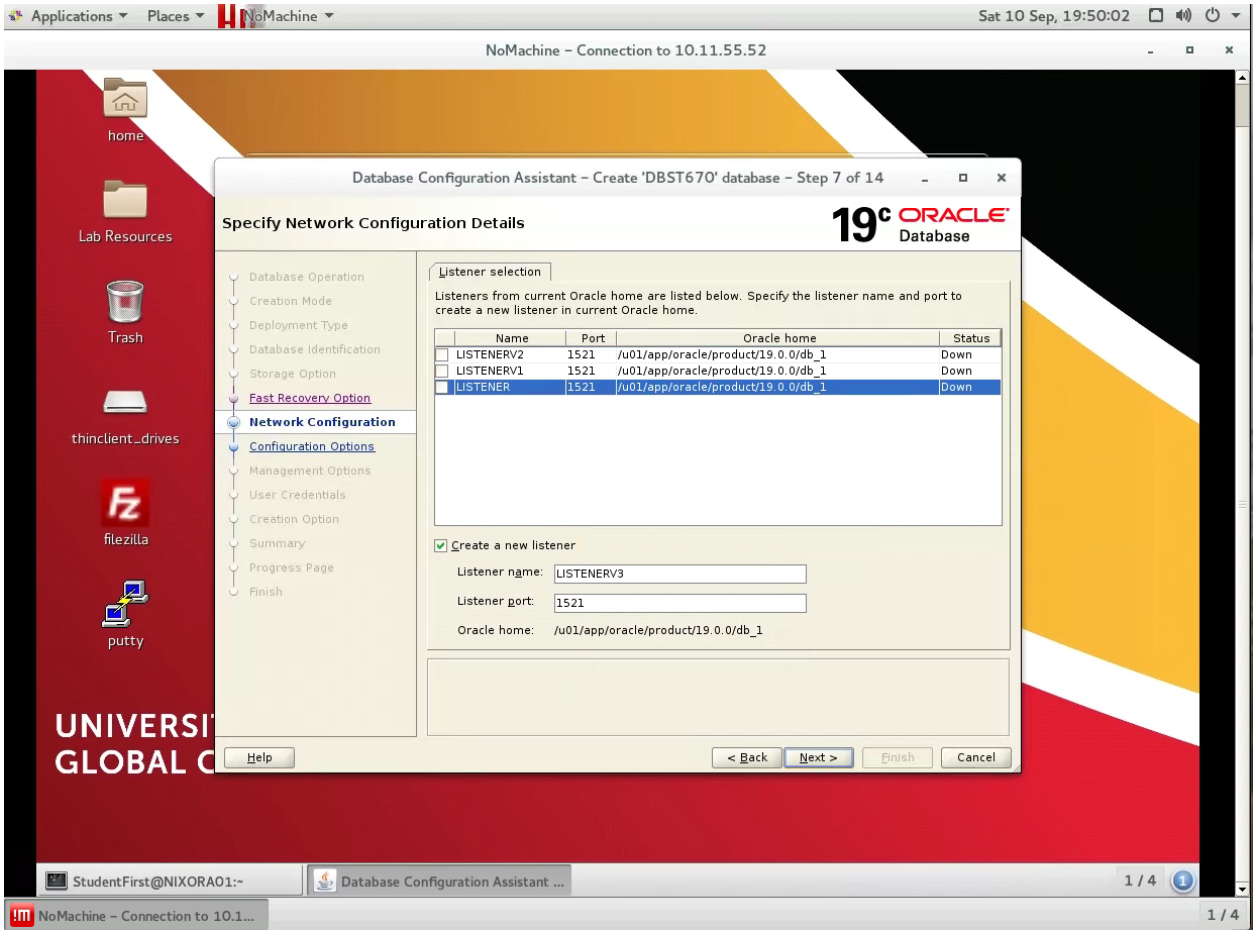

NoMachine – Connection to 10.11.55.52
Database Configuration Assistant – Create a database – Step 2 of 14
Select Database Creation Mode
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Database Options
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Typical configuration
Global database name:
DBST670
Storage type:
File System
Database files location:
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}
Browse...
Fast Recovery Area (FRA):
/home
Browse...
Database character set:
AL32UTF8 - Unicode UTF-8 Universal character set
Administrative password:
Confirm password:
Create as Container database
Pluggable database name:
Advanced configuration
Help
< Back
Next >
Finish
Cancel
@NIXORA01:~
Database Configuration Assistant ...
NoMachine – Connection to 10.1...
1 / 4

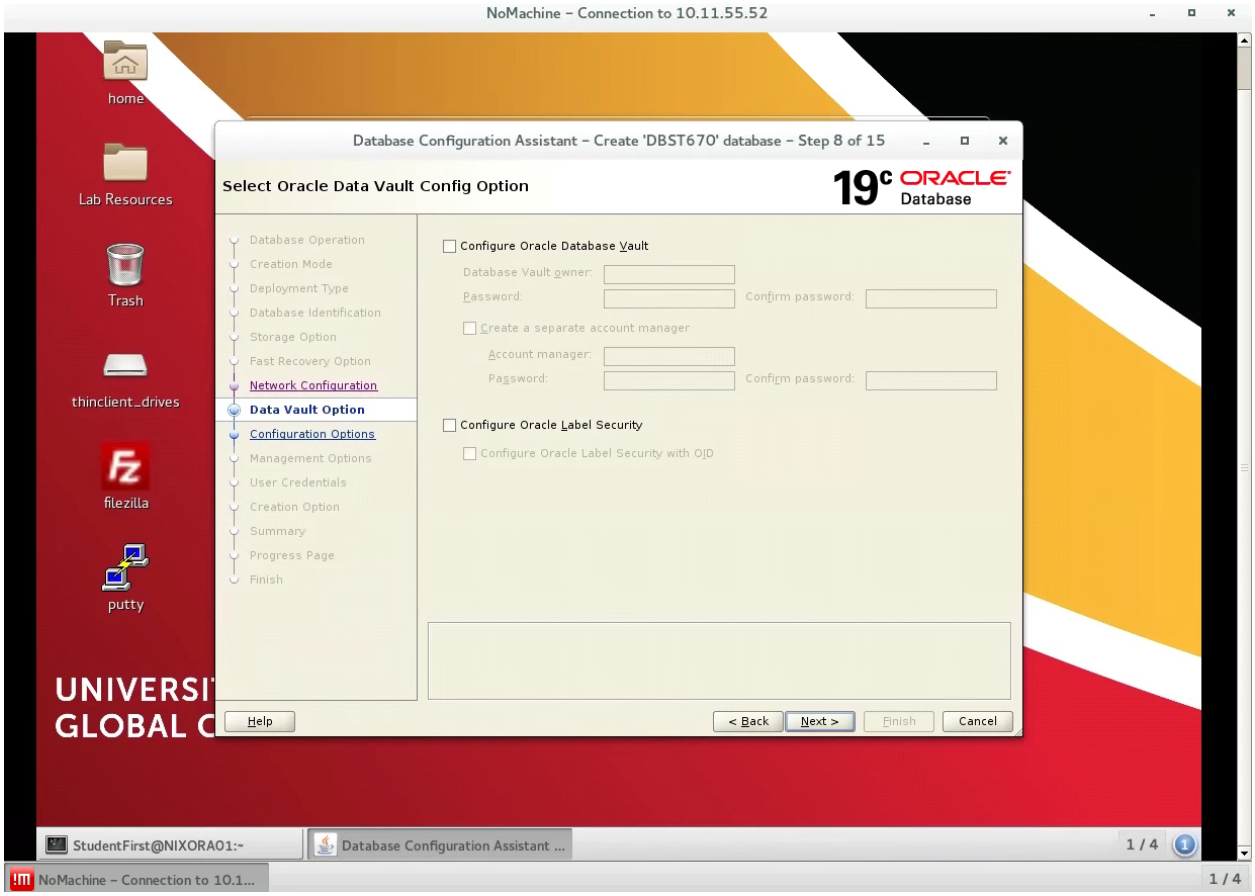

NoMachine – Connection to 10.11.55.52
Database Configuration Assistant – Create a database – Step 3 of 14
Select Database Deployment Type
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Database Options
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Select the type of database you want to create.
Database type:
Oracle Single Instance database
Configuration type:
Admin Managed
Select a template for your database.
Templates that include datafiles contain pre-created databases. They allow you to create a new database quickly. Use templates without datafiles only when necessary, such as when you need to change attributes like block size that cannot be altered after database creation.
Template name
Include datafiles
Details
Data Warehouse
Yes
View details
Custom Database
No
View details
General Purpose or Transaction Processing
Yes
View details
Template location: /u01/app/oracle/product/19.0.0/db_1/assistants/dbca/templates
Change...
Help
< Back
Next >
Finish
Cancel
NoMachine – Connection to 10.1...
1 / 4

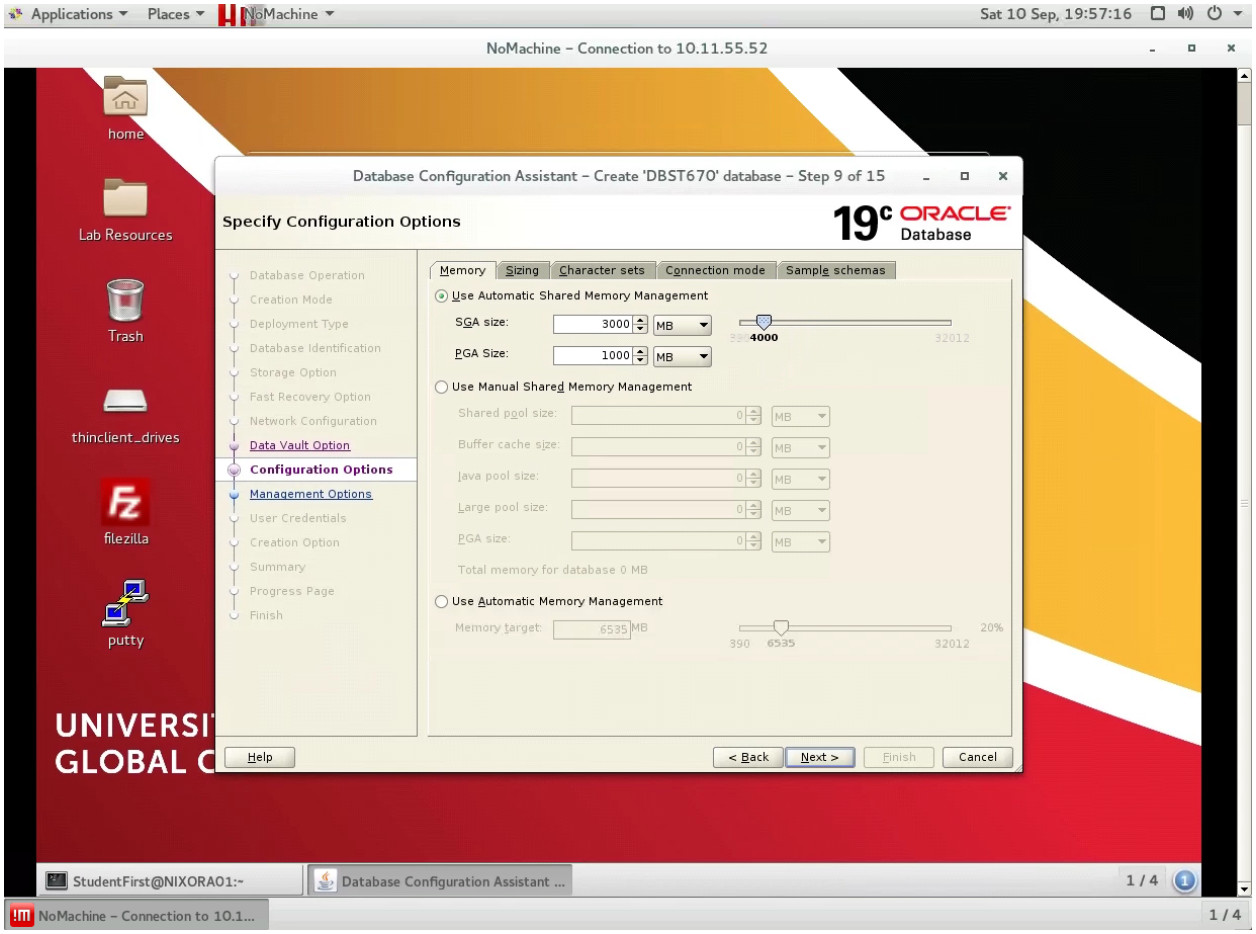

NoMachine – Connection to 10.11.55.52
Database Configuration Assistant – Create a database – Step 4 of 14
Specify Database Identification Details
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Database Options
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Provide a unique database identifier information. An Oracle database is uniquely identified by a Global database name, typically of the form "name.domain".
Global database name: DBST670
SID: DBST670
Service name:
Create as Container database
A Container database can be used for consolidating multiple databases into a single database, and it enables database virtualization. A Container database (CDB) can have zero or more pluggable databases (PDB).
Use Local Undo tablespace for PDBs
Create an empty Container database
Create a Container database with one or more PDBs
Number of PDBs: 1
PDB name: pdb
Help
< Back
Next >
Finish
Cancel
NoMachine – Connection to 10.1...
1 / 4

