



UNIVERSITY *of*
GREENWICH

Course Title: Advanced Database Technologies

Student Name: Mohamed AL-Kaisi

Student ID: 000931504

COMP1802-004

<https://classroom-vc.gre.ac.uk/>

Contents

Introduction	5
<i>DBA Requirements</i>	6
1) Create new permanent tablespace inv_tsp.....	6
2) Create several new users with the following users with a login requirement	8
3) All the users should have the following access rights.....	14
4) Create a table inv _ categories inside the newly created schema for the inventory department with the structure described in the table below. The tablespace for this table should be inv _ tsp. 20	
5) The sh user must be able to run a report with the information about all the product sold for a customer, based on the provided id.....	22
6) If you accidentally dropped an important schema (for example, hr) in your database, discuss the best functionality to use to recover it quickly and safely. Provide an explanation of the steps needed to recover data including the SQL commands for each step.....	29
7) Your users are reporting “ORA-1555: Snapshot too old” errors. What might be the cause of this? Explain what you need to do in order to solve this issue.	32
8) Discuss how different database failures can impact a business, including the financial costs, and what can be done to minimize the risk. Please include into discussion the actions that should be taken to prevent future problems	32
9) Evaluation	34
10) Completed requirement	36
11) Strength and weakness of the implementation	36
12) Conclusion.....	36
14) Bibliography	37

Figures Table

Figure 1: Creating tablespace using SQL plus terminal.....	7
Figure 2:Creating tablespace using SQL developer.....	7
Figure 3: Showing tablespace created after using SQL plus command	7
Figure 4: Creating user profile	10
Figure 5:Creating profile using SQL DEVELOPER.....	10
Figure 6: Editing Profile to prevent old password from being used.	11
Figure 7: Showing staff profile created using SQL DEVELOPER. The Screenshot shows 15-minute idle time set.	11
Figure 8: Screenshot showing the rest of the requirement implemented	11
Figure 9: The screenshot showing concurrent session has been set.....	11
Figure 10: Function that prevent user of having password that characters less than 8.	12
Figure 11: Testing verify function password.....	12
Figure 12: User One for customer service department	12
Figure 13: Second user for customer service department called Dept_Cust_2	13
Figure 14: The one user for inventory department	13
Figure 15: Prove that both tablespace attached to dept_ invent user.....	13
Figure 16: prove that both tablespaces attached to dept _ customer Service	13
Figure 17: Creating users using SQL Developer	14
Figure 18: user login failed.....	14
Figure 19: Two roles created. One for customers service users and one for inventory department user	15
Figure 20: Giving roles session privileges	15
Figure 21:Granting Dept_ customer Service role to Dept_Cust_1 user	15
Figure 22: Granting Dept_ Customer Service to Dept_Cust_2	15
Figure 23: Granting Dept _ Inventory to Dept_ invent user.	16
Figure 24:login passed for Dept_Cust_2.....	16
Figure 25. login passed for Dept_Cust_1 passed.....	16
Figure 26: Login passed for Dept_ invent	16
Figure 27: Creating roles using SQL developer	17
Figure 28: Granting roles to users.....	17
Figure 29: Roles assigned to user Trevor	18
Figure 30: Grant read only access to customer service users.....	18
Figure 31: Granting read and write access for customer service users on customer and sale table ...	19
Figure 32: inventory dept user. Read and write access for table sh. cost	19
Figure 33: inventory dept user. Read and write access for table sh. cost	19
Figure 34: Read access for dept_ inventory user on customers and sales.....	19
Figure 35: Assigning create view and create table privilege to inventory user	20
Figure 36: Created table inv _ categories for dept _ invent	20
Figure 37: Table created show inside inventory department schema	20
Figure 38: Grant read and write access for inventory department on inv _ categories to dept_ inventory role.....	21
Figure 39: alter user quotas on inv _ tsp tablespace	21
Figure 40:Insert to inv _ categories	22
Figure 41: Checking row inserted	22
Figure 42: View created for joining three tables	23

Figure 43: Small query to generate required report from the view for each customer depending to their id.....	23
Figure 44:Setting out PDF report	24
Figure 45: PDF report exported	24
Figure 46:Report builder.....	25
Figure 47: Setting SQL query inside report wizard	25
Figure 48: Queries used to generate report on report wizard	26
Figure 49: Interactive report.....	26
Figure 50: Using find tool to find data	27
Figure 51: Generating report using HTML view	27

Introduction

The Quick Sale company sells several products, such as computer hardware and software, music, clothing, and tools. The company maintains information about these products, such as product identification numbers, the category into which the product falls, orders information, the supplier, the availability status of the product, a list price, etc. Inventory information is also recorded for all products, including the warehouse where the product is available and the quantity on hand. Customer information is also tracked.

The company hired me as DBA and my first task was assigning different access rights to their employees in different departments based on their job description. Access right in this situation related to the type of information that each employee can access. This would ensure that the company follows the data protection act.

DBA Requirements

1) Create new permanent tablespace inv_tsp

The extracted specification of the tablespace

- Permanent tablespace named inv_tsp
- The size to store extra information 70MB.
- Tablespace should be management should be local.
- The size of tablespace should be automatically expandable by 10MB to a maximum size of 120 MB.

Answer:

Oracle divides its storage into logical and physical storage. The tablespace is a logical storage unit and it's divided into logical units of storage such as segments and its divided again into extents. Oracle provides different type of tablespace such as undo, temporary, sysaux , system, permanent tablespaces. Physical storage defined as data files that store tablespace objects inside it. Tablespaces have one or more data files.

I decided to created two data file by adding an extra datafile to be used for data recovery. This ensures that if one lost or crashed then the second can be used as the backup because it's stored in a different location. The requirement was given that the default size of a tablespace be 70MB then to be extended 10MB when its full and the maximum size should be 120MB that's when the last size for the data file to be extended.

The clause to enable the size of the data file to be automatically extendable is using "Auto-extend" then specifying from the requirement the size to be extendable and the limit.

Commands → Auto-extend on next 10m max size 120m

The second requirement was to set the tablespace to be managed locally and this comes with several benefits. The first benefit becoming fast and concurrent space operation as the space allocation and deallocation modify locally managed resources. Having default clause extent management local means its set to be auto allocate which means enabling the database to automatically selects the appropriate extent size. User relying on the data dictionary for this situation is reduced because of information stored in file headers and bitmap blocks.

The last requirement to set up was the segment space management on whether it should be managed automatically or set up as manual. With setting up the clause for the tablespace to be managed locally, there are two phrases included to be used to manage segment spaces such as automatic and manual. The manual uses linked lists called free lists that manage free space in the segment. The automatic management of the segments that use bitmaps, which are more efficient methods to use and is the default for all the new permeant and locally managed tablespaces. The reason is better as it delivers better space utilization than the manual and its also self-tuning, for example, the increasing number for users or instances. Even though its set to be the default for management locally tablespace or with permanent tablespace must use the clause to set it enabled using this command "SEGMENT SPACE MANAGEMENT LOCAL". (Docs.oracle.com, 2019)

The final version showing the executed code for tablespace.

```
SQL> ed
Wrote file afiedt.buf

 1  CREATE TABLESPACE inv_tsp
 2  DATAFILE '/u01/app/oracle/oradata/inv_tsp01.dbf' size 70M,
 3  '/u01/app/oracle/fast_recovery_area/inv_tsp02.dbf' size 70M
 4  AutoEXTEND ON
 5  NEXT 10M
 6  MAXSIZE 120M
 7  EXTENT MANAGEMENT LOCAL
 8* SEGMENT SPACE MANAGEMENT AUTO
SQL> /
```

Tablespace created.

Figure 1: Creating tablespace using SQL plus terminal

What another way tablespace can be created?

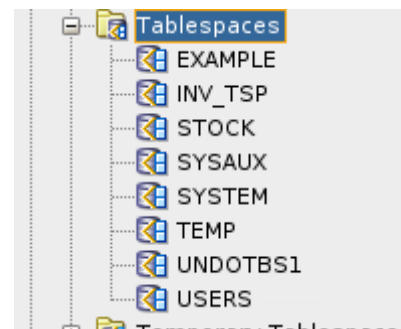


Figure 3: Showing tablespace created after using SQL plus command

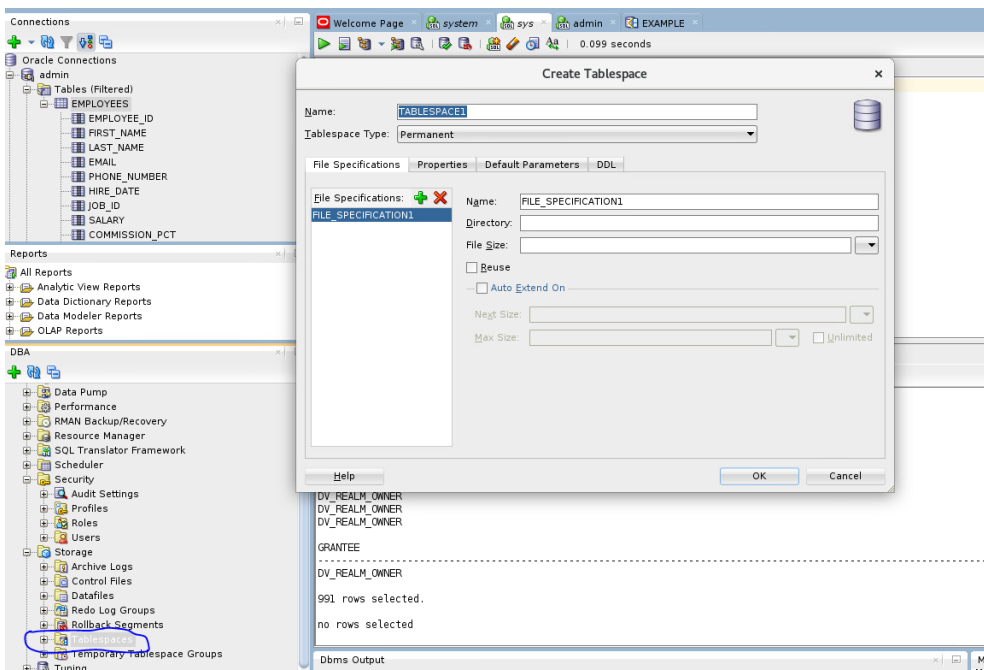


Figure 2: Creating tablespace using SQL developer

2) Create several new users with the following users with a login requirement

The extracted specification from the requirements are

- Create two users for the customer service departments.
- Create one user for the inventory departments.
- The default tablespace for all users the one above we created called inv_tsp.
- All the users above should have unique usernames and passwords.
- The password should be at least 8 characters long and the user should not have the password is shorter than 8 characters.
- The users should be forced to change their password every 6 months and not allowed to reuse the same password again.
- If during the login they specified password 3 times wrong, then the account should be locked for 30 hours.
- All the new users should be allowed to leave their sessions without interacting with it for no more than 15 minutes.

Answer:

Oracle database has a feature where you can create a profile with a set of limitations on the database resources and the user password. Once you assign users to the created profile then it cannot exceed the limits was set on the profile. This makes it easier to add restrictions to the number of users and fast set up that happens quickly because you have not set it up for each user as it can be all done at once by just assigning once the profile to a user or edit the profile to adjust more restriction.

- Password_lock_Time :

This clause specifies the number of days that an account will be locked after the specified number of following failed login attempts.

- Failed_login_attempts

This clause set up how many time a user can fail to login. Both failed login attempts clause and password_lock_time works together to fit the requirement that the user should be allowed to fail login no more than 3 times and if it does then lock the account for 30 hours hour which is 1.25 days.

- Password_life_time

This clause helps to specify the number of days for the same password to be used for authentication. The decision to set this clause for the requirement and with password grace time so that the password can expire if it's not changed within the grace period.

- Password_grace_time

This clause specifies the number of days after the grace period where the system starts giving warning issues to users and login allowed.

- Password_verify_function verify_password:

This clause used to connect the profile to a PLSQL function that will be used to set the requirement that the password should be no less than 8 characters. This requirement was specifically mentioned there should be no other enforcement than password characters should be no less than 8 characters. The provided function ora12c_verify_function has more restriction than just the 8 characters such as checking the password if contain oracle or checking if the username reversed and so on. The requirement suggests not to have any other enforcement. The code in that function ensuring the character length for the password will be copied into a newer function and set up to a profile.

- IDLE_TIME

The clause above specifies the permitted period of the time that inactive user session stays active and that can be expressed in a minute and after the time set finished then the session dies. This was done if the clause set when its included in the profile. The reason we are using the clause as the requirement suggested that users should be allowed to leave their session without interaction for at least 15 minutes.

- SESSIONS_PER_USER

The clause above specifies the number of concurrent sessions that each user can have. After creating the profile and looking back at the requirement, I identified a missing requirement that an old password should never use. According to oracle documentation, both clauses below should be used to disable the old password from being used again.

- Password Reuse Time and Password Reuse Max:

The requirement suggests that the change of the password should happen, and the old password should never be used. Both of those clauses used together to set to prevent the old password from being used. Oracle documentation suggested that if you set one of them to be unlimited then the old password can never be used. This functionality will be implemented as it has been requested as part of the requirement. If you specify an integer for either of these parameters and specify UNLIMITED for the other, then the user can never reuse a password.

```

1 CREATE PROFILE staff_profile LIMIT
2 PASSWORD_LOCK_TIME 1.25
3 FAILED_LOGIN_ATTEMPTS 3
4 PASSWORD_LIFE_TIME 175.5
5 PASSWORD_GRACE_TIME 7
6 PASSWORD_VERIFY_FUNCTION VERIFY_PASSWORD
7 IDLE_TIME 15
8* SESSIONS_PER_USER 1
SQL> /

```

Profile created.

Figure 4: Creating user profile

Figure 4 shows the creation of a profile and the use of all clauses explained before a page with the requirement that was given for their users.

The profile can also be created using the SQL developer software and login as DBA using sys with role dba. After login just navigates to the sys user with role dba then click on security and afterward right-click on Profile and click on new.

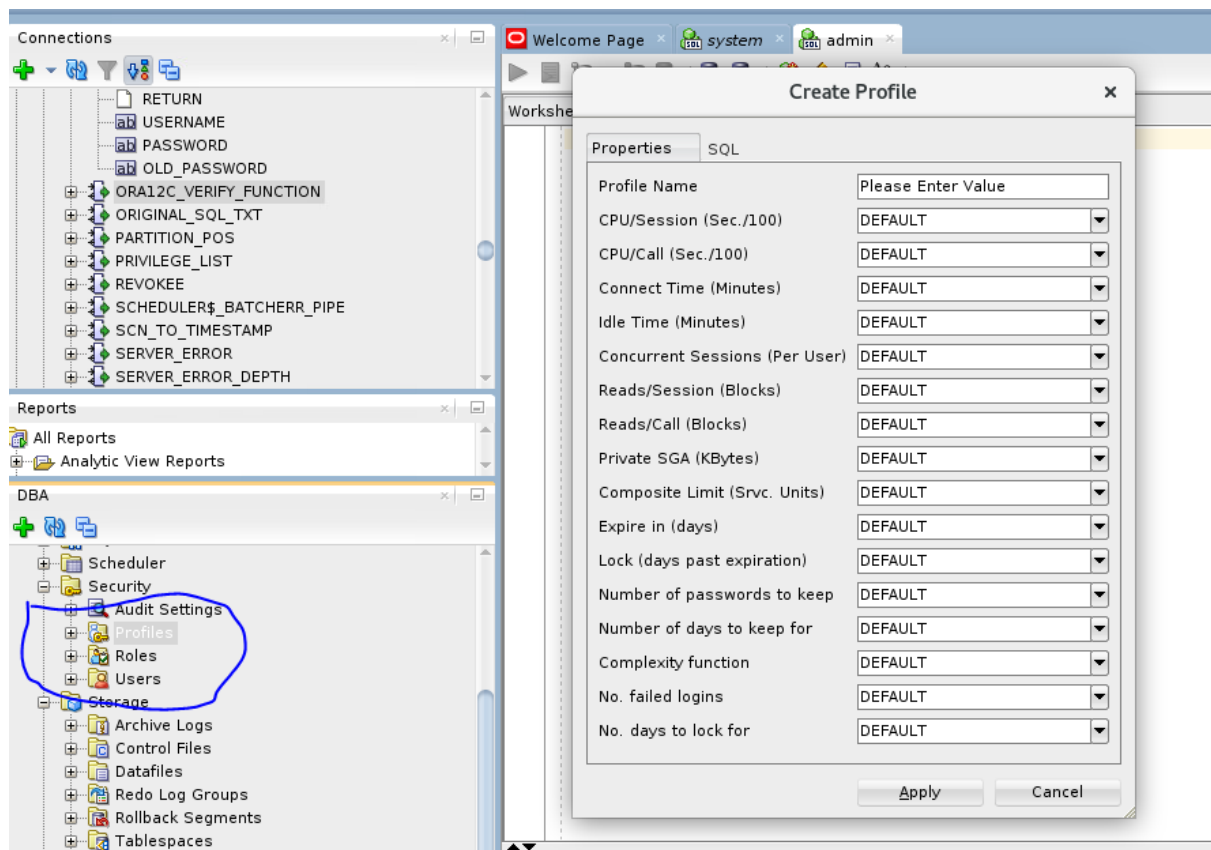


Figure 5: Creating profile using SQL DEVELOPER

After creating the profile and checking the requirement again to what it's been created. One requirement that an old password cannot be reused again has not been implemented. Instead of deleting the created profile and creating a profile again, oracle allows sysdba to edit the profile.

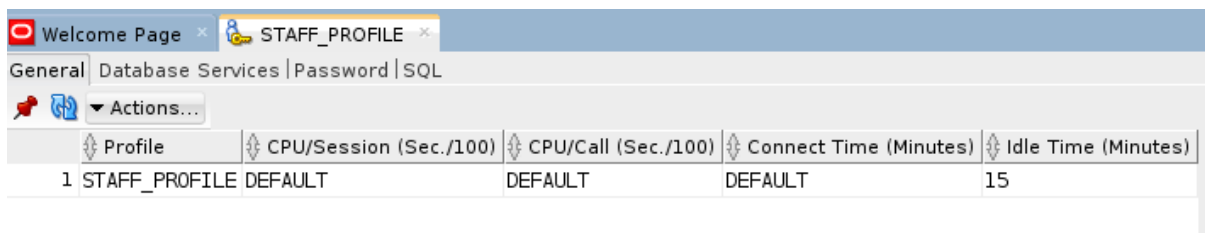
```

1 alter profile staff_profile limit
2 password_reuse_max 1
3* password_reuse_time UNLIMITED
SQL> /

Profile altered.

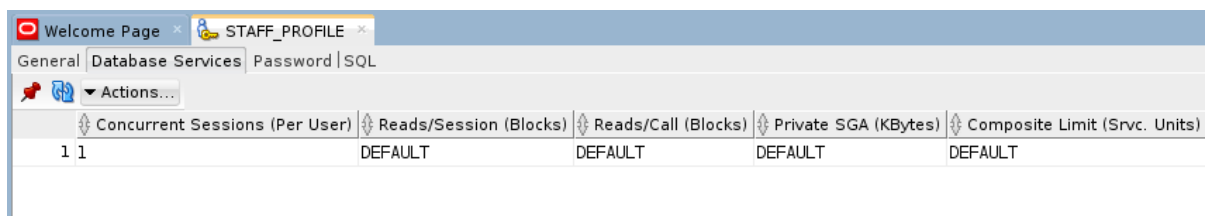
```

Figure 6: Editing Profile to prevent old password from being used.



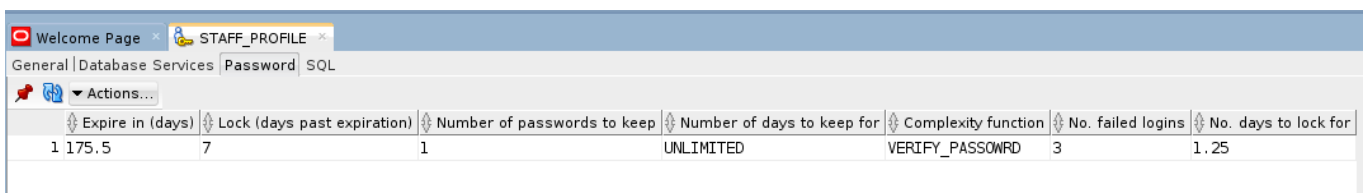
	Profile	CPU/Session (Sec./100)	CPU/Call (Sec./100)	Connect Time (Minutes)	Idle Time (Minutes)
1	STAFF_PROFILE	DEFAULT	DEFAULT	DEFAULT	15

Figure 7: Showing staff profile created using SQL DEVELOPER. The Screenshot shows 15-minute idle time set.



	Concurrent Sessions (Per User)	Reads/Session (Blocks)	Reads/Call (Blocks)	Private SGA (KBytes)	Composite Limit (Svc. Units)
1	1	DEFAULT	DEFAULT	DEFAULT	DEFAULT

Figure 9: The screenshot showing concurrent session has been set.



	Expire in (days)	Lock (days past expiration)	Number of passwords to keep	Number of days to keep for	Complexity function	No. failed logins	No. days to lock for
1	175.5	7	1	UNLIMITED	VERIFY_PASSWORD	3	1.25

Figure 8: Screenshot showing the rest of the requirement implemented

The function verifies the password in figure 10 created to prevent users from entering their password with less than 8 characters. The function allows users to create a password from 8 and up characters. This is only enforcement inside the function, and it meets the user requirement.

```
create or replace FUNCTION VERIFY_PASSWORD (username varchar2,
new_pass varchar2,
old_pass varchar2 )
RETURN boolean AS
BEGIN

IF LENGTH(new_pass) < 8 THEN
raise_application_error(-20003,'Your password must be 8 characters and up');
END IF;
RETURN (TRUE);

END VERIFY_PASSWORD;
```

Figure 10: Function that prevent user of having password that characters less than 8.

```
SQL> CREATE USER TEMOR IDENTIFIED BY 1234567 PROFILE staff_profile;
CREATE USER TEMOR IDENTIFIED BY 1234567 PROFILE staff_profile
*
ERROR at line 1:
ORA-28003: password verification for the specified password failed
ORA-20003: Your password must be 8 characters and up
```

Figure 11: Testing verify function password

The requirement was to create three users that two of them related to the customer service department and one for the inventory departments. The requirement for limiting user recourses to the database is the same such as password and so on. However, accessing database will be different depends on their privileges will be set in the next section. The three users should be allocated to Tablespace such as temp and the default the one created in the first task. To ensure the requirement was created for users to limit privileges and resources will now use it by attaching users to profile staff.

```
SQL> CREATE USER Dept_Cust_1 IDENTIFIED BY ORACLEDBA2019
2 TEMPORARY TABLESPACE TEMP
3 DEFAULT TABLESPACE inv_tsp
4 PROFILE staff_profile;
```

User created.

Figure 12: User One for customer service department

The first user created in figure 11 for the customer service department has the staff profile granted with the two-table space attached to it as it was required in the requirement. The reason for attaching the user to temporary tablespace as well as permanent to improve the concurrency of multiple sort operation, reduce their overhead and avoid oracle database management operations. Notice the password is more than 8 characters that prove our verify password worked as the user-created with no errors.

```

1 CREATE USER Dept_Cust_2 IDENTIFIED BY greenwich2019
2 TEMPORARY TABLESPACE TEMP
3 DEFAULT TABLESPACE inv_tsp
4* PROFILE staff_profile
SQL> /

```

User created.

Figure 13: Second user for customer service department called Dept_Cust_2

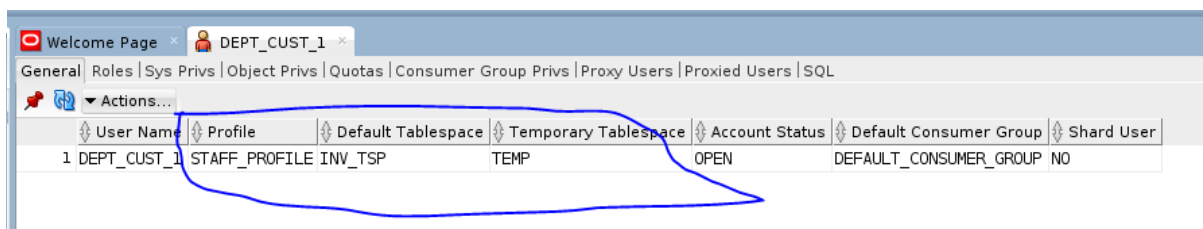
```

1 CREATE USER Dept_invent IDENTIFIED BY FLAS2019
2 TEMPORARY TABLESPACE TEMP
3 DEFAULT TABLESPACE inv_tsp
4* PROFILE staff_profile
SQL> /

```

User created.

Figure 14: The one user for inventory department



	User Name	Profile	Default Tablespace	Temporary Tablespace	Account Status	Default Consumer Group	Shard User
1	DEPT_CUST_1	STAFF_PROFILE	INV_TSP	TEMP	OPEN	DEFAULT_CONSUMER_GROUP	NO

Figure 16: prove that both tablespaces attached to dept _ customer Service

Creating users using SQL developer

- Must be logged in as a sysdba short system administrator with the dba role.
- Navigate to the security folder then right-click on the user then create a new.
- Figure 17 shows how to create a user using SQL developer.