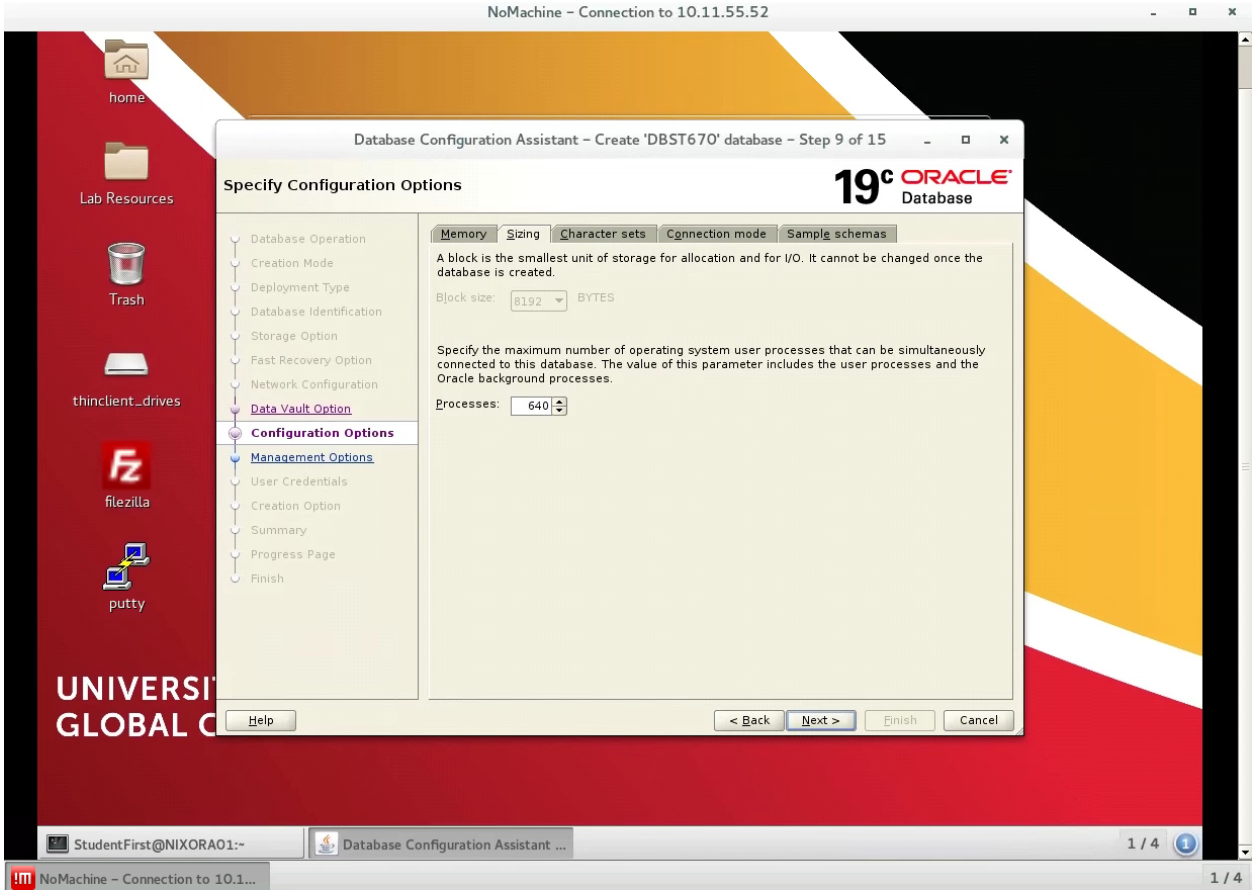

Applications
Places
NoMachine
Sat 10 Sep, 19:46:20
NoMachine – Connection to 10.11.55.52
Applications
Places
Database Configuration Assistant – Create a ...
Sat 10 Sep, 19:46:20
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 5 of 14
Select Database Storage Option
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Database Options
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Use template file for database storage attributes
Storage type and location for database files will be picked up from the specified template (General Purpose or Transaction Processing).
Use following for the database storage attributes
All the database files will be put at the specified location below. You can customize the name and location of each datafile in the subsequent screen.
Database files storage type:
File System
Database files location:
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}
Browse...
Oracle Managed files option will enable Oracle to automatically generate the names of the datafiles for simplified database management.
Use Oracle-Managed Files (OMF)
Multiplex redo logs and control files...
File location variables...
Help
< Back
Next >
Finish
Cancel

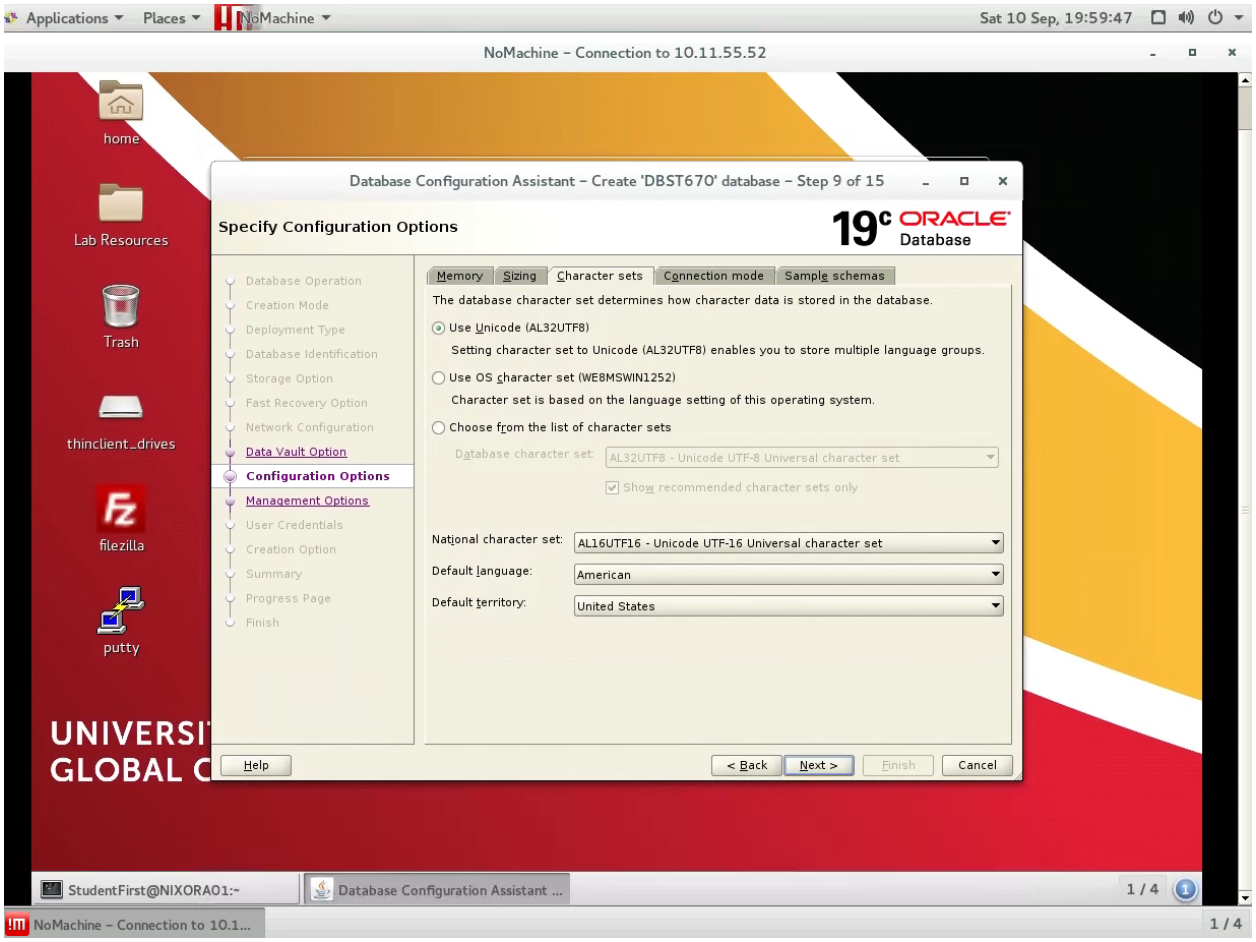

Applications
Places
NoMachine
Sat 10 Sep, 19:47:50
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 6 of 14
Select Fast Recovery Option
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Database Options
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Choose the recovery options for the database.
Specify Fast Recovery Area
Recovery files storage type:
File System
Fast Recovery Area:
/home
Browse...
Fast Recovery Area size:
8256
MB
Enable archiving
Edit archive mode parameters...
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
1 / 4
Microsoft Edge
NoMachine – Connection to 10.1...
1 / 4

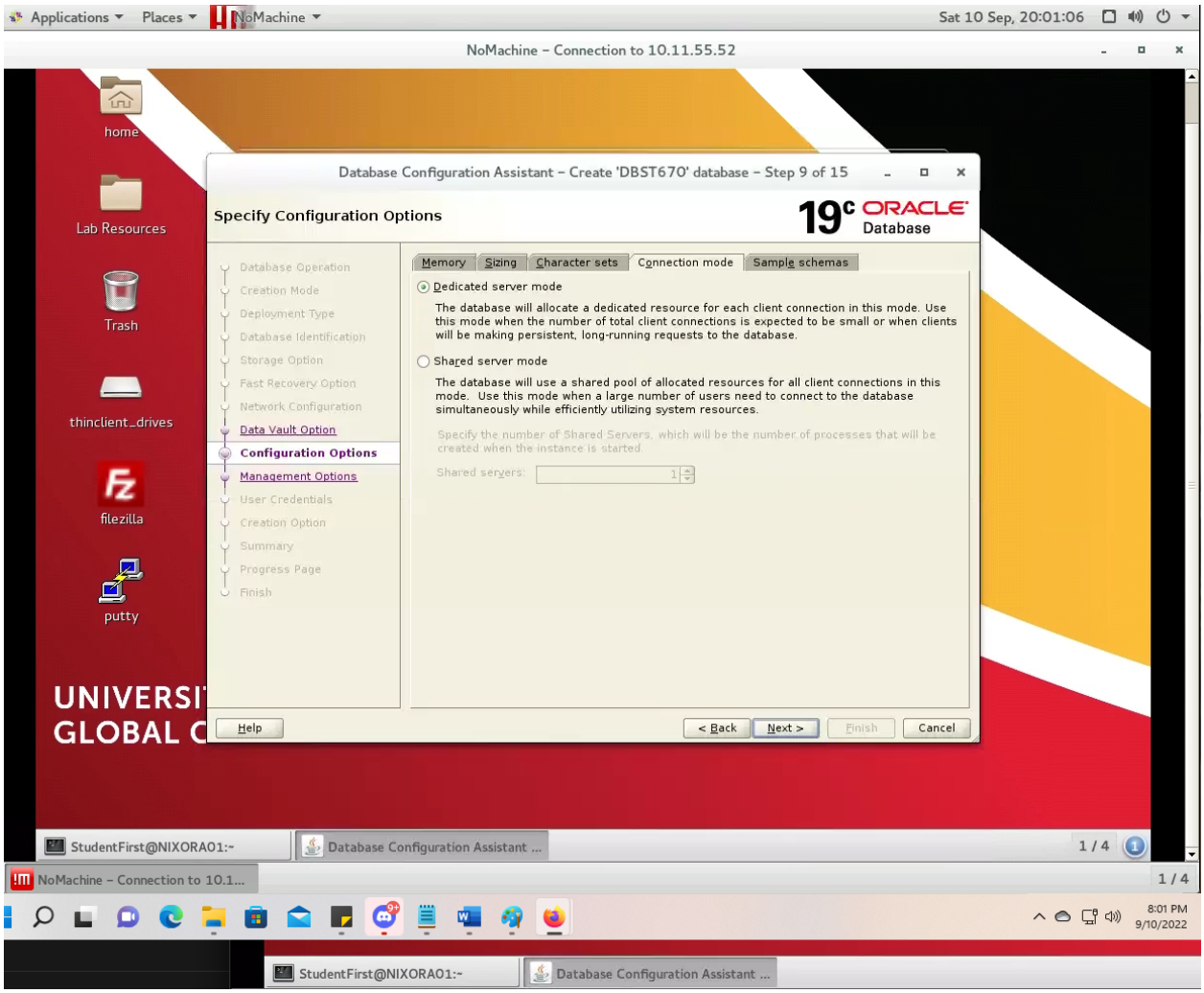

Applications
Places
NoMachine
Sat 10 Sep, 19:50:02
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 7 of 14
Specify Network Configuration Details
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Listener selection
Listeners from current Oracle home are listed below. Specify the listener name and port to create a new listener in current Oracle home.
Name
Port
Oracle home
Status
LISTENERV2
1521
/u01/app/oracle/product/19.0.0/db_1
Down
LISTENERV1
1521
/u01/app/oracle/product/19.0.0/db_1
Down
LISTENER
1521
/u01/app/oracle/product/19.0.0/db_1
Down
Create a new listener
Listener name:
LISTENERV3
Listener port:
1521
Oracle home:
/u01/app/oracle/product/19.0.0/db_1
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
1 / 4
NoMachine – Connection to 10.1...
1 / 4

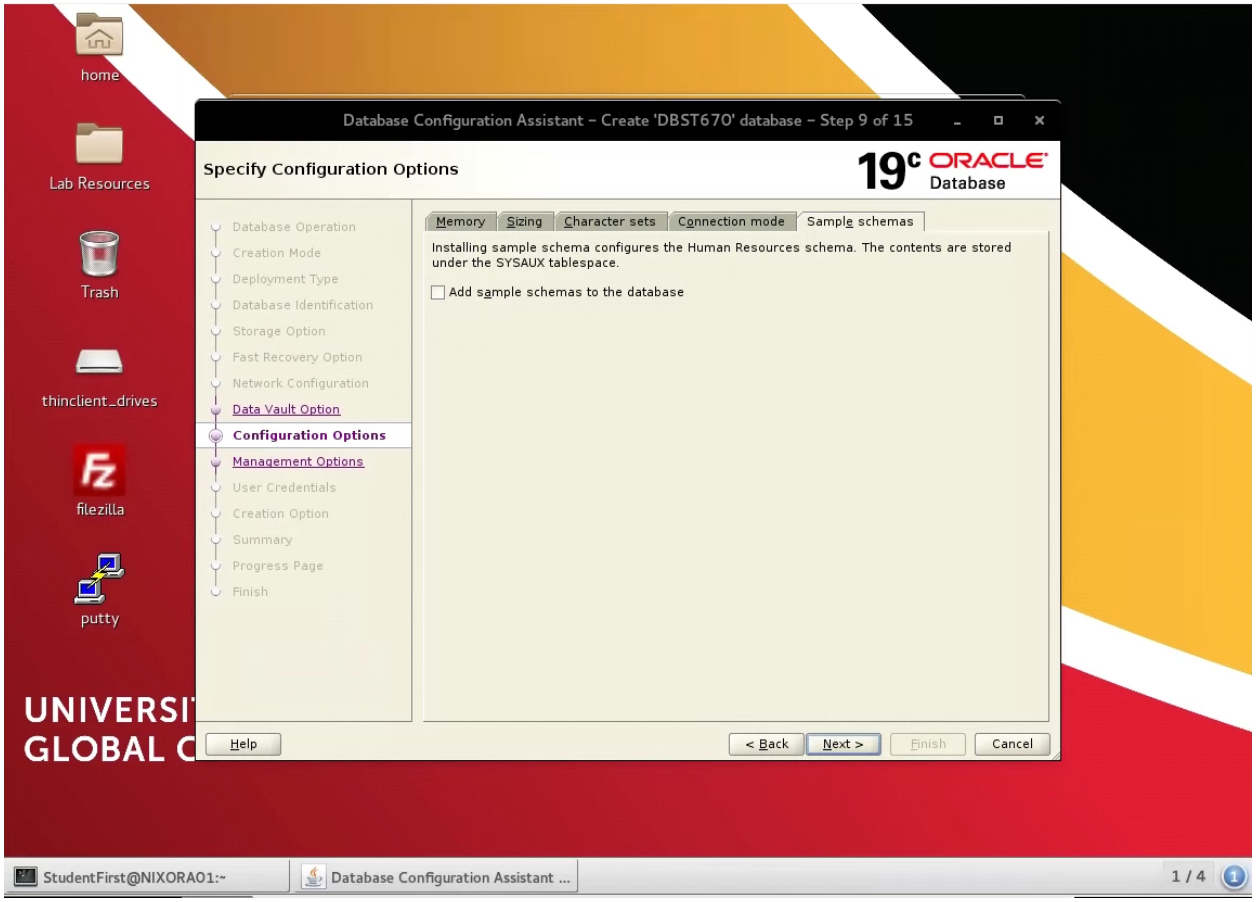

NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 8 of 15
Select Oracle Data Vault Config Option
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Configure Oracle Database Vault
Database Vault owner:
Password:
Confirm password:
Create a separate account manager
Account manager:
Password:
Confirm password:
Configure Oracle Label Security
Configure Oracle Label Security with OID
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
1 / 4
NoMachine – Connection to 10.1...
1 / 4

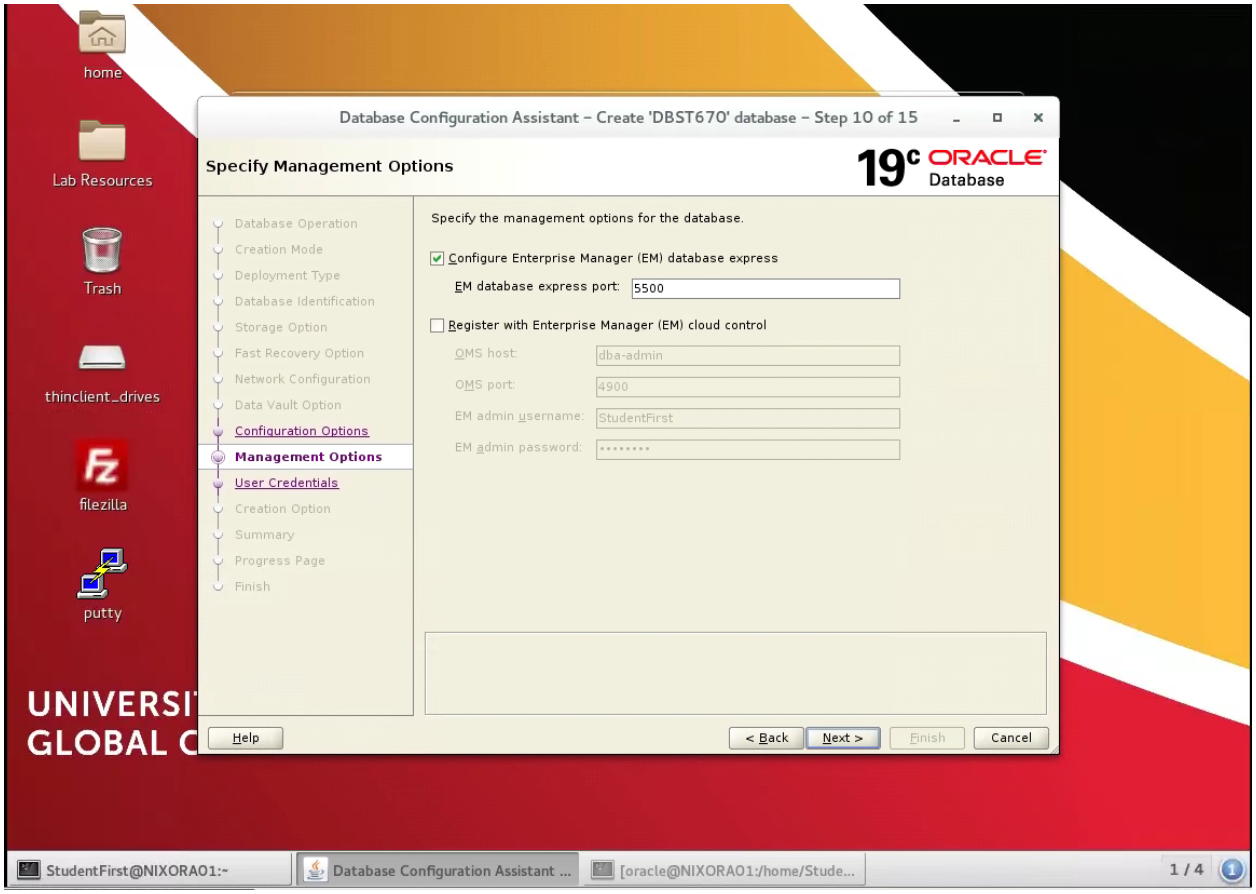

Applications
Places
NoMachine
Sat 10 Sep, 19:57:16
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 9 of 15
Specify Configuration Options
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Memory
Sizing
Character sets
Connection mode
Sample schemas
Use Automatic Shared Memory Management
SGA size: 3000 MB
PGA Size: 1000 MB
4000
32012
Use Manual Shared Memory Management
Shared pool size:
Buffer cache size:
Java pool size:
Large pool size:
PGA size:
Total memory for database 0 MB
Use Automatic Memory Management
Memory target: 6535 MB
20%
390
6535
32012
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
1 / 4
NoMachine – Connection to 10.1...
1 / 4


NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 9 of 15
Specify Configuration Options
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Memory
Sizing
Character sets
Connection mode
Sample schemas
A block is the smallest unit of storage for allocation and for I/O. It cannot be changed once the database is created.
Block size:
8192
BYTES
Specify the maximum number of operating system user processes that can be simultaneously connected to this database. The value of this parameter includes the user processes and the Oracle background processes.
Processes:
640
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
1 / 4
NoMachine – Connection to 10.1...
1 / 4