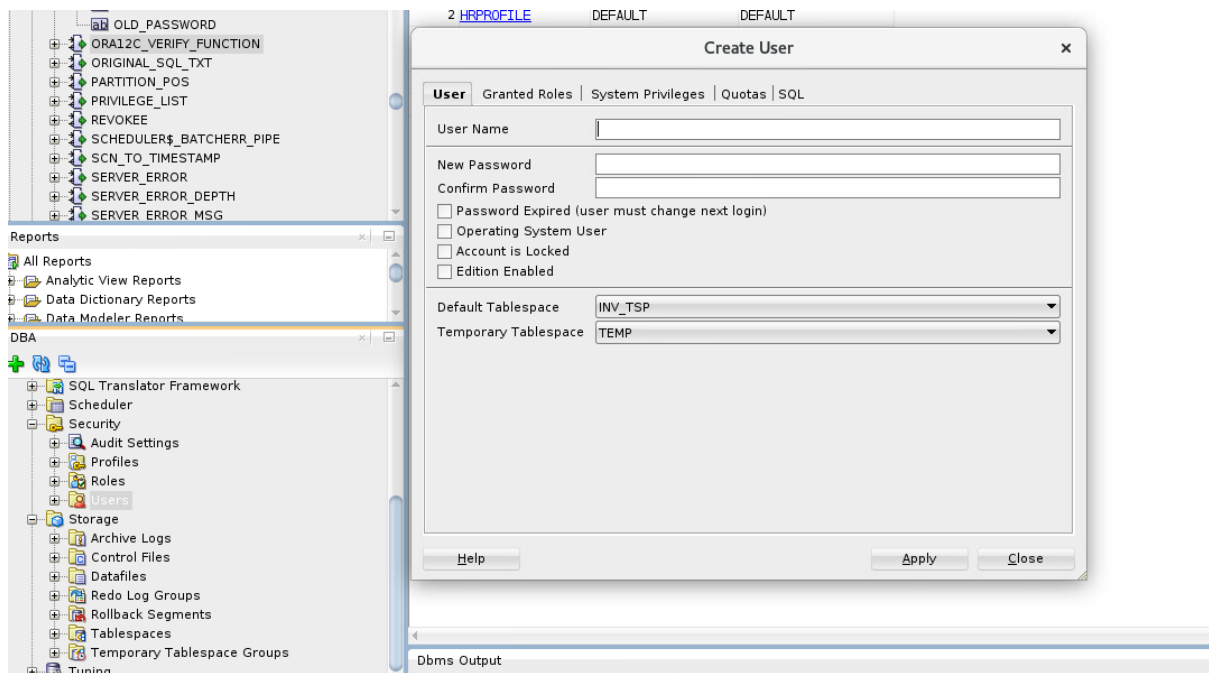


Figure 17: Creating users using SQL Developer

3) All the users should have the following access rights

- a) All the users created above should be able to login to the database.

After creating profiles and users then it was time to test the users created if their account works as they expected. After trying to login with first user Dept_cust_1 user but presented with an error that login denied, and that user needs to create session privilege.

```
Enter user-name: Dept_Cust_1/ORACLEDBA2019
ERROR:
ORA-01045: user DEPT_CUST_1 lacks CREATE SESSION privilege; logon denied
```

Figure 18: user login failed

User privilege is having a right to execute an SQL statement or a chance to access another user object. ORACLE created an enhancement to make the administrator job easier by creating a feature called "ROLES" created by an administrator to group together privileges and assigning the role to users. Instead of having to attach and edit each user within the system and it's also easier to remove them too.

The two roles created are one for the two users for the customer service department and one created for the inventory department. This is again better for the future prospect when the administrator wants to add more users or edit their privileges then all can be done in fewer commands or operations.

```
SQL> CREATE ROLE Dept_CustomerService;  
Role created.  
  
SQL> CREATE ROLE Dept_Inventory;  
Role created.
```

Figure 19: Two roles created. One for customers service users and one for inventory department user

Roles need privileges assigned to them in order to enable get access to the schema or run anything. It created again by only administer to group privileges together and assign them to specific users with specific privileges.

The first privilege assigned to both roles would be the create session privilege to both roles to allow their users to log in to the system. Session per user for users in limited in their profile that each user only allowed to login and create one session per time.

```
SQL> GRANT CREATE SESSION TO Dept_CustomerService;  
Grant succeeded.  
  
SQL> GRANT CREATE SESSION TO Dept_Inventory;  
Grant succeeded.
```

Figure 20: Giving roles session privileges

Figure 21 above shows that both roles now have created session privileges that would allow our users to log in to the database. However, the roles still not been granted to the users.

```
SQL> GRANT Dept_CustomerService to Dept_Cust_1;  
Grant succeeded.
```

Figure 21:Granting Dept_ customer Service role to Dept_Cust_1 user

```
SQL> GRANT Dept_CustomerService to Dept_Cust_2;  
Grant succeeded.
```

Figure 22: Granting Dept_ Customer Service to Dept_Cust_2

```
SQL> GRANT Dept_Inventory to Dept_invent;  
Grant succeeded.
```

Figure 23: Granting Dept _ Inventory to Dept_invent user.

After creating roles, assign sessions then granting roles to our users than its time to test the users again by login into the SQL plus.

The test for three users that we created has passed

```
Enter user-name: Dept_Cust_1/ORACLEDBA2019  
Connected to:  
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production  
Version 19.3.0.0.0  
SQL> █
```

Figure 26. login passed for Dept_Cust_1 passed

```
Enter user-name: Dept_Cust_2/greenwich2019  
Connected to:  
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production  
Version 19.3.0.0.0  
SQL> █
```

Figure 25:login passed for Dept_Cust_2

```
Enter user-name: Dept_invent/FLAS2019  
Connected to:  
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production  
Version 19.3.0.0.0  
SQL>
```

Figure 24: Login passed for Dept_invent

Otherways for creating roles and assign them to users using SQL developer by navigating down to the security folder. Right-click on the roles folder and click on create new. The menu pops up the then right name of the role and clicks apply. Before doing anything you must ensure that you are signed in as sys with role dba as it has privileges to do such a thing.

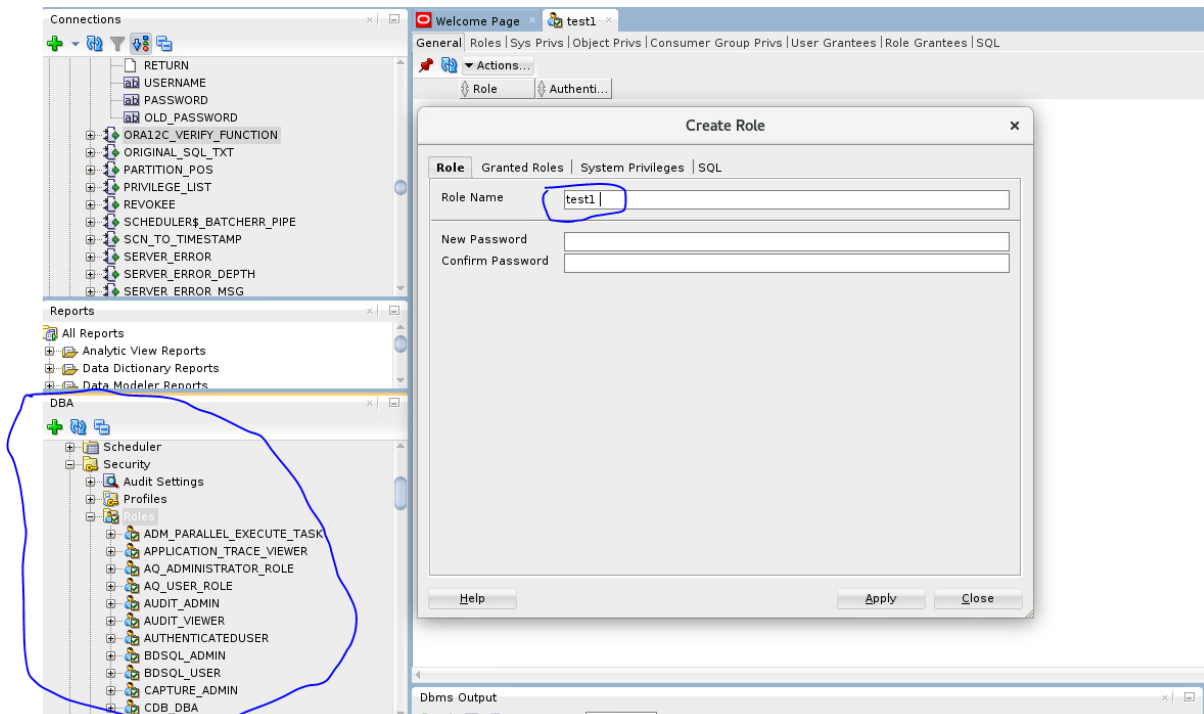


Figure 27: Creating roles using SQL developer

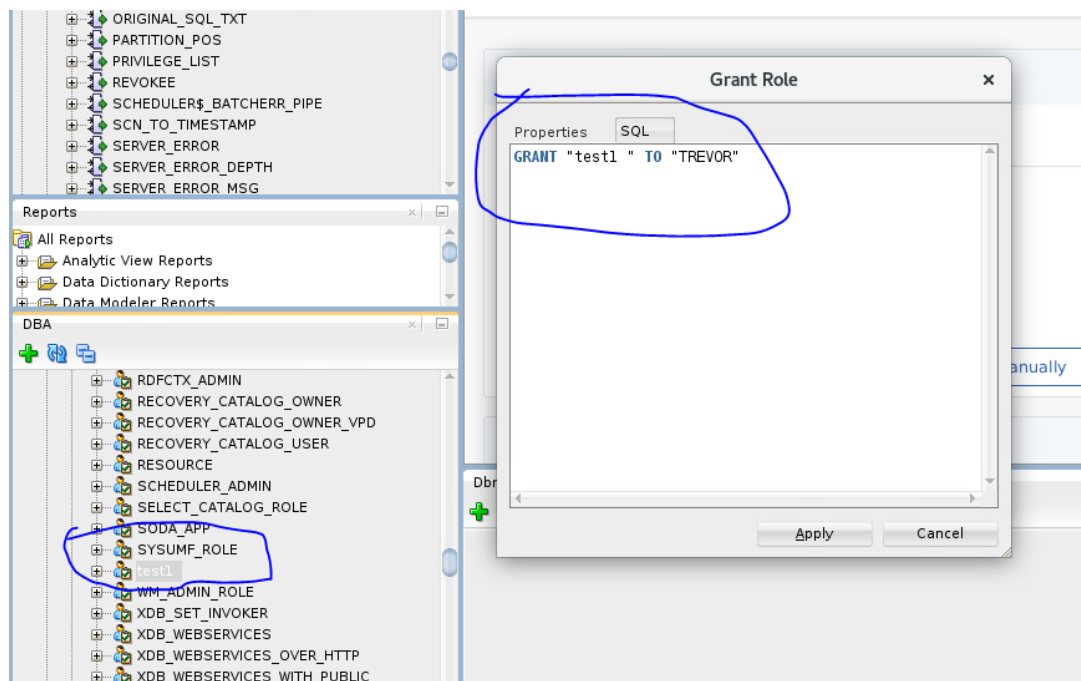


Figure 28: Granting roles to users

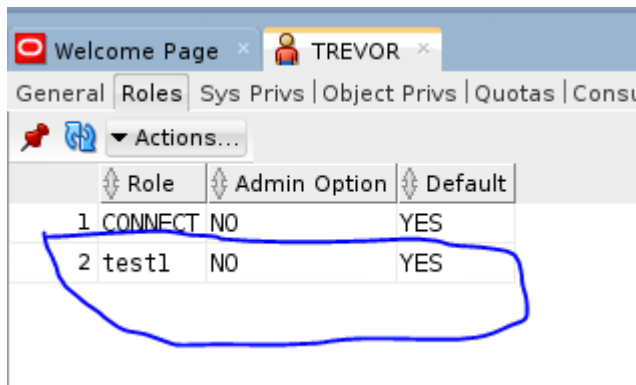


Figure 29: Roles assigned to user Trevor

- b) The two users for the customer department should have READ/ONLY access to countries, products and promotions tables. They should have READ/WRITE accesses to customers and sales tables.

Giving read access only to both customer service customers meaning that both users will be only able to use the select statement command to view that data of countries, products and promotion tables. This means we will be granting select access to both users of the customer service department. Previously created a role for customer service department users that group together privileges so for this situation will be granting the read-only privileges on countries, products, and promotion table to Dept_Customer Service role. Once the privilege granted to the role then the users of that role will get the privilege.

```
SQL> GRANT SELECT ON sh.Countries to Dept_CustomerService;
Grant succeeded.

SQL> GRANT SELECT ON sh.Products to Dept_CustomerService;
Grant succeeded.

SQL> GRANT SELECT ON sh.PROMOTIONS to Dept_CustomerService;
Grant succeeded.
```

Figure 30: Grant read only access to customer service users

The second requirement was to give read and write access for customer service department users on customers and sales tables. This means they have all the CURD operation privilege for both tables.

```
SQL> GRANT SELECT,INSERT,UPDATE,DELETE ON sh.CUSTOMERS to DEPT_CustomerService;
Grant succeeded.

SQL> GRANT SELECT,INSERT,UPDATE,DELETE ON sh.SALES to Dept_CustomerService;
Grant succeeded.
```

Figure 31: Granting read and write access for customer service users on customer and sale table

- c) The user for the inventory department should have READ/WRITE accesses to the table cost and channels as well as READ/ONLY access to CUSTOMERS and Sales. This user should be allowed to create new tables and new views.

Answer

The same command used to give access to customer service users will be used here to give read/ write access or give read-only access. The first one request that the inventory used for the inventory department should have read/ write access to both table cost and channels. Read and write meaning the user giving the CURD operation privileges access such as select, insert, update and delete.

```
SQL> GRANT SELECT,INSERT,UPDATE,DELETE ON sh.COSTS to Dept_Inventory;
Grant succeeded.
```

Figure 33: inventory dept user. Read and write access for table sh. cost

```
SQL> GRANT SELECT,INSERT,UPDATE,DELETE ON sh.CHANNELS to Dept_Inventory;
Grant succeeded.
```

Figure 32: inventory dept user. Read and write access for table sh. cost

Another requirement that was set that dept inventory should also get a read-only access to both sh. customers and sh. sales tables

```
SQL> GRANT SELECT ON sh.CUSTOMERS to Dept_Inventory;
Grant succeeded.

SQL> GRANT SELECT ON sh.SALES to Dept_Inventory;
Grant succeeded.
```

Figure 34: Read access for dept_ inventory user on customers and sales

The last requirement was to give both create the view and create table privileges to inventory department users. After giving those privileges then the user can create the table and create a view in the database.

```
SQL> GRANT CREATE VIEW,CREATE TABLE TO Dept_inventory;
```

Grant succeeded.

Figure 35: Assigning create view and create table privilege to inventory user.

- 4) Create a table inv _ categories inside the newly created schema for the inventory department with the structure described in the table below. The tablespace for this table should be inv _ tsp.

The requirement was to create a table inside the schema for the inventory department with a specific requirement such as adding constraint that the values in the column should not be null, setting up the primary key and ensuring a row called “available” that should be only one character either ‘y’ or ‘no.’ The table created using the sys user with privilege granting it to inventory department schema.

```
SQL> ED
Wrote file afiedt.buf

 1 CREATE TABLE Dept_invent.inv_categories (
 2 cat_id NUMBER(3) CONSTRAINT c_id NOT NULL,
 3 sub_cat_id NUMBER(7) CONSTRAINT s_id NOT NULL,
 4 cat_name VARCHAR2(25) CONSTRAINT c_n NOT NULL,
 5 sub_cat_name VARCHAR2(25) CONSTRAINT s_c NOT NULL,
 6 cat_description VARCHAR2(50) CONSTRAINT c_des_id NOT NULL,
 7 available CHAR(1) CHECK (available IN('Y','N')),
 8 CONSTRAINT invent_PK PRIMARY KEY (cat_id,sub_cat_id)
 9 )
10* TABLESPACE inv_tsp
SQL> /
```

Table created.

Figure 36: Created table inv _ categories for dept _ invent



Figure 37: Table created show inside inventory department schema

The table may appear on the department inventory schema, but the user still has no privilege to access the table for either read or write. After creating the table then assigned privilege for dept inventory to access their own table having read and write access. Notice in figure 33 assigning privileges dept_inventory role rather than the user. The role has only one user attached to it from the inventory department so if later on the department was to add more users to this role and do not want all to have the same privileges then can edit the role or add more roles.

```
SQL> ed
Wrote file afiedt.buf

1* GRANT SELECT,INSERT,UPDATE,DELETE on dept_invent.inv_categories TO Dept_inventory
SQL> /

Grant succeeded.
```

Figure 38: Grant read and write access for inventory department on inv _ categories to dept_inventory role.

After trying to test the privileges by inserting data into this new table using the user dept_invent and received an error that user ORA-01950. The error explained that the user dept_invent have no privileges on tablespace inv _ tsp for creating tables, indexing and executing scripts such as procedures and triggers. I altered the three users and gave them 100M quotas to give them privileges to create objects such as inserting data into tables with permission.

```
SQL> ed
Wrote file afiedt.buf

1* alter user Dept_Cust_1 quota 100M on inv_tsp
SQL> /

User altered.

SQL> alter user dept_invent quota 100M on inv_tsp;

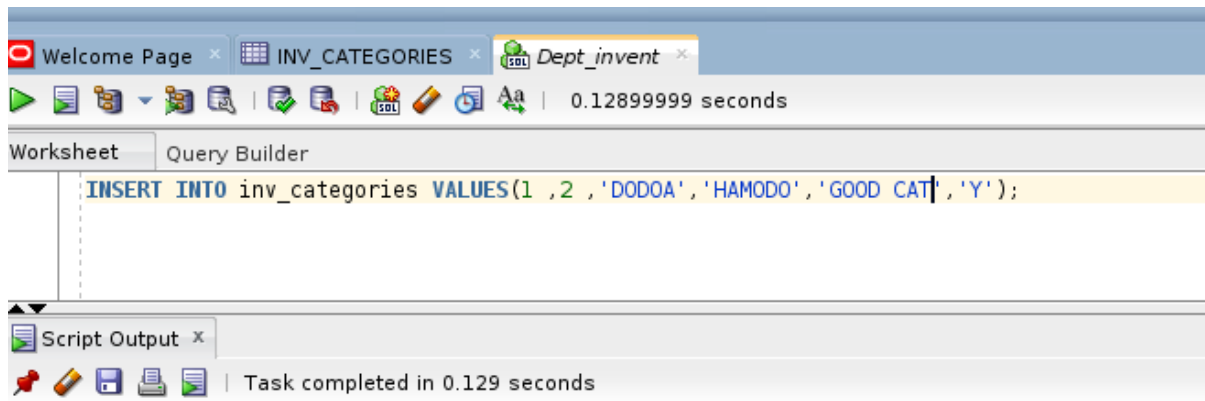
User altered.

SQL> alter user Dept_Cust_2 quota 100M on inv_tsp;

User altered.
```

Figure 39: alter user quotas on inv _ tsp tablespace

Testing by inserting into data inv _ categories after giving quotas and the result was successful.



1 row inserted.

Figure 40: Insert to inv _ categories

	CAT_ID	SUB_CAT_ID	CAT_NAME	SUB_CAT_NAME	CAT_DESCRIPTION	AVAILABLE
1	1	2	DODOA	HAMODO	GOOD CAT	Y

Figure 41: Checking row inserted

- 5) The sh user must be able to run a report with the information about all the product sold for a customer, based on the provided id.

The requirement was to allow the sh user to produce a report about the product sold for a specific customer. The output information should show customer id, first name, product name, quantity sold, and amount sold. All that information should come out from three tables such as customers, sales and products. The connection of table through keys from customers to sales then sales to products. We are assuming that the user sh doesn't have much SQL query knowledge, so we must ensure the solution does not require much knowledge in the SQL query.

There are different types of reports can be generated using SQL developers such as SQL report, interactive report, report on web service result and wizard report.

- SQL report basically creates a report based on custom SQL statements or using PL/SQL function that returning the SQL select statement provided by the user.
- The interactive report can be created using the SQL developer builder by creating a report with parent query and child query.

First solution

The first possible way to solve this problem is by creating a view using a joint statement. The view is a logical table based on one table or more. The table created from the view does not physically exist as it's stored in the oracle data dictionary and does not store any data. To be able to create a view will need Create view privileges if we are normal users. The view gets executed when it gets called.

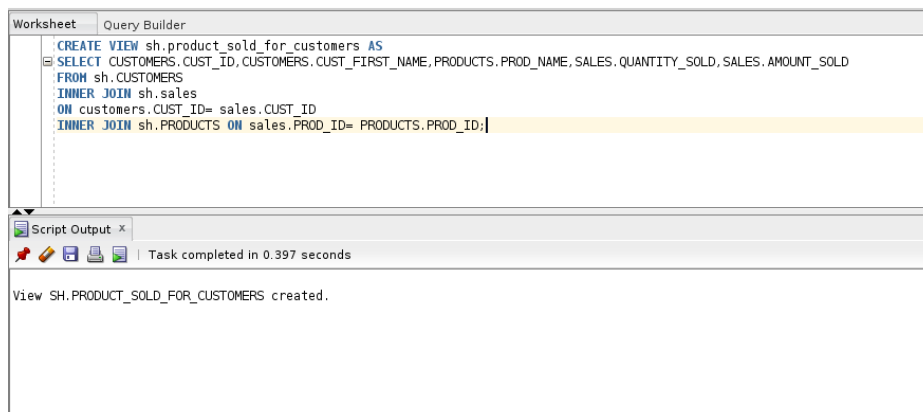


Figure 42: View created for joining three tables

Figure 42 showing the creation of view that brings the product sold for each customer in the customer's table joining customers, sales and products table. After the creation of the view, the way to extract information becomes easier by running a simple query to extract information from the view using “select” * from sh. product_sold_for_customers, where cust_id = 2238 “ as an example.

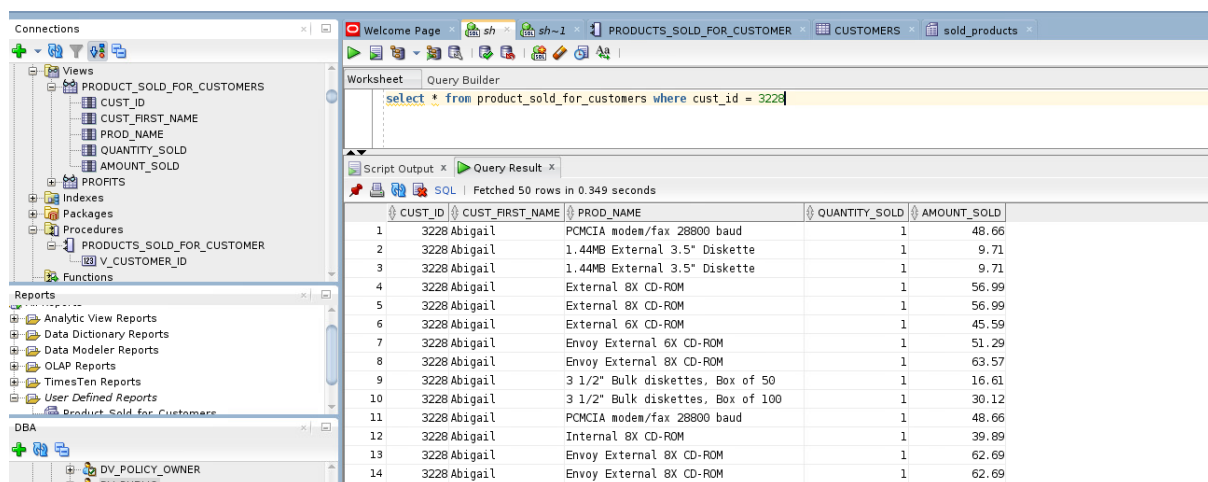


Figure 43: Small query to generate required report from the view for each customer depending to their id

Figure 43 shows running a small query in SQL worksheet for users sh to view all the details required for each customer on what product they bought. The information extracted from the views that join three tables together to give the information about the product sold for each customer extracted using customer ID. Extracting the information generated in the Query result by right click on it then click on export. After that set to PDF then set the title, subject, and location to store the pdf and click export.

Figure 44: Setting out PDF report

CUST ID	PROD NAME	QUANTITY SOLD	AMOUNT SOLD
3228	1.44MB External 3.5" Diskette	1	9.71
3228	3 1/2" Bulk diskettes, Box of 50	1	16.61
3228	Envoy External 8X CD-ROM	1	62.69
3228	Model K8822S Cordless Phone Battery	1	29
3228	Bounce	1	19.29
3228	Model C93822D Wireless Phone Battery	1	21.33
3228	Model SM26273 Black Ink Cartridge	1	26.09
3228	Model A3827H Black Image Cartridge	1	96.66
3228	Model K8822S Cordless Phone Battery	1	32.83
3228	O/S Documentation Set - French	1	48.78
3228	DVD-R Disc with Jewel Case, 4.7 GB	1	7.92
3228	Model SM26273 Black Ink Cartridge	1	29.65
3228	Model NM500X High Yield Toner Cartridge	1	204.44
3228	Model C93822D Wireless Phone Battery	1	22.31
3228	Internal 8X CD-ROM	1	39.89
3228	Model K3822L Cordless Phone Battery	1	17.77

Figure 45: PDF report exported

However, in the way done previously the user still, must write a small query in the SQL worksheet.