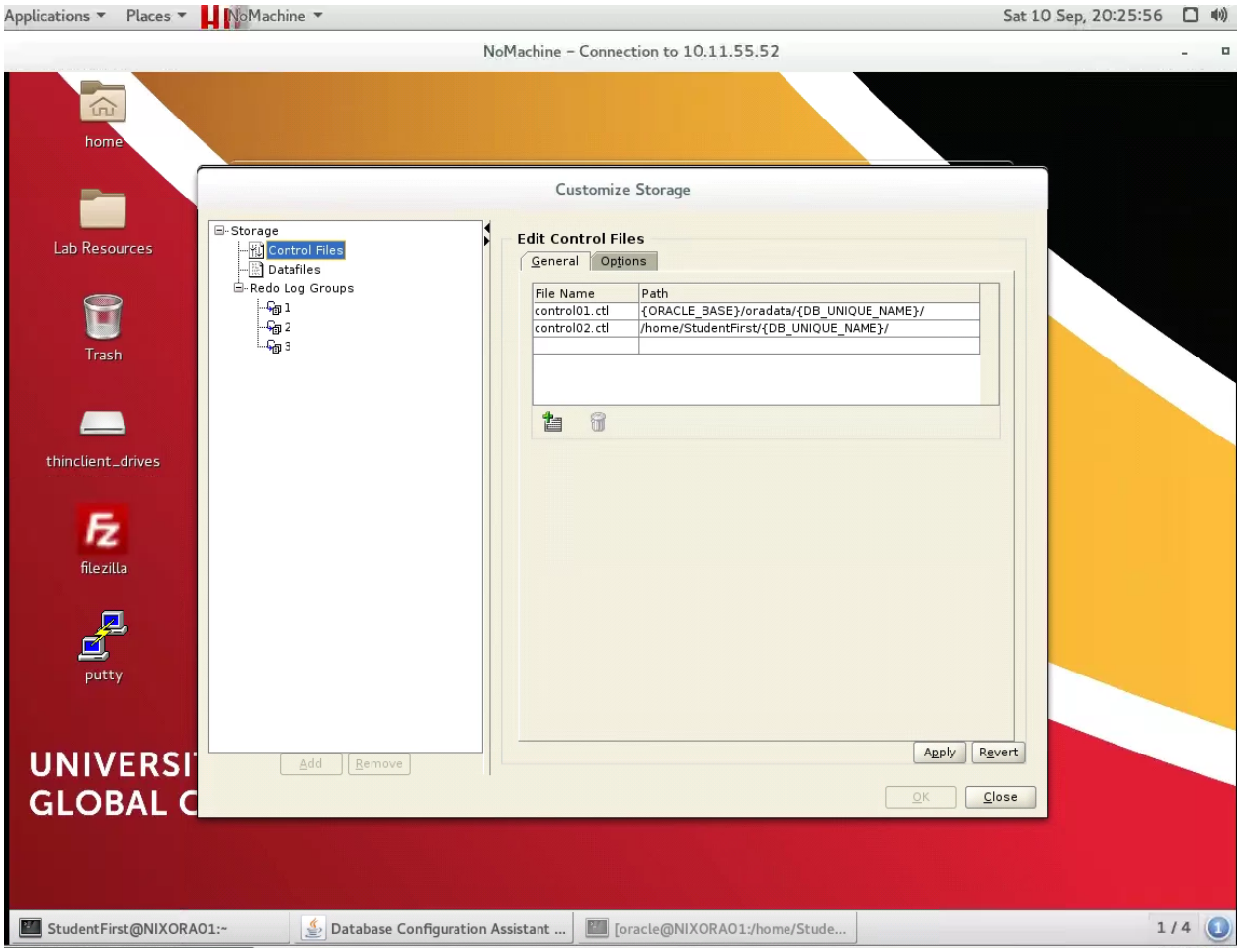

Applications
Places
NoMachine
Sat 10 Sep, 19:59:47
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 9 of 15
Specify Configuration Options
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Memory
Sizing
Character sets
Connection mode
Sample schemas
The database character set determines how character data is stored in the database.
Use Unicode (AL32UTF8)
Setting character set to Unicode (AL32UTF8) enables you to store multiple language groups.
Use OS character set (WE8MSWIN1252)
Character set is based on the language setting of this operating system.
Choose from the list of character sets
Database character set:
AL32UTF8 - Unicode UTF-8 Universal character set
Show recommended character sets only
National character set:
AL16UTF16 - Unicode UTF-16 Universal character set
Default language:
American
Default territory:
United States
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
1 / 4
NoMachine – Connection to 10.1...
1 / 4

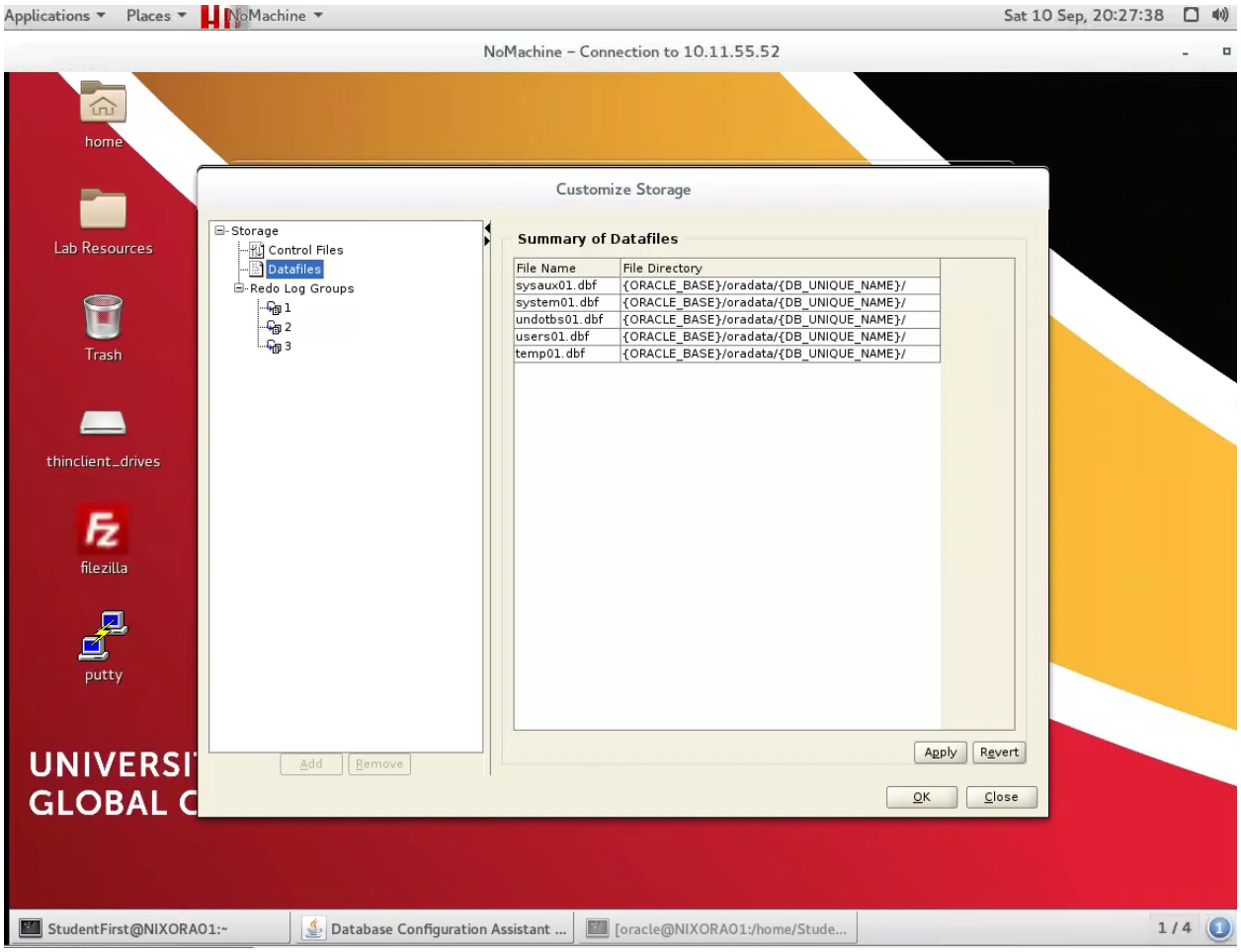

Applications
Places
NoMachine
Sat 10 Sep, 20:01:06
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 9 of 15
Specify Configuration Options
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Memory
Sizing
Character sets
Connection mode
Sample schemas
Dedicated server mode
The database will allocate a dedicated resource for each client connection in this mode. Use this mode when the number of total client connections is expected to be small or when clients will be making persistent, long-running requests to the database.
Shared server mode
The database will use a shared pool of allocated resources for all client connections in this mode. Use this mode when a large number of users need to connect to the database simultaneously while efficiently utilizing system resources.
Specify the number of Shared Servers, which will be the number of processes that will be created when the instance is started.
Shared servers:
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
1 / 4
NoMachine – Connection to 10.1...
8:01 PM
9/10/2022

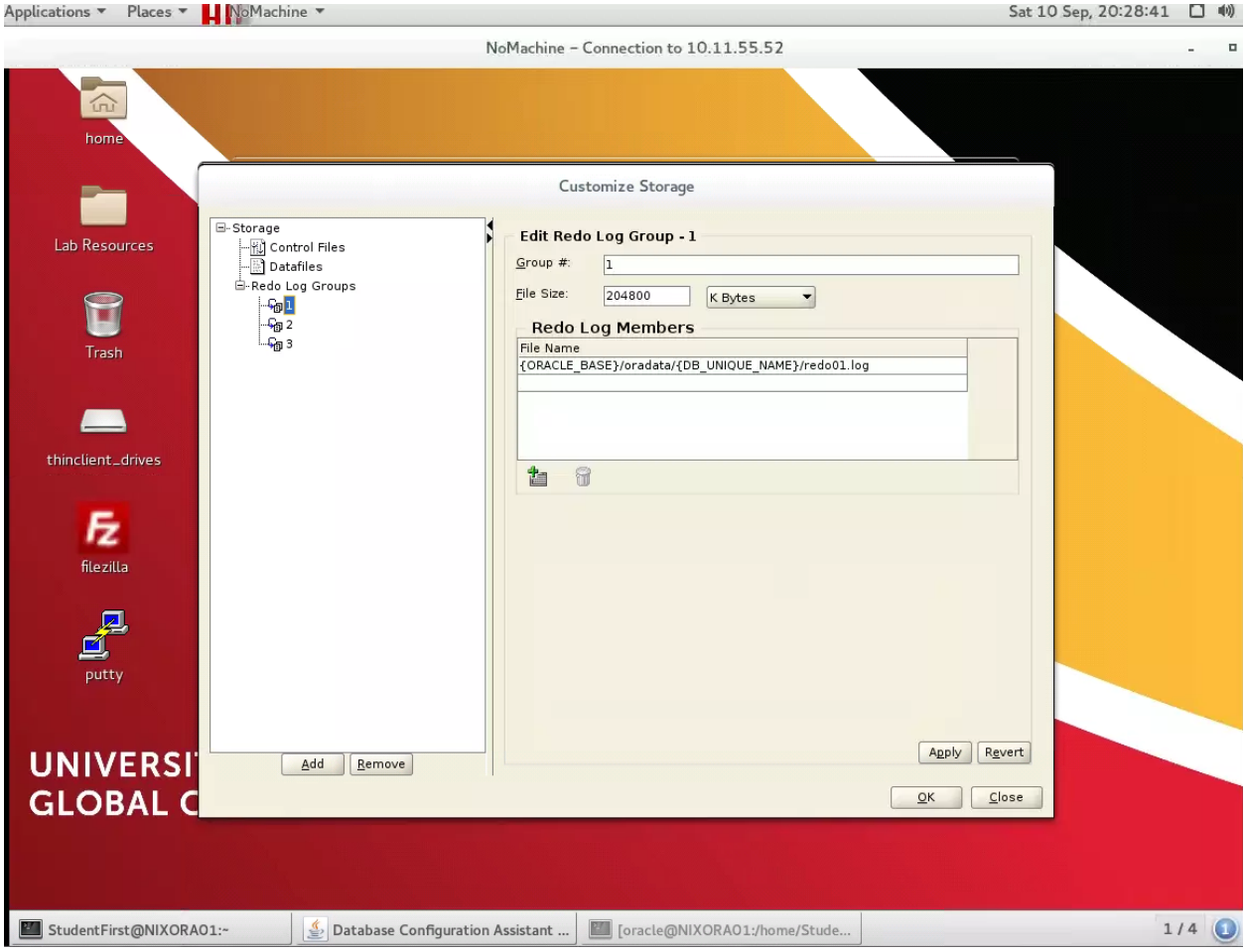

home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 9 of 15
Specify Configuration Options
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Memory
Sizing
Character sets
Connection mode
Sample schemas
Installing sample schema configures the Human Resources schema. The contents are stored under the SYSAUX tablespace.
Add sample schemas to the database
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
1 / 4