Second solution

ORACLE 12C SQL developer provide users with a Report builder or you can call it to report wizard.

Navigate like in figure 46 and right-click on the user-defined report and create a new report

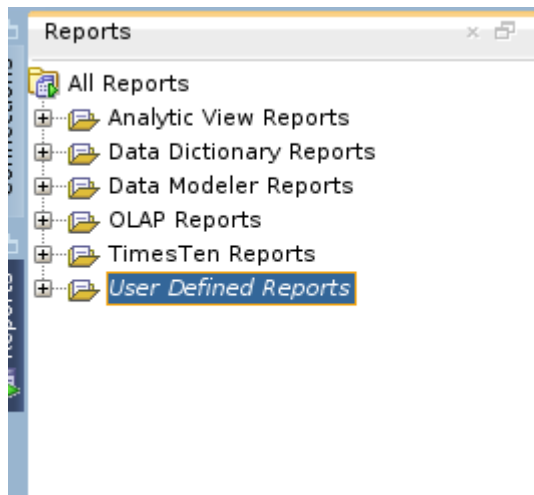


Figure 46:Report builder

Instead of asking the user sh to write the small SQL query, we can use the report builder and generate the user-defined report by asking the user to edit the report by entering customer id to the query then run the report.

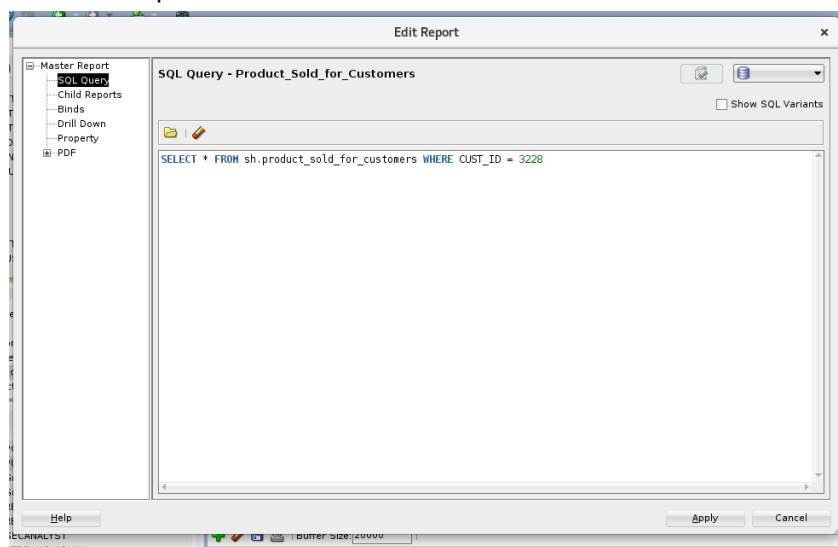


Figure 47: Setting SQL query inside report wizard

However, This is not a great way as the user have to always be able to edit the customer id then run the report to generate into pdf or HTML. Using the report wizard I found out better for creating an interactive report that was mentioned (Smith, 2019) in his own YouTube video. His way does not acquire user knowledge in writing an SQL query to generate a report. In figure 48 the report wizard menu has parent report and child report. The parent report will be used to select all information from the customer table then the child report will have the join statement to get all the requirements to need for the report. What makes this a good way of creating a report as the user

does not need to enter anything other than interacting with the data presented after clicking on the report. In figure 48 notice that the child query takes the ID from the parent query.



Figure 48: Queries used to generate report on report wizard

In figure 49 you can see two table views generated on the screen after clicking on sold products report. When you click on any customer row in the top table then below it will generate the report for product sold for that customer user clicked on. It's basically taking the customer id from of the selected row into the child query automatically to run the full report and that's what called an interactive report.

The screenshot shows the 'sold_products' report with two tables. The top table lists customers, and the bottom table lists products sold for the selected customer. A text box explains that clicking a customer row generates the report for products sold for that customer.

	CUST_ID	CUST_FIRST_NAME	CUST_LAST_NAME
1	49671	Abigail	Ruddy
2	3228	Abigail	Ruddy
3	6783	Abigail	Ruddy
4	10338	Abigail	Ruddy
5	13894	Abigail	Ruddy
6	17449	Abigail	Ruddy
7	21005	Abigail	Ruddy
8	24561	Abigail	Ruddy
9	28116	Abigail	Ruddy
10	31671	Abigail	Ruddy

	CUST_ID	PROD_NAME	QUANTITY_SOLD	AMOUNT_SOLD
1	17449	Mouse Pad	1	11.38
2	17449	External 6X CD-ROM	1	47.82
3	17449	External 8X CD-ROM	1	59.78
4	17449	DVD-RW Discs, 4.7GB, Pack of 3	1	24.07
5	17449	Multimedia speakers- 3" cones	1	47.43
6	17449	DVD-RW Discs, 4.7GB, Pack of 3	1	21.59
7	17449	3 1/2" Bulk diskettes, Box of 50	1	16.51
8	17449	Bounce	1	22.17
9	17449	Smash up Boxing	1	33.72
10	17449	SIMM- 8MB PCMCIAII card	1	135.1

Figure 49: Interactive report

It might be difficult to find a customer going through a lot of data then you can use the find tool in SQL developer that enables you to find a certain data in the first table to generate a report.

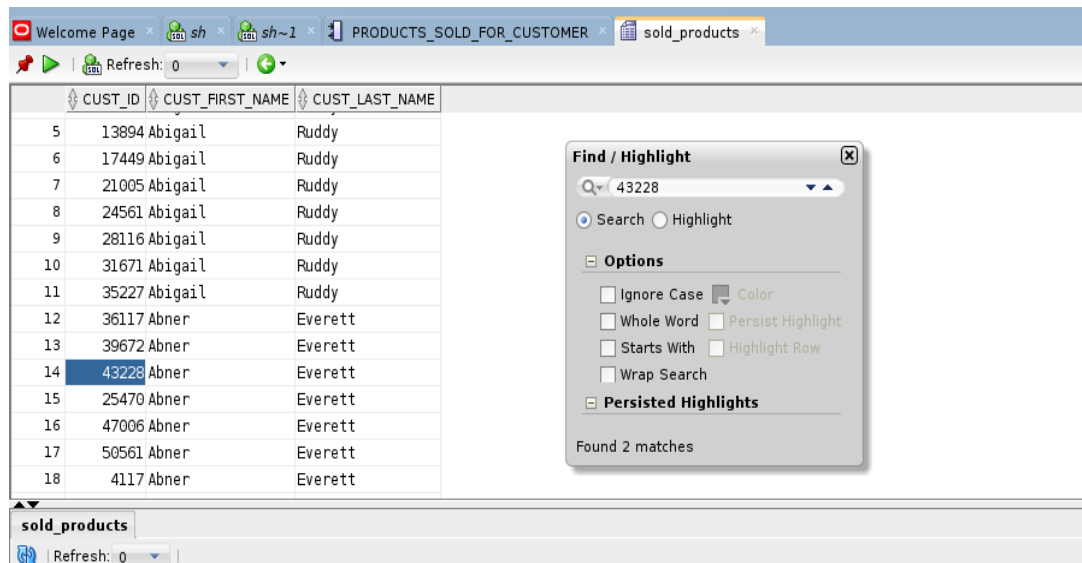


Figure 50: Using find tool to find data

Finally exported data from a report into an HTML view which is another way of viewing a report. In example 45 shows you can also export the report into PDF file.

The screenshot shows a Mozilla Firefox browser window displaying an HTML report titled 'Products Sold for Customers'. The report is a table with the following columns: CUST_ID, CUST_FIRST_NAME, PROD_NAME, QUANTITY SOLD, and AMOUNT SOLD. The data is as follows:

CUST_ID	CUST_FIRST_NAME	PROD_NAME	QUANTITY SOLD	AMOUNT SOLD
3228	Abigail	PCMCIA modem/fax 28800 baud	1	48.66
3228	Abigail	1.44MB External 3.5" Diskette	1	9.71
3228	Abigail	1.44MB External 3.5" Diskette	1	9.71
3228	Abigail	External 8X CD-ROM	1	56.99
3228	Abigail	External 8X CD-ROM	1	56.99
3228	Abigail	Envoy External 6X CD-ROM	1	51.29
3228	Abigail	Envoy External 8X CD-ROM	1	63.57
3228	Abigail	3 1/2" Bulk diskettes, Box of 50	1	16.61
3228	Abigail	3 1/2" Bulk diskettes, Box of 100	1	30.12
3228	Abigail	PCMCIA modem/fax 28800 baud	1	48.66
3228	Abigail	External 6X CD-ROM	1	45.59
3228	Abigail	Envoy External 8X CD-ROM	1	62.69
3228	Abigail	Envoy External 8X CD-ROM	1	62.69
3228	Abigail	Internal 8X CD-ROM	1	39.89
3228	Abigail	3 1/2" Bulk diskettes, Box of 50	1	16.61
3228	Abigail	3 1/2" Bulk diskettes, Box of 100	1	30.12
3228	Abigail	3 1/2" Bulk diskettes, Box of 100	1	30.12
3228	Abigail	Internal 8X CD-ROM	1	39.89
3228	Abigail	PCMCIA modem/fax 28800 baud	1	48.66
3228	Abigail	PCMCIA modem/fax 28800 baud	1	48.66
3228	Abigail	1.44MB External 3.5" Diskette	1	9.71
3228	Abigail	External 6X CD-ROM	1	45.59
3228	Abigail	External 8X CD-ROM	1	56.99
3228	Abigail	Envoy External 6X CD-ROM	1	51.29
3228	Abigail	Envoy External 8X CD-ROM	1	62.69
3228	Abigail	Envoy External 8X CD-ROM	1	62.91
3228	Abigail	Internal 6X CD-ROM	1	34.67
3228	Abigail	Internal 8X CD-ROM	1	39.89

Figure 51: Generating report using HTML view

Creating the report using SQL plus

Again, if the user sh has not enough knowledge of running SQL queries. This solution works by creating a procedure in the schema for this user to execute by sending one parameter which is the customer id. Understanding how the procedure works are done by the system writing documentation that explains simply to user sh and other users how to do certain things within their own privileges given. The procedure extract information from the view that created sh.products_sold_for_customers depends on the customer ID sent to the parameters

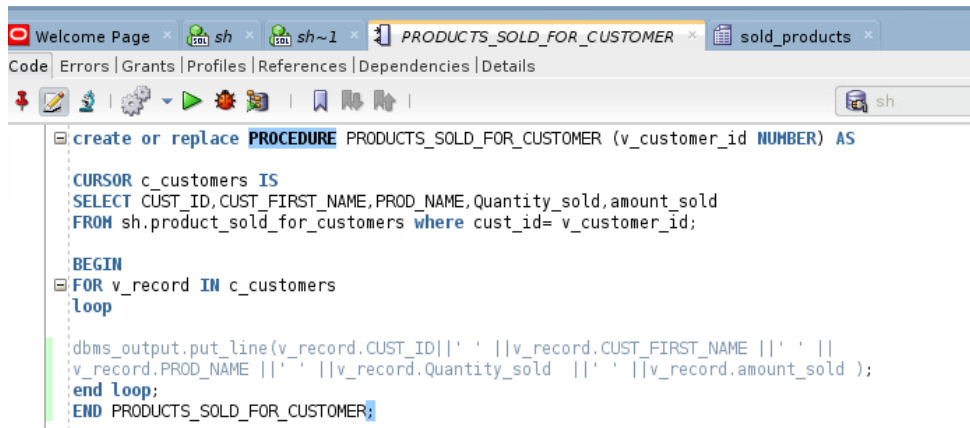


Figure 52: SQL PL for extracting information needed for the report

However, the user needs to execute the procedure in SQL plus if SQL developer not found and be able to export to report. Using Spool technology, we can extract the report from the SQL plus.

```
SQL> set serveroutput on
SQL> spool /home/oracle/spool/sold_products_for_3228
SQL> execute PRODUCTS_SOLD_FOR_CUSTOMER(3228);
```

Figure 53: Using spool technology to extract information into the report

In conclusion, for someone with no background in the SQL query, I would suggest using the second solution with the use of SQL developer. However, it all requires some knowledge to enter and do a certain thing within the data. Providing documentation on how to do a certain task would be useful for the user to understand how to run a certain task. The question was providing a solution without needing to have much knowledge in SQL and all the three solutions won't require much knowledge to use them.

Problem Solving Requirements:

- 6) If you accidentally dropped an important schema (for example, hr) in your database, discuss the best functionality to use to recover it quickly and safely. Provide an explanation of the steps needed to recover data including the SQL commands for each step.