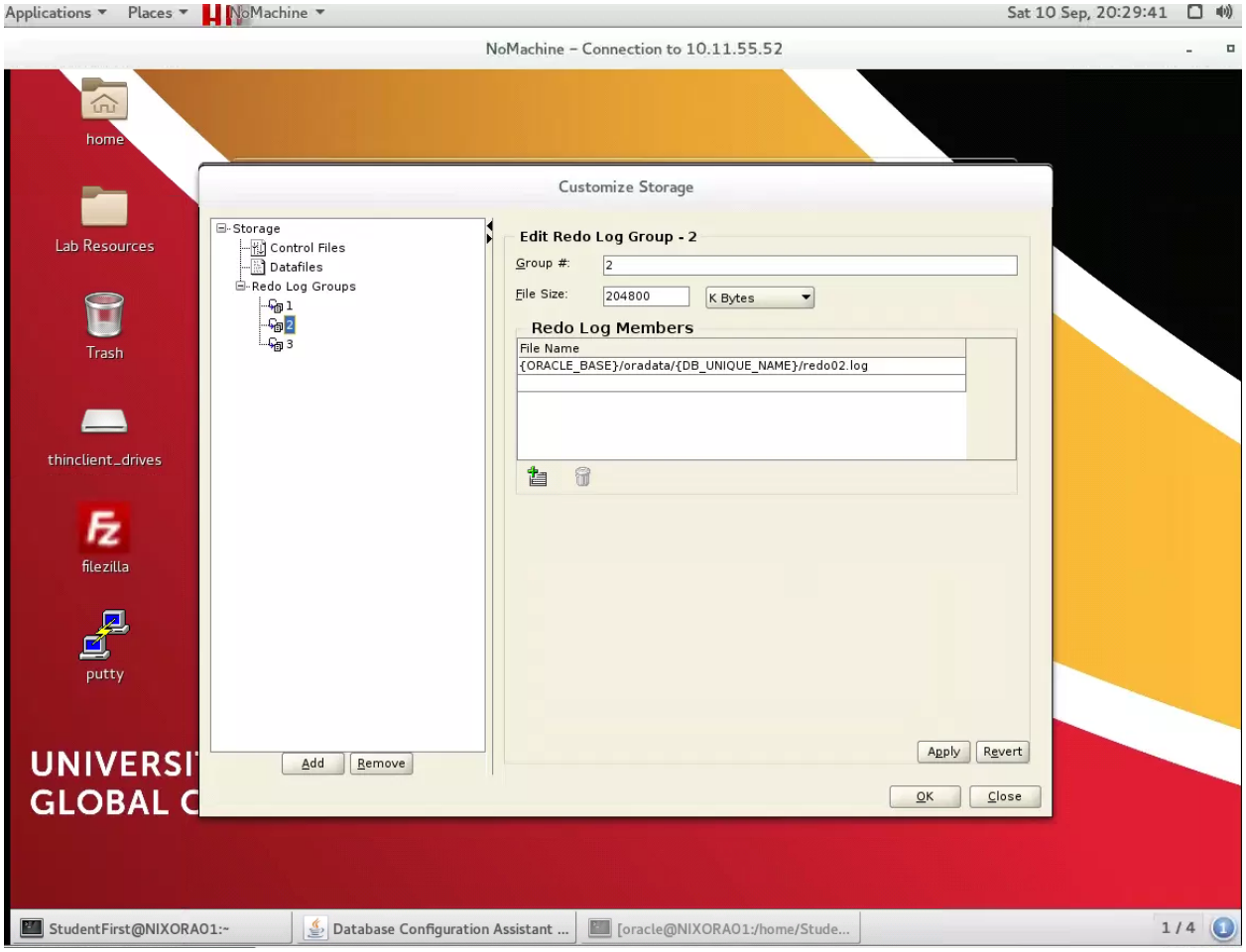

home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 10 of 15
Specify Management Options
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Specify the management options for the database.
Configure Enterprise Manager (EM) database express
EM database express port: 5500
Register with Enterprise Manager (EM) cloud control
OMS host: dba-admin
OMS port: 4900
EM admin username: StudentFirst
EM admin password:
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
[oracle@NIXORA01:/home/Stude...
1 / 4

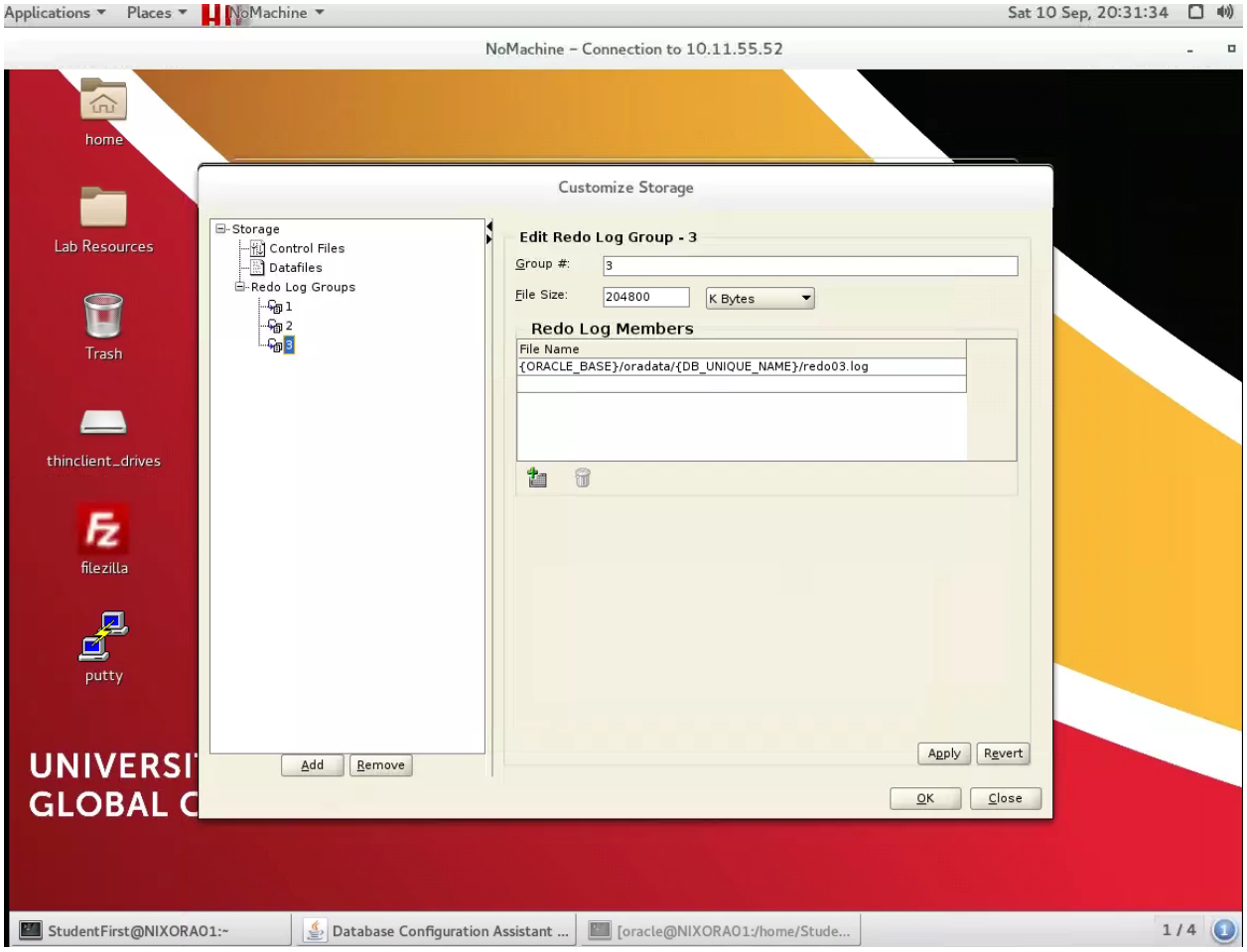

Applications
Places
NoMachine
Sat 10 Sep, 20:22:48
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 11 of 15
Specify Database User Credentials
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
You must specify passwords for the following user accounts in the new database for security reasons.
Use different administrative passwords
Password
Confirm password
SYS
SYSTEM
Use the same administrative password for all accounts
Password:
Confirm password:
Messages:
Password: [DBT-06208] The 'ADMIN' password entered does not conform to the Oracle recommended standards.
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
[oracle@NIXORA01:/home/Stude...
1 / 4