<https://dbaclass.com/article/how-to-recover-a-dropped-user-using-flashback-oracle/>

The technology that would be used to solve this issue would be the flashback technology. This technology will help us to flashback the database to the past when that user was available then take an export dump of the schema to restore it to the database to the same state when it was dropped.

Perquisite

- The database must be opened in archive mode
- Flashback must be turned on

You can check the flashback turned on by writing this query. In our case, it has been turned off. Select flashback_on from V\$DATABASE.

To test this situation, I decide to create a user with tables then drop the user again and use the flashback technology to retrieve the user with their object. However, my experiment ended with where I need to turn on the flashback technology due to the risk of messing up the system.

Firstly, I created the user hr1 to be dropped using sys dba account.

```
SQL> create user hr1 identified by oracle;
```

User created.

Figure 54: Creating user for test

After creating the user then need to assign the session to the user to allow them to log in to the database and granting privileges to create the table.

```
SQL> grant create session to hr1;
```

Grant succeeded.

Figure 55: Granting sessions to a user

```
SQL> grant create table to hr1;
```

Grant succeeded.

```
SQL> connect hr1/oracle
```

Connected.

```
SQL> create table test(id number);
```

Table created.

After creating the table and setting data inside the table then its time to insert data inside and afterward get the current-SCN number that gives us a state number that can help us to retrieve the database to a certain state before the user got dropped. Usually, you use the timestamp of the last action to retrieve the database back, but you can use the SCN to retrieve the information back. You can always use the flashback version query to get details of SCN and all the changes made throughout, so you understand to what version you need to get the database back.

FLASHBACK DATABASE TO TIMESTAMP (SYSTIMESTAMP – INTERVAL '30' MINUTES); This will flash the database back to half-hour. Obviously, this can be changed depends on how much you want to go back. Or you can use the currents SCN extracted from the flashback version query after you run the command.

```
SQL> insert into hr1.test2 values(3);

1 row created.

SQL> select current_scn from V$DATABASE;

CURRENT_SCN
-----
        6771204
```

Figure 56: Inserting data into table for testing

After getting the SCN number then its time to drop the user.

```
SQL> drop user hr1 cascade;
```

User dropped.

Figure 57: User drop

The user has been dropped and now its time to recover the user with his own objects. Use this command to start retrieving the schema hr1 with its own objects.

```

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.

Total System Global Area 1157627168 bytes
Fixed Size                  8895776 bytes
Variable Size               920524000 bytes

```

Figure 58: Shutting down the database and start in mount

```

SQL> flashback database to scn 67717204;
flashback database to scn 67717204
*
ERROR at line 1:
ORA-38726: Flashback database logging is not on.

SQL> select flashback_on from V$DATABASE;

FLASHBACK_ON
-----
NO

SQL> ALTER DATABASE FLASHBACK ON;
ALTER DATABASE FLASHBACK ON
*
ERROR at line 1:
ORA-38706: Cannot turn on FLASHBACK DATABASE logging.
ORA-38707: Media recovery is not enabled.

```

Figure 59 Using Flashback technology for recovery

After flashing back, the database using the SCN number or timestamp before the user has been dropped then use this command below.

- ALTER DATABASE OPEN READ ONLY command used to check the data.
- Select * from hr1.test2;

The user test schema now its back in the database table and its own object but however anything is done with.

7) Your users are reporting “ORA-1555: Snapshot too old” errors. What might be the cause of this? Explain what you need to do in order to solve this issue. All explanation below derived from (Enmotech Blog, 2019).

This error can be caused by one of the problems described below

- This error can be caused due to the result of having insufficient rollback segments. When there are too many transactions that are modifying data, performing commits and rollback this error would appear due to limitation of rollback segment size. If the rollback segment size too small, then all rollback data is overwritten meaning the data deleted will be lost. To ensure this error would not appear is by making larger rollback segments available to ensure the rollback data kept for a longer time.
- This error can appear due to having the program not closing cursor after using fetch and update statements. To avoid ensure the cursor is closed when not needed anymore.
- This error can happen due to having the FETCH is executed after a COMMIT issued. This would make the number of rollbacks recorded once the last cursor closed to fill the rollback segments and that will start to begin overwriting the earlier recorded.

8) Discuss how different database failures can impact a business, including the financial costs, and what can be done to minimize the risk. Please include into discussion the actions that should be taken to prevent future problems

Type of database failure

- System crashes such as software, hardware or network error that causes data transactions to fail.
- Transaction or system error caused by the user interrupting the transaction or some operation caused failure.
- Concurrency control when several transactions may be in deadlock and that transaction can be aborted to be restarted later.
- Network failure: This failure can occur when you are using a client-server configuration or distributed database system where multiple database servers connected by communication networks. Having those types of failure such as communication software failures such as communication software failure or aborted asynchronous connection will interrupt the normal operation of the database system.
- Media failure is the most dangerous failure due to the reason that it can result in loss of data if proper ways of data recovery have been followed before. It also takes a lot of time to able to recover the database than any other issue. An example of media failure would be a disk controller failure or disk head crash which results in the database residing on the disk will be lost.
- Natural and physical disasters can cause data damage such as fires, earth quick and power failure.

Impact on a business

There is an impact on business in so many ways when any database failure happens. The first risk thought would be losing customers or client contacts that were stored in the database. This means the risk will be on sales, revue, and profitability because the business cannot contact their own customers about their new products and so on. This is from case to

case because for example large companies now are deepened on their call centres and highly depended on their customer relationship management system that helps to track information about customer issues and sales. Having a severe failure with the database can bring the business that relies on the database to propel it into bankruptcy.

Having an extreme data loss of the entire database or even temporarily can impact on the organization to fail. The organization during the database failure cannot be able for example to update their employee's records, fulfil the order, manufacture goods, produce financial reports or provide service. There are risks as well when even if the database failure temporarily for example employees can be asked to wait for the database to be recovered until to start working on their tasks and that can reduce productivity. Another case scenario that loss of data makes it difficult for managers to measure the company's profitability because of the inability to get information about the business such as sales, orders and so on. Having not been able to gather information on certain indicators in the business meaning the managers are running blind as to the health of the business. Lost of data can affect decision making which leads to reduced revenue and higher expenses to recover the data leading to loss of profitability.

Data loss in companies can lead to lawsuits and fines such as shareholder lawsuits that failure to protect data could lead to litigation. An example of litigation when the data loss tied to a negative change in the share price of the company stock. Other legal actions that can be sued for failure to perform duties that written in contracts or not being able to produce goods and services that they have been paid for. Another case that when a customer loses data then it can sue the company for direct and collateral damage. All those situations explained can lead to financial harm to the business.

Actions that should be taken to prevent future problems.

The best actions to consider in terms of database failures would be ensuring to have back up recovery mythology running in place. Having always a back up of the system or database solve this problem as there are different causes to database failure and not all measured always as an example human error. There are different types of back up for each type of database but as this for this prototype requirement all looking at oracle database then its clear to explain about the type of recovery can be used to ensuring database recovered. Also, the company can look at consistent back up and using cloud for storing their own data. All the actions mentioned in (Isaca.org, 2019) article that will be explained in below sections.

- **Logical backups**

This type of backup created using the oracle utilities “exp” for all the Oracle versions. Data pump can be used in this situation. The whole database, tables, tablespace and individual schemas can be backed up. Restoring the data can be done using the utilities provided by Oracle called “imp” or data pump. Using this backup to recovery to the point of failure is not a possible way.

- **Physical offline or cold backup**

Running this type of backup require the database to be shut down then the copy made with all the data files and other component of the database.

- Physical online or cold backup

This way of backup can be used when the database is up and running. There are points that should be kept in mind when using this type of backup.

Firstly, either ensure that tablespaces are in backup mode and backup the rest of files using the operating system copy command or use Oracle tool provided called RMAN that created for backup and recovery. RMAN technology uses the database control file to keep its catalogue or DBA set up schema for each database in a separate database for RMAN catalogues. DBAs need to understand the different types of backup such as full, incremental and differential backups that are set using RMAN scripts. The DBA must also put the database in archive mode and backup all the archived redo logs. Lastly, it's very important that the DBA must not forget to backup at the end of each backup the RMAN catalogue. The DBA can do the export backup of the RMAN catalogue schema.

9) Evaluation

The solution implemented for the requirement presented to us by the company was implemented using SQL PLUS and SQL DEVELOPER. The process would have been easier if we had been provided with an Enterprise management system.

Firstly, looking at the course requirement it seemed an easy course work as we already did most of it in our lab work submission. However, most of the requirements can have many different solutions. Each question took me quite a lot of time to research ensuring my solution is good but also to explain different ways of solving the same solution.

The prototype solution created has all the solutions for the problems provided. The tablespace created with all the requirements and the tablespace has datafile created as backup in case the default data file corrupted then recovery datafile can replace the existing one. The requirement for creating users for customer service and inventory department is a strong solution as I created two profiles which each relate to a department.

A profile is a collection of attributes that apply to a user which makes it a point of reference for multiple users with the same attributes. For now, the customer service department has two users with a set of rules and privileges which, all the rules set on the profile and attached to both users or delete users or remove certain rules.

This makes it easier in later for DBA to create more users for the customer service department as it only needs to alter new users to connect with the profile. This only works when all users require the same attributes. After setting rules for the users then another requirement was setting privileges for users to be able to use the database for their own needs. The same concept applied with a profile by creating a single point of reference for multiple users, but this time called Roles. Again, two roles created due to the reason that both users from the customer's service department have the same privileges and one for the only user from the inventory department.

All privileges for the required tables, creating sessions, tables, views and curd operations were set to the roles then roles granted to customer service and inventory department users. After creating the users and granting them privileges then trying to insert into tables with privileges but presented with an error that the user does not have tablespace privileges for inserting. I found that even when you grant user privileges for curd operations and attach a tablespace for the user is not enough for inserting data or do any CURD operations or others. The error was explained online that each user needs a specified size referred to as quotas in the tablespace and it can be unlimited as well but still need to be specified so users can use all their privileges. This taught me that always test implementation after created to ensure the specified requirement works correctly as expected. Having a confirmation of the script worked correctly is not enough when you are logged in as sys dba.

Running a test after each requirement helped me to ensure everything works correctly which shows the solution prototype solves the requirements given at the first stage. In terms of the problem-solving question, I would have liked to be able to test the solution especially the first question when the flashback database needs to be altered and archive altered on as practical always helps to understand the theory provided better.

This term time management was not great at all but it's understandable that was my first term for doing MSC in Big Data and Business intelligence. However, the structure of this course was set perfect as all covered in the lab and lectures are the same thing for course work. The flip classroom forced us to understand and read the whole presentation before the lecture and do the QUIZZ as will be tested in the class. This type of technique helped us to manage my time well and not spend more time on this course work as we usually spend at least 8 hours every week on this course work which is enough to deliver this course work.

Oracle provides documentation online that I found is very helpful to explain all the necessary information needed to complete this course work. The lecture slides and videos provided explain all the necessary information with feedback from the labs to prevent any necessary mistakes from being done again.

I have learned a lot from this course about research skills and logical thinking. I would say that I have learned to be thorough to always check whether the solution created can be better using different techniques. I would say the lecturer taught us to always evaluate our methodology to ensure that the solution created works for all. I have learned to think always of the user using the program after being completed to ensure put the user requirement at heart. After completing this course, I feel confident to speak about database architecture and implementation to clients. During the summer holiday, I will apply for the oracle exam to become a certified Oracle database administrator.

The changes would be making to the system would be measuring the quotas referred to a space needed for each user in the tablespace. The second change I would make on the report by formatting report nicely for the three-solution created. Apart from that, I think most of the solution provided is very strong and it fits the requirement requested. The last

changes would be improving my time management to be able to take on the challenge to complete the extra credit requirements.

10) Completed requirement

DBA requirement	
1	Fully implemented
2.a	Fully implemented
2.b	Fully implemented
2.c	Fully implemented
2.d	Fully implemented
2.e	Fully implemented
3.a	Fully implemented
3.b	Fully implemented
3.c	Fully implemented
4	Fully implemented
5	Fully implemented
6	Fully implemented
7	Fully implemented
8	Fully implemented
9	Fully implemented

11) Strength and additional features

- Requirement 1-5 build strongly and tested out throughout the process and screenshots included.
- Three types of scenarios implemented for building the report in the requirement 5. One of the scenario mentioned as interactive report type in ORACLE documentation.
- Alternative tools provided that can be used for each requirement as an example at the end of each requirements.
- Due to time management the tadeonal feature has not being implemented.

12) Conclusion

This provided have provided the solutions for both Part 1 and part 2 for the provided problem. All the requirements achieved with good technique of problem-solving such as making the solution scalable for future changes using the profile and roles and so on. The future improvement would be looking at providing back up the database and providing a working solution for the problem-solving questions for users to understand practically rather than theory on how to solve a certain solution.