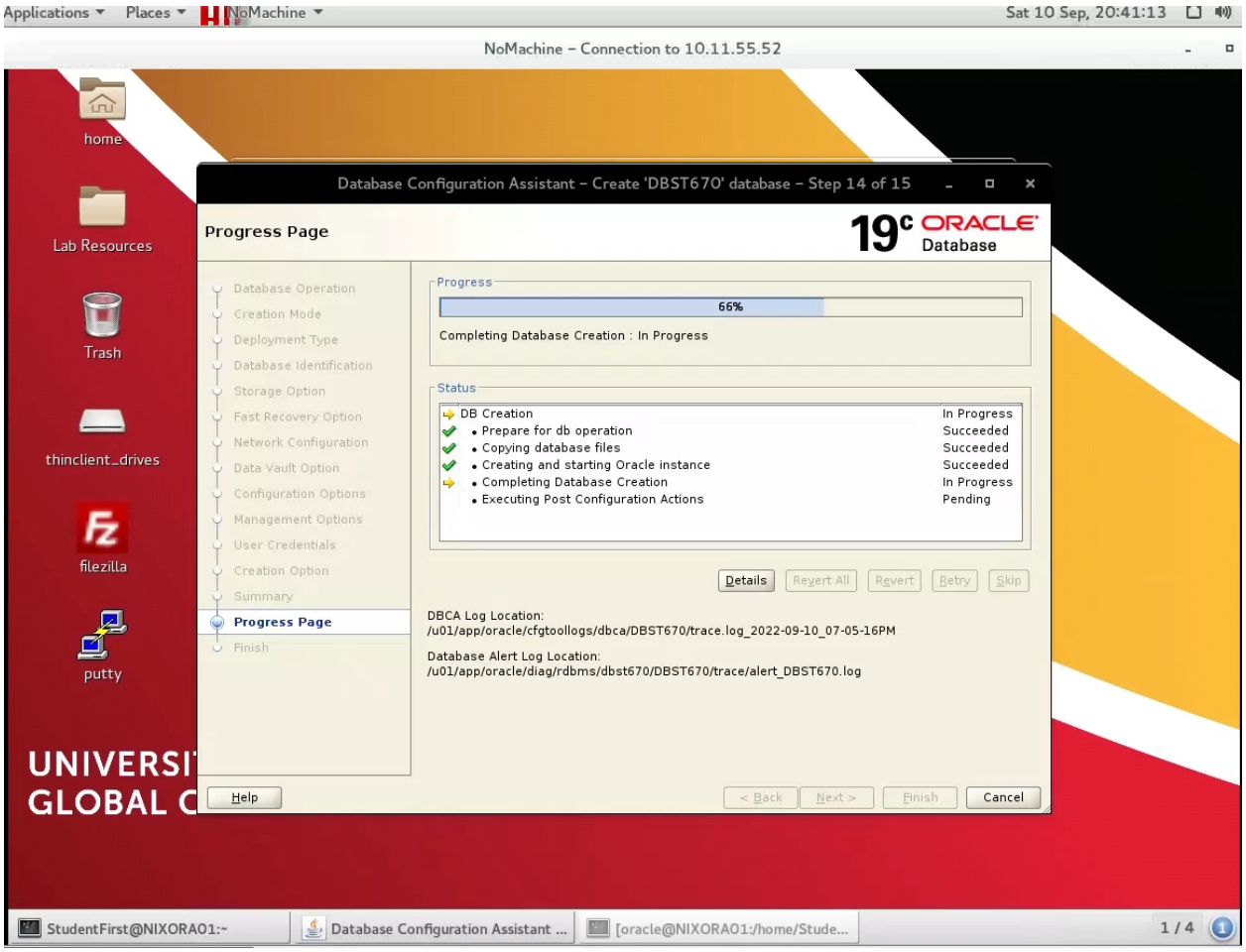

Applications
Places
NoMachine
Sat 10 Sep, 20:25:56
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Customize Storage
Storage
Control Files
Datafiles
Redo Log Groups
1
2
3
Edit Control Files
General
Options
File Name
Path
control01.ctl
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/
control02.ctl
/home/StudentFirst/{DB_UNIQUE_NAME}/
Add
Remove
Apply
Revert
OK
Close
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
[oracle@NIXORA01:/home/Stude...
1 / 4


Applications
Places
NoMachine
Sat 10 Sep, 20:27:38
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Customize Storage
Storage
Control Files
Datafiles
Redo Log Groups
1
2
3
Summary of Datafiles
File Name
File Directory
sysaux01.dbf
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/
system01.dbf
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/
undotbs01.dbf
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/
users01.dbf
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/
temp01.dbf
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/
Add
Remove
Apply
Revert
OK
Close
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
[oracle@NIXORA01:/home/Stude...
1 / 4

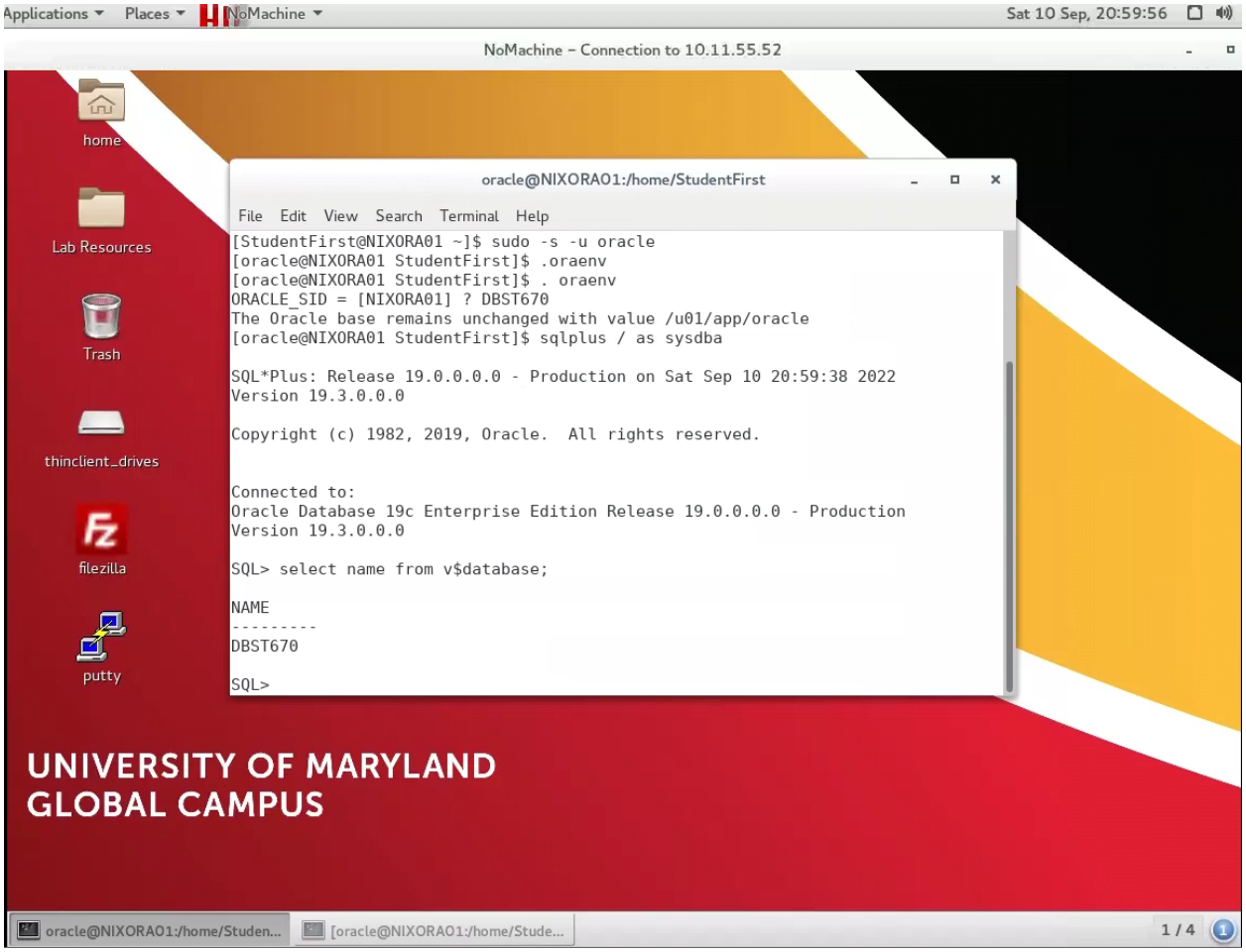

Applications
Places
NoMachine
Sat 10 Sep, 20:28:41
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Customize Storage
Storage
Control Files
Datafiles
Redo Log Groups
1
2
3
Edit Redo Log Group - 1
Group #:
1
File Size:
204800
K Bytes
Redo Log Members
File Name
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/redo01.log
Add
Remove
Apply
Revert
OK
Close
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
[oracle@NIXORA01:/home/Stude...
1 / 4


Applications
Places
NoMachine
Sat 10 Sep, 20:29:41
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Customize Storage
Storage
Control Files
Datafiles
Redo Log Groups
1
2
3
Edit Redo Log Group - 2
Group #:
2
File Size:
204800
K Bytes
Redo Log Members
File Name
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/redo02.log
Add
Remove
Apply
Revert
OK
Close
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
[oracle@NIXORA01:/home/Stude...
1 / 4

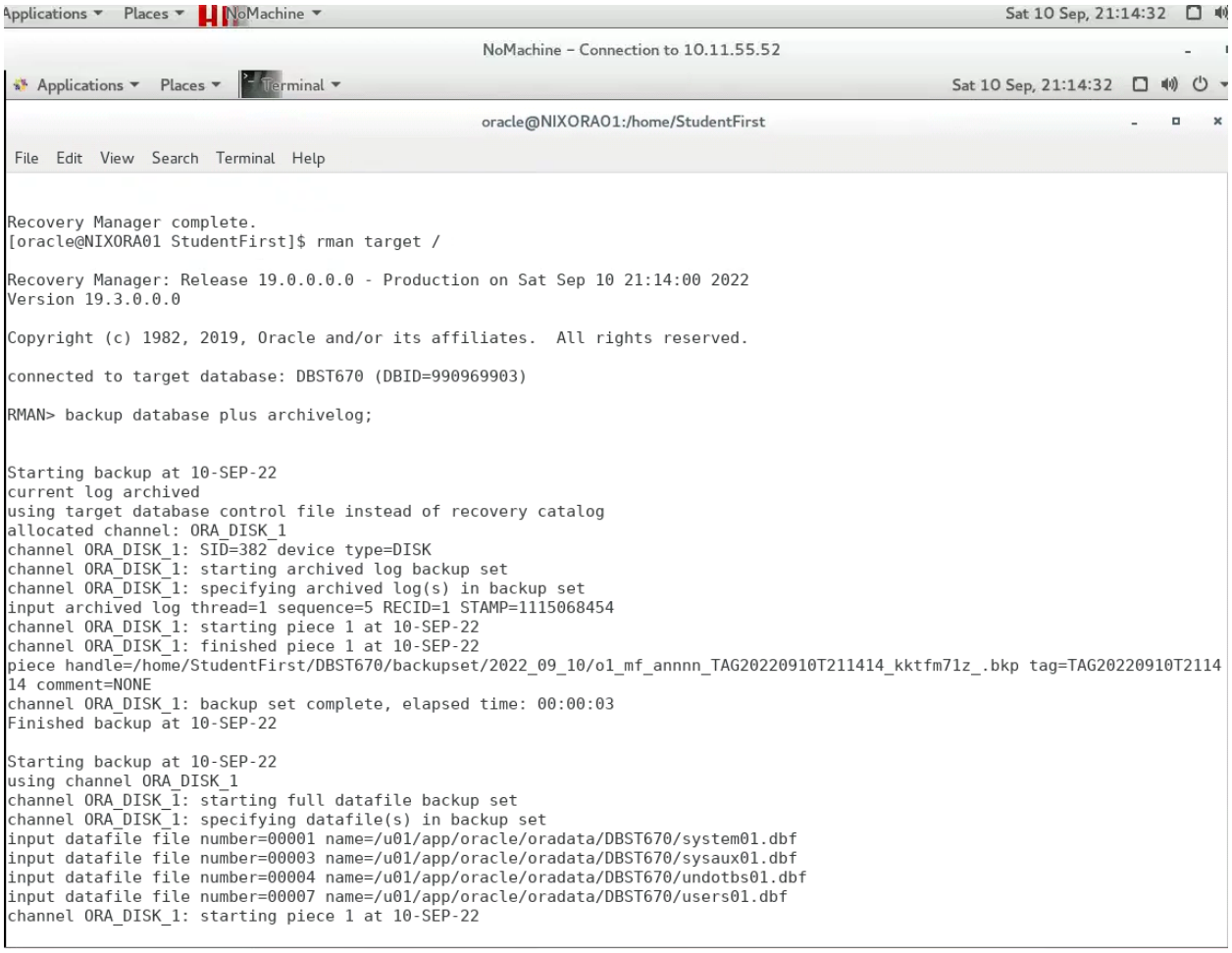

Applications
Places
NoMachine
Sat 10 Sep, 20:31:34
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Customize Storage
Storage
Control Files
Datafiles
Redo Log Groups
1
2
3
Edit Redo Log Group - 3
Group #:
3
File Size:
204800
K Bytes
Redo Log Members
File Name
{ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/redo03.log
Add
Remove
Apply
Revert
OK
Close
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
[oracle@NIXORA01:/home/Stude...
1 / 4