13) References

Docs.oracle.com. (2019). *CREATE PROFILE*. [online] Available at: https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6010.htm [Accessed 24 Nov.

Docs.oracle.com. (2019). *Creating Tablespaces*. [online] Available at: https://docs.oracle.com/cd/B28359_01/server.111/b28310/tspaces002.htm#ADMIN01102 [Accessed 24 Nov. 2019].

Docs.oracle.com. (2019). *Managing User Privileges and Roles*. [online] Available at: https://docs.oracle.com/cd/A97630_01/server.920/a96521/privs.htm#:~:targetText=A%20user%20privilege%20is%20a,together%20privileges%20or%20other%20roles. [Accessed 25 Nov. 2019].

Isaca.org. (2019). *Database Backup and Recovery Best Practices*. [online] Available at: <https://www.isaca.org/Journal/archives/2012/Volume-1/Pages/Database-Backup-and-Recovery-Best-Practices.aspx> [Accessed 12 Dec. 2019].

Oracle Tutorial. (2019). *Oracle CREATE PROFILE: Setting Database Resource & Password Limits*. [online] Available at: <https://www.oracletutorial.com/oracle-administration/oracle-create-profile/> [Accessed 24 Nov. 2019].

Petrocelli, T. (2019). *Data Loss and Business Risk | The Changing Face of Data Protection | InformIT*. [online] Informit.com. Available at: <http://www.informit.com/articles/article.aspx?p=422303&seqNum=4> [Accessed 12 Dec. 2019].

Smith, J. (2019). *YouTube*. [online] Youtube.com. Available at: https://www.youtube.com/watch?v=_AtQbENBCE0 [Accessed 12 Dec. 2019].

Enmotech Blog. (2019). *Ways to avoid ORA-01555 snapshot too old error...!!*. [online] Available at: <https://blog.enmotech.com/2018/09/10/ora-01555-snapshot-old-error-ways-to-avoid-ora-01555-snapshot-too-old-error/> [Accessed 12 Dec. 2019].

14) Bibliography

CloudSecureTech. (2019). *What are the causes of Database Loss?*. [online] Available at: <https://www.cloudsecuretech.com/what-are-the-causes-of-database-loss/> [Accessed 12 Dec. 2019].

Dlsweb.rmit.edu.au. (2019). *Backup and Recovery - Types of Database Failure*. [online] Available at: https://www.dlsweb.rmit.edu.au/Toolbox/knowmang/content/backup/failure_types.htm#top [Accessed 12 Dec. 2019].

Docs.oracle.com. (2019). *Authorization: Privileges, Roles, Profiles, and Resource Limitations*. [online] Available at: https://docs.oracle.com/cd/B19306_01/network.102/b14266/authoriz.htm#i1007654 [Accessed 12 Dec. 2019].

Docs.oracle.com. (2019). *Managing User Privileges and Roles*. [online] Available at: https://docs.oracle.com/cd/A97630_01/server.920/a96521/privs.htm [Accessed 12 Dec. 2019].

Sciencedirect.com. (2019). *Database Failure - an overview | ScienceDirect Topics*. [online] Available at: <https://www.sciencedirect.com/topics/computer-science/database-failure> [Accessed 12 Dec. 2019].

WellData. (2019). *The consequences of a database failure | WellData*. [online] Available at: <https://www.welldata.co.uk/consequences-database-failure/> [Accessed 12 Dec. 2019].



Course Title: Advanced Database Technologies

Student Name: Mohamed AL-Kaisi

Student ID : 000931504

Part 1

Contents

Lab 2: Creating a Database.....	4
Lab3: Managing database instance	7
Lab 4: Managing database security	8
Lab 5: Managing tablespaces and other storage structures	10
Lab6: Managing schema objects and data.....	14
Lab 7: Using flashback technology.....	20
Lab 8: Moving data between systems	24
Lab 9: Managing backup and recovery	26

Table of Figures

Figure 1: Definition of variables	4
Figure 2: Control files created	5
Figure 3:Redo log file	5
Figure 4:Created Datafiles	6
Figure 5: Number of extent created	10
Figure 6: Tablespace taking most space	11
Figure 7: SQL developer filter	11
Figure 8: Tablespace size	13
Figure 9: Required table created	14
Figure 10: Inserting data into table.....	15
Figure 11: Using explain plan for	16
Figure 12: Prove of index created	17
Figure 13: Finding out status and serial id	18
Figure 14: Checking active sessions and taking SID and serial key to kill sessions	19
Figure 15: Query to kill the session.....	19
Figure 16: Session killed	19
Figure 17: Dropped objects.....	20
Figure 18: Bin with objects inside	21
Figure 19: Executing Flashback version query	22
Figure 20: Logical VS physical	24
Figure 21:Tablespace allocated for admin	25
Figure 22: Datafile physical structure location where the tablespace stored and this means the physical storage is system01.dbf for table admin .employees.....	25

Lab 2: Creating a Database

Question 1: *What are 'file location variables' and their purpose? Explain.*

Answer:

Firstly, the name above can have a different explanation, from the name before knowing anything about oracle database but have some basic programming. You can understand that those are variables that files location stored inside it. File location variables used to identify clearly the location of where the database stored within the UNIX system or other system.

b) Which variables have been created and what values do they have?

ORACLE_BASE_HOME: This file location specifies the directory including the oracle software.

DB_UNIQUE_NAME: This is the name of the database.

ORACLE_BASE: This environment variable specifies the base directory structure for optimal flexible architecture compliant installations.

DB_NAME: This is the name of the database that will be created.

ORACLE_HOME: This name represents the home directory of the user where the database will be stored.

SID: This name represents the database that is used at the given time.

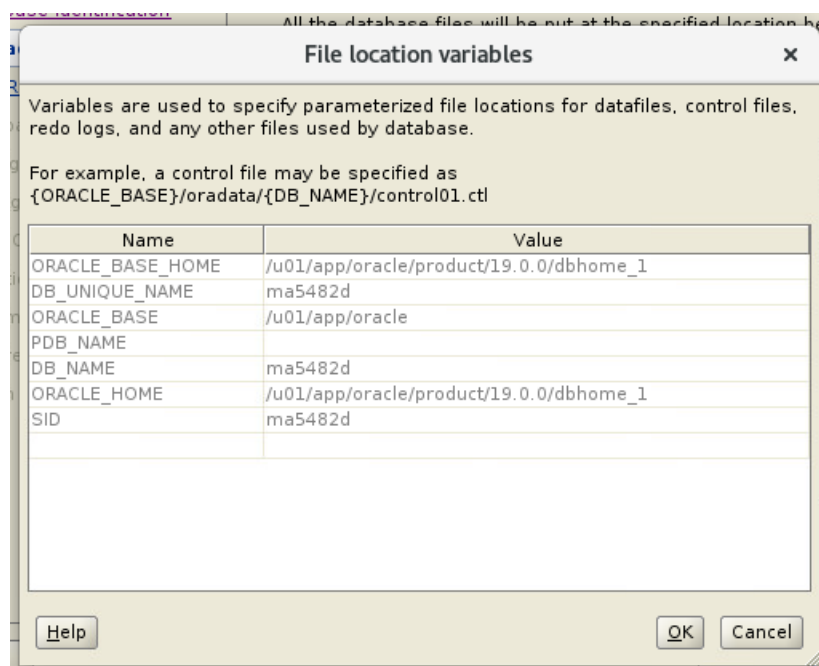


Figure 1: Definition of variables

Question 2: How many control files have been created and where they are located (the exact directory path)? Provide a screen-shot and brief explanation.

Answer: Each created database has a control file that considered as a small binary file to record the physical structure of the database. Control files includes database name, location of associated datafiles and redo log files, timestamp of the database creation, current log sequence number and finally check point information. In oracle documentation mentioned that without the control file, the database recovery can be very difficult to do. The file is created as soon as the database created which at least one copy of the control file created during the creation of the database. (Docs.oracle.com, 2019). When I created database, two control files have been created such control one and control 2, see screenshot below.

Screen-shot (if required):

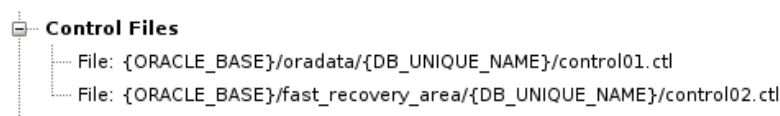


Figure 2: Control files created

Question 3: How many redo log groups have been created? What is the location (the exact directory path) of the online redo log files? Provide a screen-shot and brief explanation

Answer: This is the most important files that needed for recovery operation containing two or more files that operate by storing all changes made to the database. All the instances of oracle database associate redo log to protect the database in case of any unknown failure. Three redo log files created during the creation of the database.(log,2019)

Screen-shot (if required):

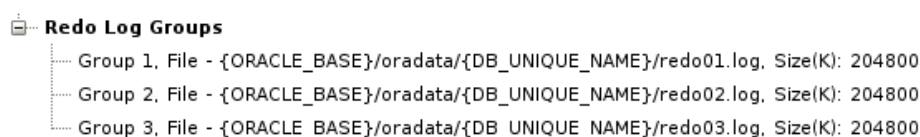
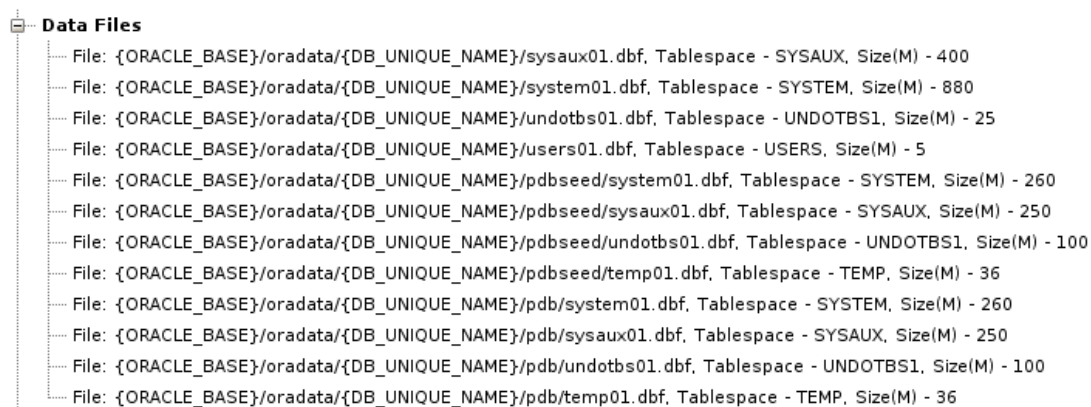


Figure 3:Redo log file

Question 4: *How many data files have been created and where they are located (the exact directory path)? Provide a screen-shot and brief explanation.*

Answer: In oracle database any table space has one or more physical datafiles. Data files can only be linked with one table space and one database. Oracle generate a datafile for a table space by specifying amount of disk space plus the overhead needed for the file header. After the creation of the data file, in this case we are using Linux operating system then it will be responsible for clearing old data and permission from the file before assigning to oracle. 12 datafiles have been created during the creation of the database.(files,2019)

Screen-shot (if required):



The screenshot displays a list of 12 datafiles under the heading "Data Files". Each entry specifies the file path, the tablespace it belongs to, and its size in megabytes (M). The files are as follows:

- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/sysaux01.dbf, Tablespace - SYSAUX, Size(M) - 400
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/system01.dbf, Tablespace - SYSTEM, Size(M) - 880
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/undotbs01.dbf, Tablespace - UNDOTBS1, Size(M) - 25
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/users01.dbf, Tablespace - USERS, Size(M) - 5
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/pdbseed/system01.dbf, Tablespace - SYSTEM, Size(M) - 260
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/pdbseed/sysaux01.dbf, Tablespace - SYSAUX, Size(M) - 250
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/pdbseed/undotbs01.dbf, Tablespace - UNDOTBS1, Size(M) - 100
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/pdbseed/temp01.dbf, Tablespace - TEMP, Size(M) - 36
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/pdb/system01.dbf, Tablespace - SYSTEM, Size(M) - 260
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/pdb/sysaux01.dbf, Tablespace - SYSAUX, Size(M) - 250
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/pdb/undotbs01.dbf, Tablespace - UNDOTBS1, Size(M) - 100
- File: {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}/pdb/temp01.dbf, Tablespace - TEMP, Size(M) - 36

Figure 4:Created Datafiles

References

Docs.oracle.com. (2019). *Managing Control Files*. [online] Available at: https://docs.oracle.com/cd/B19306_01/server.102/b14231/control.htm [Accessed 9 Oct. 2019].

log, R. (2019). *What Is the Redo Log?*. [online] Docs.oracle.com. Available at: https://docs.oracle.com/cd/B28359_01/server.111/b28310/onlineredo001.htm#ADMIN11302 [Accessed 9 Oct. 2019].

files, d. (2019). *Tablespaces, Datafiles, and Control Files*. [online] Docs.oracle.com. Available at: https://docs.oracle.com/cd/B19306_01/server.102/b14220/physical.htm#i2004 [Accessed 9 Oct. 2019].