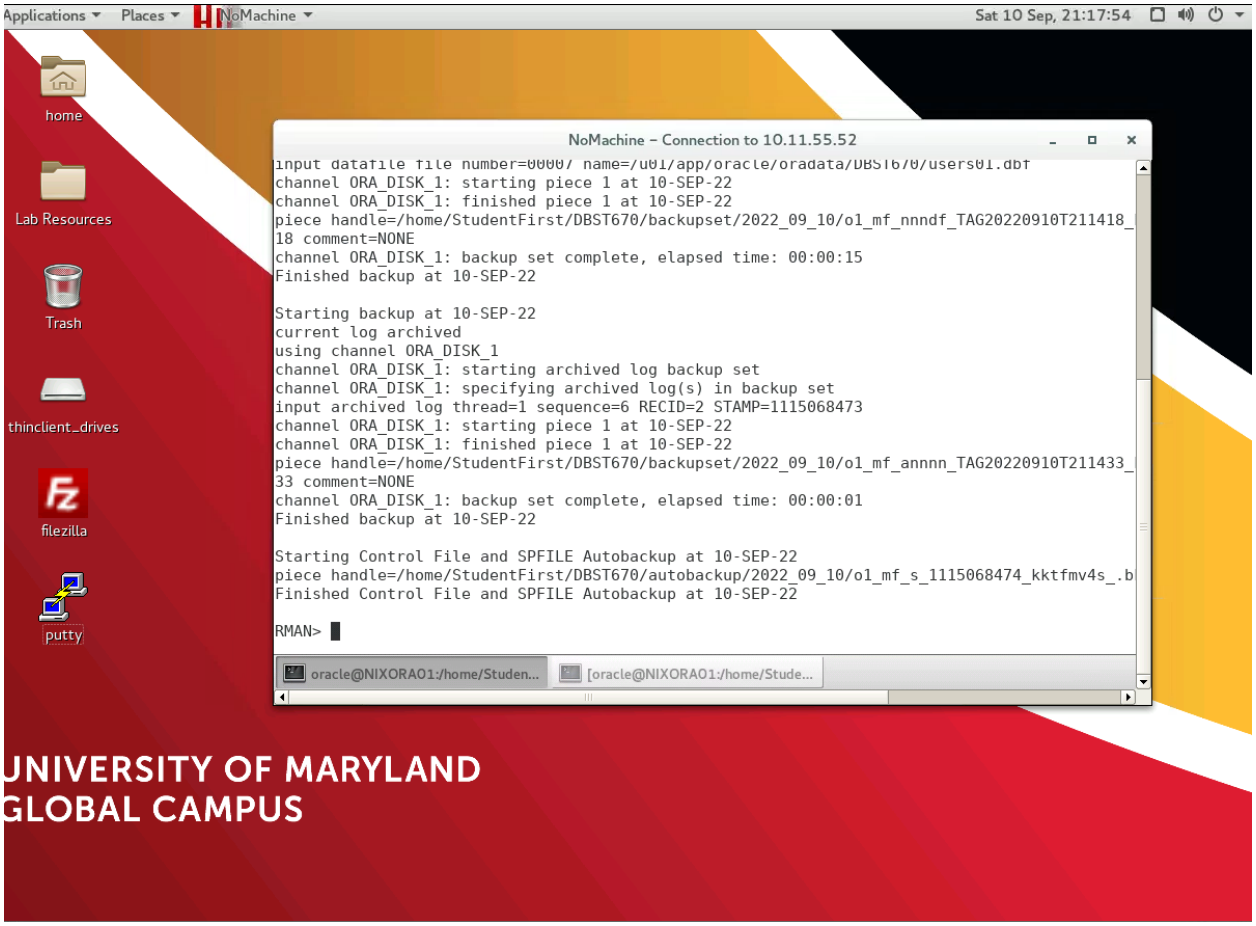

Applications
Places
NoMachine
Sat 10 Sep, 20:41:13
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 14 of 15
Progress Page
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Progress
66%
Completing Database Creation : In Progress
Status
DB Creation In Progress
Prepare for db operation Succeeded
Copying database files Succeeded
Creating and starting Oracle instance Succeeded
Completing Database Creation In Progress
Executing Post Configuration Actions Pending
Details
Revert All
Revert
Retry
Skip
DBCA Log Location:
/u01/app/oracle/cfgtoollogs/dbca/DBST670/trace.log_2022-09-10_07-05-16PM
Database Alert Log Location:
/u01/app/oracle/diag/rdbms/dbst670/DBST670/trace/alert_DBST670.log
Help
< Back
Next >
Finish
Cancel
StudentFirst@NIXORA01:~
Database Configuration Assistant ...
[oracle@NIXORA01:/home/Stude...
1 / 4

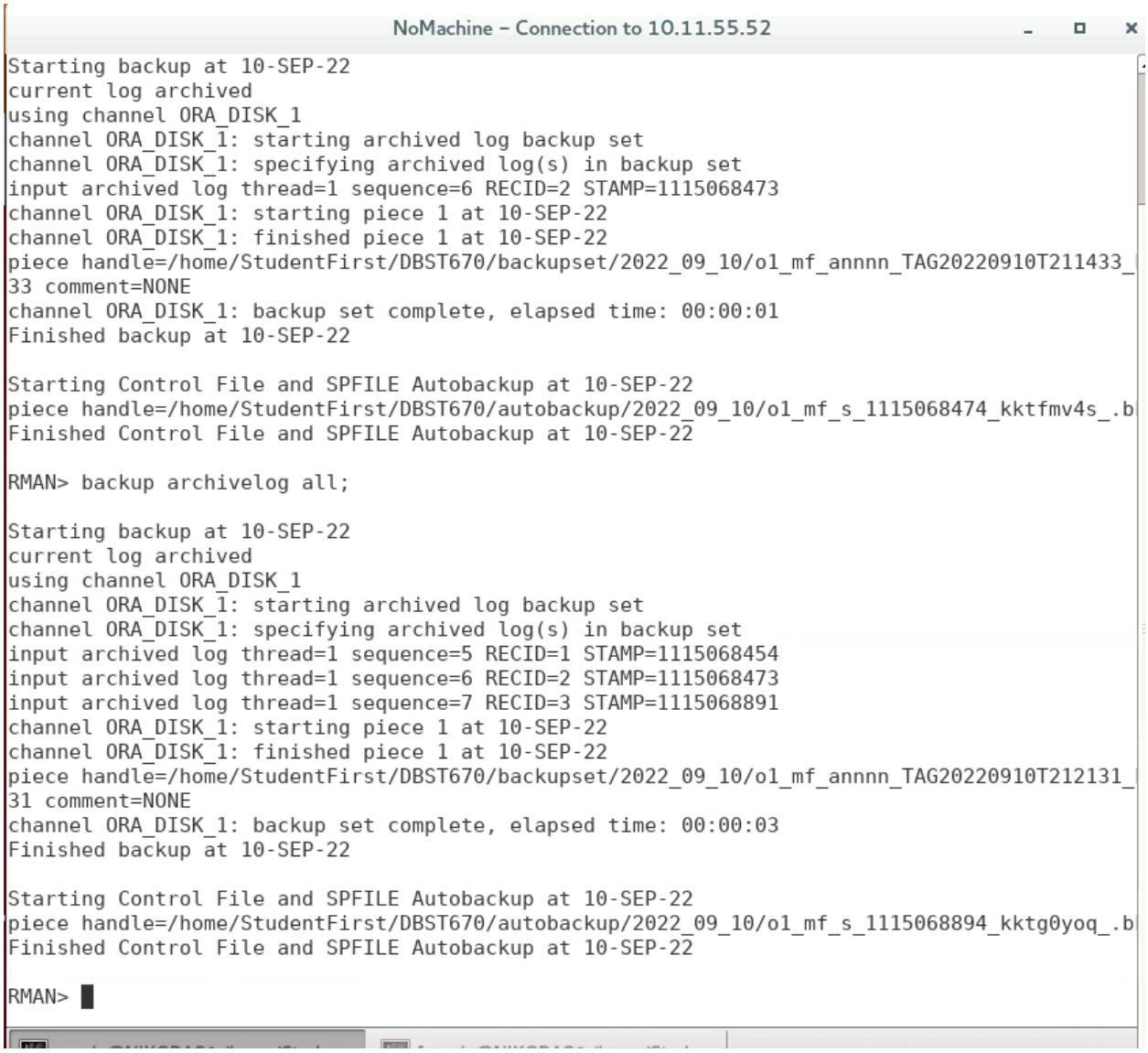

Applications
Places
NoMachine
Sat 10 Sep, 20:48:44
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
Database Configuration Assistant – Create 'DBST670' database – Step 15 of 15
Finish
19c ORACLE Database
Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Data Vault Option
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish
Database creation complete. For details check the logfiles at:
/u01/app/oracle/cfgtoollogs/dbca/DBST670.
Database Information:
Global Database Name: DBST670
System Identifier(SID): DBST670
Server Parameter File name: /u01/app/oracle/product/19.0.0/db_1/dbs/spfileDBST670.ora
EM Database Express URL: https://dbst670-nixora01.labs.daas.umuc.edu:5500/em
Note: All database accounts except SYS and SYSTEM are locked. Select the Password Management button to view a complete list of locked accounts or to manage the database accounts. From the Password Management window, unlock only the accounts you will use. Oracle strongly recommends changing the default passwords immediately after unlocking the account.
Password Management...
Help
< Back
Next >
Finish
Close

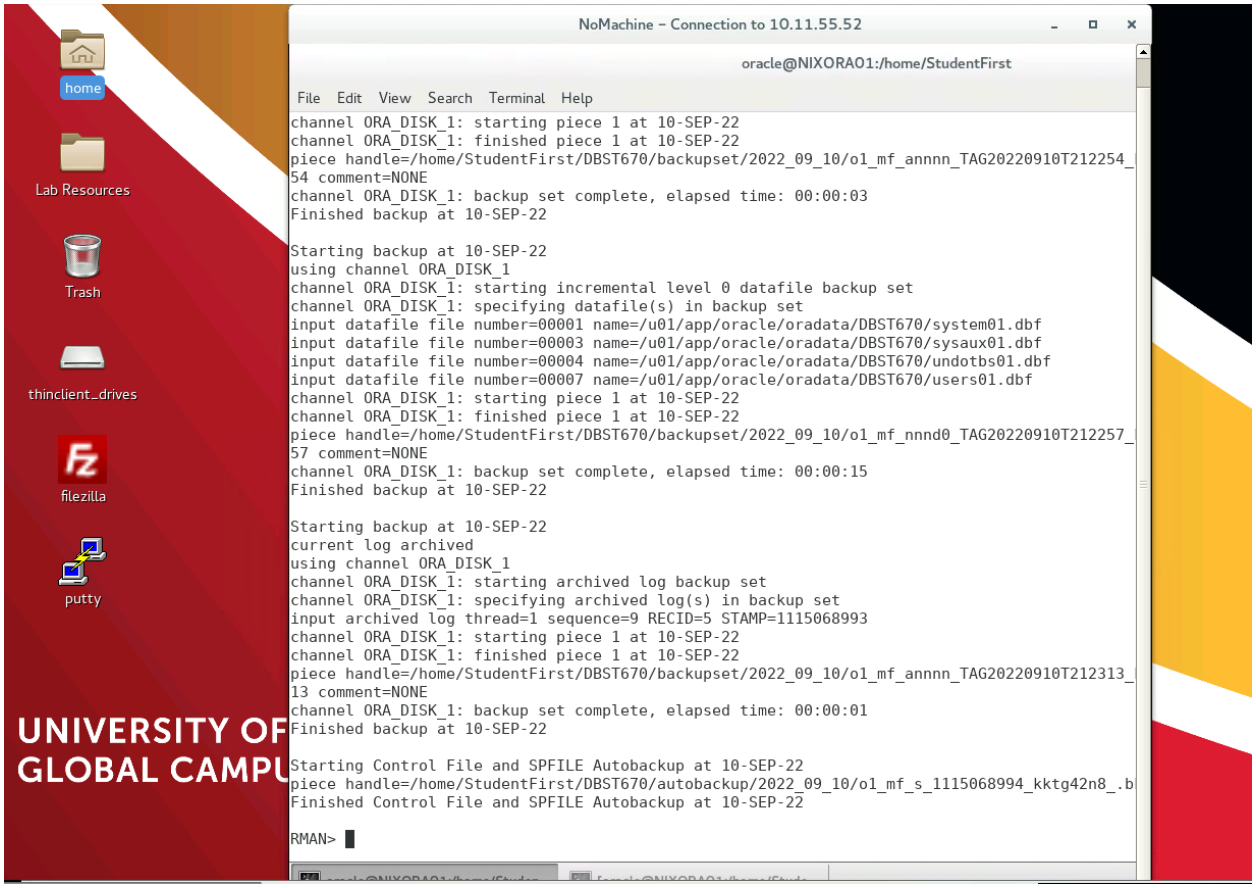

Applications
Places
NoMachine
Sat 10 Sep, 20:59:56
NoMachine – Connection to 10.11.55.52
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
oracle@NIXORA01:/home/StudentFirst
File Edit View Search Terminal Help
[StudentFirst@NIXORA01 ~]$ sudo -s -u oracle
[oracle@NIXORA01 StudentFirst]$ .oraenv
[oracle@NIXORA01 StudentFirst]$. oraenv
ORACLE_SID = [NIXORA01] ? DBST670
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@NIXORA01 StudentFirst]$ sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Sat Sep 10 20:59:38 2022
Version 19.3.0.0.0
Copyright (c) 1982, 2019, Oracle. All rights reserved.
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
SQL> select name from v$database;
NAME

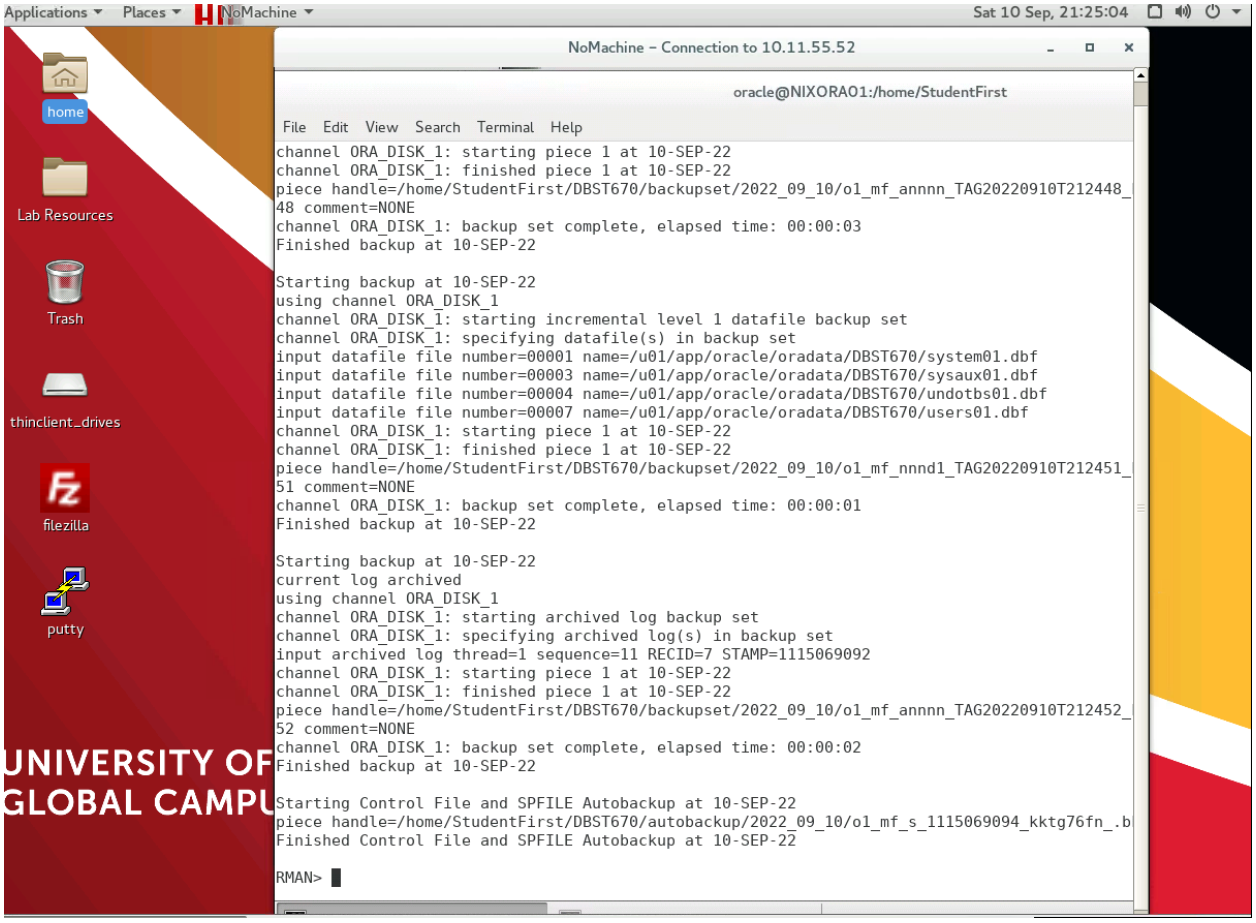
DBST670
SQL>
UNIVERSITY OF MARYLAND
GLOBAL CAMPUS
oracle@NIXORA01:/home/Studen...
[oracle@NIXORA01:/home/Stude...
1 / 4


Applications Places NoMachine Sat 10 Sep, 21:14:32
NoMachine – Connection to 10.11.55.52
Applications Places Terminal Sat 10 Sep, 21:14:32
oracle@NIXORA01:/home/StudentFirst
File Edit View Search Terminal Help
Recovery Manager complete.
[oracle@NIXORA01 StudentFirst]$ rman target /
Recovery Manager: Release 19.0.0.0.0 - Production on Sat Sep 10 21:14:00 2022
Version 19.3.0.0.0
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.
connected to target database: DBST670 (DBID=990969903)
RMAN> backup database plus archivelog;
Starting backup at 10-SEP-22
current log archived
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=382 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=5 RECID=1 STAMP=1115068454
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_annnn_TAG20220910T211414_kktfm71z_.bkp tag=TAG20220910T2114
14 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 10-SEP-22
Starting backup at 10-SEP-22
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=/u01/app/oracle/oradata/DBST670/system01.dbf
input datafile file number=00003 name=/u01/app/oracle/oradata/DBST670/sysaux01.dbf
input datafile file number=00004 name=/u01/app/oracle/oradata/DBST670/undotbs01.dbf
input datafile file number=00007 name=/u01/app/oracle/oradata/DBST670/users01.dbf
channel ORA_DISK_1: starting piece 1 at 10-SEP-22


Applications
Places
NoMachine
Sat 10 Sep, 21:17:54
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
NoMachine – Connection to 10.11.55.52
input datafile file number=00007 name=/u01/app/oracle/oradata/DBST670/users01.dbf
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_nnndf_TAG20220910T211418_
18 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 10-SEP-22
Starting backup at 10-SEP-22
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=6 RECID=2 STAMP=1115068473
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_annnn_TAG20220910T211433_
33 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 10-SEP-22
Starting Control File and SPFILE Autobackup at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/autobackup/2022_09_10/o1_mf_s_1115068474_kktfmv4s_.b
Finished Control File and SPFILE Autobackup at 10-SEP-22
RMAN>
oracle@NIXORAO1:/home/Studen...
[oracle@NIXORAO1:/home/Stude...
UNIVERSITY OF MARYLAND
GLOBAL CAMPUS


NoMachine – Connection to 10.11.55.52
Starting backup at 10-SEP-22
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=6 RECID=2 STAMP=1115068473
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_annnn_TAG20220910T211433_
33 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 10-SEP-22
Starting Control File and SPFILE Autobackup at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/autobackup/2022_09_10/o1_mf_s_1115068474_kktfmv4s_.b
Finished Control File and SPFILE Autobackup at 10-SEP-22
RMAN> backup archivelog all;
Starting backup at 10-SEP-22
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=5 RECID=1 STAMP=1115068454
input archived log thread=1 sequence=6 RECID=2 STAMP=1115068473
input archived log thread=1 sequence=7 RECID=3 STAMP=1115068891
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_annnn_TAG20220910T212131_
31 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 10-SEP-22
Starting Control File and SPFILE Autobackup at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/autobackup/2022_09_10/o1_mf_s_1115068894_kktg0yoq_.b
Finished Control File and SPFILE Autobackup at 10-SEP-22
RMAN>


home
Lab Resources
Trash
thinclient_drives
filezilla
putty
UNIVERSITY OF
GLOBAL CAMPU
NoMachine – Connection to 10.11.55.52
oracle@NIXORAO1:/home/StudentFirst
File Edit View Search Terminal Help
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_annnn_TAG20220910T212254_
54 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 10-SEP-22
Starting backup at 10-SEP-22
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=/u01/app/oracle/oradata/DBST670/system01.dbf
input datafile file number=00003 name=/u01/app/oracle/oradata/DBST670/sysaux01.dbf
input datafile file number=00004 name=/u01/app/oracle/oradata/DBST670/undotbs01.dbf
input datafile file number=00007 name=/u01/app/oracle/oradata/DBST670/users01.dbf
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_nnnd0_TAG20220910T212257_
57 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 10-SEP-22
Starting backup at 10-SEP-22
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=9 RECID=5 STAMP=1115068993
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_annnn_TAG20220910T212313_
13 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 10-SEP-22
Starting Control File and SPFILE Autobackup at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/autobackup/2022_09_10/o1_mf_s_1115068994_kktg42n8_.b
Finished Control File and SPFILE Autobackup at 10-SEP-22
RMAN>


Applications
Places
NoMachine
Sat 10 Sep, 21:25:04
home
Lab Resources
Trash
thinclient_drives
filezilla
putty
UNIVERSITY OF
GLOBAL CAMPU
NoMachine – Connection to 10.11.55.52
oracle@NIXORA01:/home/StudentFirst
File Edit View Search Terminal Help
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_annnn_TAG20220910T212448_
48 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 10-SEP-22
Starting backup at 10-SEP-22
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=/u01/app/oracle/oradata/DBST670/system01.dbf
input datafile file number=00003 name=/u01/app/oracle/oradata/DBST670/sysaux01.dbf
input datafile file number=00004 name=/u01/app/oracle/oradata/DBST670/undotbs01.dbf
input datafile file number=00007 name=/u01/app/oracle/oradata/DBST670/users01.dbf
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_nnnd1_TAG20220910T212451_
51 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 10-SEP-22
Starting backup at 10-SEP-22
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=11 RECID=7 STAMP=1115069092
channel ORA_DISK_1: starting piece 1 at 10-SEP-22
channel ORA_DISK_1: finished piece 1 at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/backupset/2022_09_10/o1_mf_annnn_TAG20220910T212452_
52 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 10-SEP-22
Starting Control File and SPFILE Autobackup at 10-SEP-22
piece handle=/home/StudentFirst/DBST670/autobackup/2022_09_10/o1_mf_s_1115069094_kktg76fn_.b
Finished Control File and SPFILE Autobackup at 10-SEP-22
RMAN>