Lab3: Managing database instance

Question 1: Earlier in this lab (step 2c), you've changed the `JOB_QUEUE_PROCESSES` parameter to 50. Explain why the value was 20 in step 6.

Answer:

This is because after changing the value, I did not commit the changes. It does show the change when you do without committing but when you restart the database then changes have not made and still 20 as before it was stored temporary. After using commit function, the value have changed to 50.

Question 2: Is the parameter `OPEN_CURSORS` static or dynamic? What does it control? Explain the difference between dynamic and static initialisation parameters.

Answer:

Open Cursors parameter controls how many tasks should run to avoid consuming too much library cache RAM. Sessions in the database can execute many SQL statements and the open cursor parameter manage the total number of open cursors for each session. Depends the amount we set, oracle will allocate up to the same amount in the library cache. The starting value for open cursors is usually set by the oracle at creation time similar to sessions and processes parameter. One risk we can face, if we set open cursor too high then we can risk having a task short with ORA-01000 error. (Burleson, 2019). The difference between both is that dynamic initialization parameters can get changed for current oracle database instances and it changes immediately.

Static cannot change for current instance while its running, change the parameter in the text file or server parameter file and restart the database then it will change.

References

Burleson, D. (2019). open_cursors Tips. [online] Dba-oracle.com. Available at: http://www.dba-oracle.com/t_open_cursors.htm [Accessed 14 Oct. 2019].

Lab 4: Managing database security

Question 1: What are the benefits and drawbacks of using roles? Provide an example.

Answer:

Benefits

- Preventing data from being public by giving limited access to data when its needed, which makes it easier to manage security.
- Being able to change the privileges for users within single grant or revoke.
- Managing the privileges for database application such as application roles.
- Managing the privileges for a user group such as user roles.
-

Drawbacks

The biggest disadvantages would be giving a user more privileges than a user actually need. This can lead to a security issue if the user abused the extra privileges. For example, giving a business user a role as Resource, which mean that is able to create, modify and delete certain types of schema objects related to the user. The business user can delete the company in mistake or use the data against the company and so on.

Question 2: If you created a user in SQL*Plus using following command:

```
CREATE USER user1 IDENTIFIED BY passw1;
```

what default and temporary tablespaces will be assigned to the user in this case? Explain.

Answer:

In the SQL statement above the temporary and default table space have not being set at the time of creating an execution so the user will be assigned automatically to the default temporary tablespace specified at database creation. If there is no default temporary tablespace assigned, then it will use the system table space or another permanent default table space that defined by the database administrator. The default and temporary table space in the statement above is set to use system table space created at creation or permanent space created by system administrator. It's recommended to change the default table space from SYSTEM to other tablespace for example in here to users. Storing user's data in default permanent table and not in the SYSTEM table space reduce the issue of having the entire database to become nonfunctional.

Question 3: You have been asked to implement the following security requirement: the user should be allowed 10 min of idle time, only 1 opened session and is required to change the password after their first login and every 3 months. Write SQL commands to implement this requirement and explain.

Answer:

CREATE PROFILE NEW_PROFILE

LIMIT IDLE_TIME 10

PASSWORD_LIFE_TIME 90

SESSION_PER_USER 1;

CREATE USER TEMOR identified by oracle

PASSWORD EXPIRE

PROFILE NEW_PROFILE;

Lab 5: Managing tablespaces and other storage structures

Question 1: How many extents are occupied by the table EMPLOYEES? Explain and make a screenshot.

Answer: In the screen shot below shows all the extents are 1MB for all the table spaces including Employees which mean this the default size that oracle have set up to ensure each extent contains at least five database blocks. The default size was not specified for this then default size was 1MB. (Docs.oracle.com, 2019)

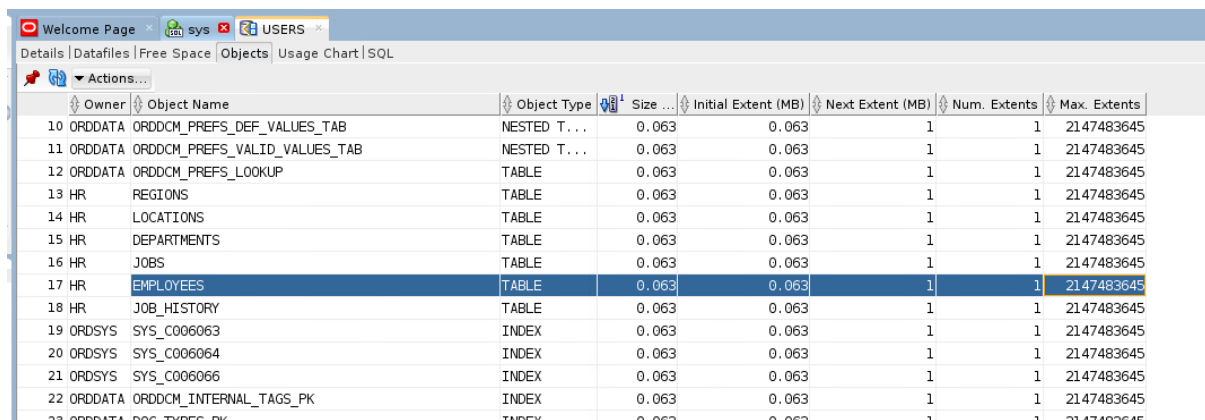
Definition

Extent is a logical unit of database storage space that made up of several contiguous data blocks. Extents make up a segment and when segment place all used then oracle allocates new extents for the segment. This is created during the creation of a table that oracle set up to the data segment an initial extent of specified number of blocks. The initial extent is created and reserved for the table rows. When the initial extent space full then oracle will allocate an incremental extent for that segment. This new extent is a subsequent extent of the same or greater size that the initial extent that was allocated at the start (Docs.oracle.com, 2019).

Oracle allocate extents by using different algorithms to give the right number of extents to a table space and that depends if they are managed locally or dictionary managed. Oracle looks for free space to allocate to a new extent when using locally managed table(Docs.oracle.com, 2019)

Screen-shot (if required):

The screenshot below shows that the number of extents allocated to Employees table is 1MB.



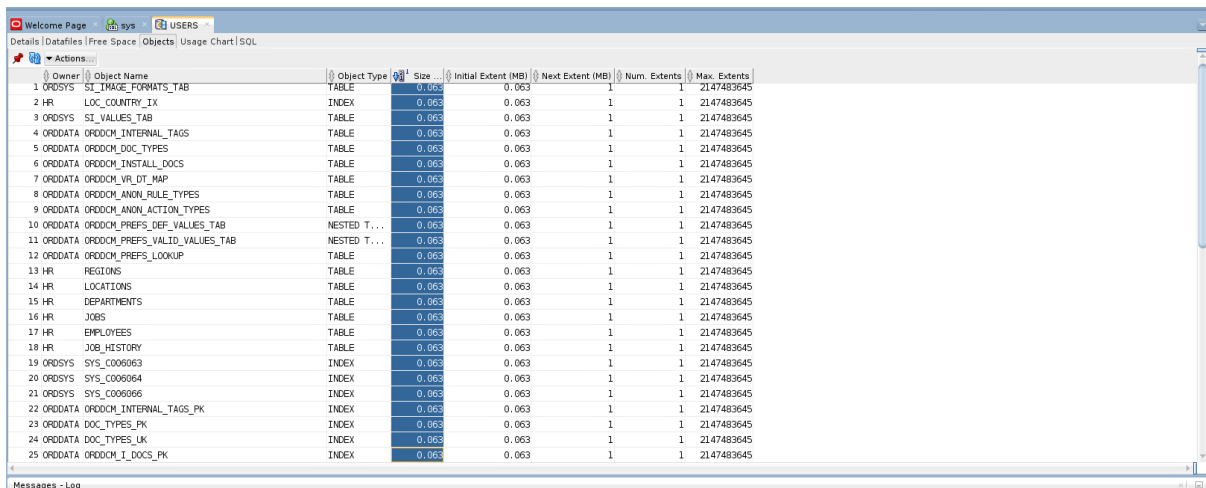
Owner	Object Name	Object Type	Size ...	Initial Extent (MB)	Next Extent (MB)	Num. Extents	Max. Extents
ORDDATA	ORDDCM_PREFS_DEF_VALUES_TAB	NESTED T...	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_PREFS_VALID_VALUES_TAB	NESTED T...	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_PREFS_LOOKUP	TABLE	0.063	0.063	1	1	2147483645
HR	REGIONS	TABLE	0.063	0.063	1	1	2147483645
HR	LOCATIONS	TABLE	0.063	0.063	1	1	2147483645
HR	DEPARTMENTS	TABLE	0.063	0.063	1	1	2147483645
HR	JOBS	TABLE	0.063	0.063	1	1	2147483645
HR	EMPLOYEES	TABLE	0.063	0.063	1	1	2147483645
HR	JOB_HISTORY	TABLE	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006063	INDEX	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006064	INDEX	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006066	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_INTERNAL_TAGS_PK	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_INTERNAL_TAGS_PK	INDEX	0.063	0.063	1	1	2147483645

Figure 5: Number of extent created

Question 2: What is the name of the object of an INDEX type in the USERS tablespace that takes up the most space (is the largest)? Explain and provide a screenshot.

Answer:

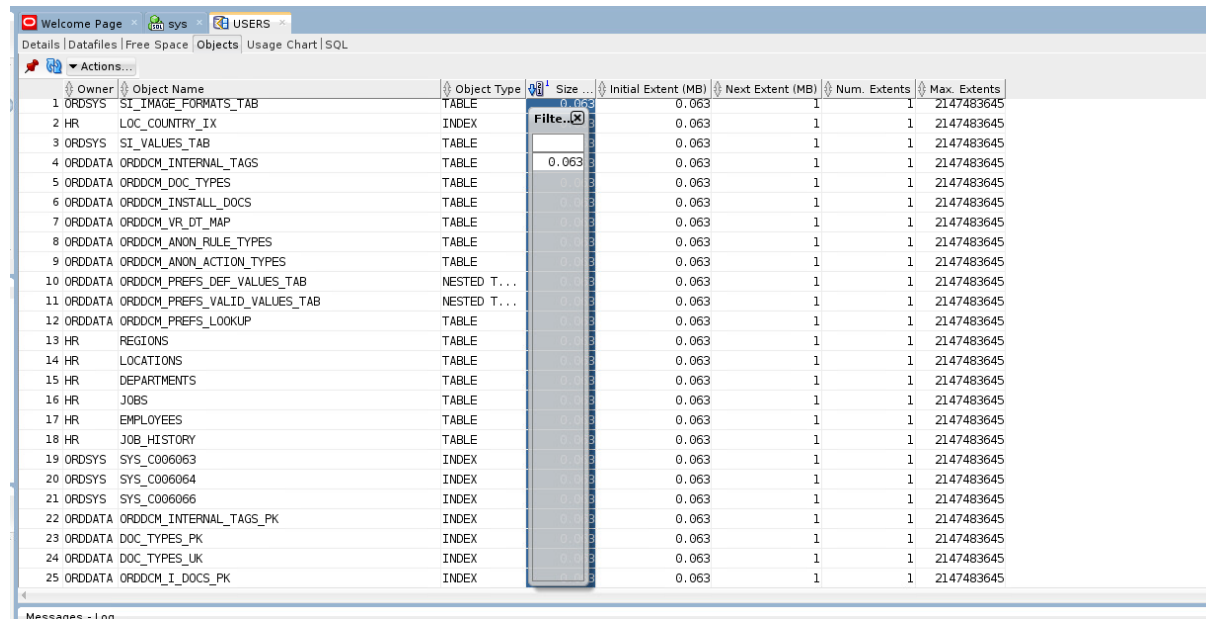
Table space within index segment is either the default table space of the owner or created using the Create statement which in our case we created the user's tablespace. All the objects within an index type have the same size in the users table space which is 0.063 as default size of each objects. (Docs.oracle.com, 2019)



Owner	Object Name	Object Type	Size	Initial Extent (MB)	Next Extent (MB)	Num. Extents	Max. Extents
ORDSYS	SI_IMAGE_FORMATS_TAB	TABLE	0.063	0.063	1	1	2147483645
HR	LOC_COUNTRY_IX	INDEX	0.063	0.063	1	1	2147483645
ORDSYS	SI_VALUES_TAB	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_INTERNAL_TAGS	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_DOC_TYPES	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_INSTALL_DOCS	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_VR_DT_MAP	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_ANON_RULE_TYPES	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_ANON_ACTION_TYPES	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_PREFS_DEF_VALUES_TAB	NESTED T...	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_PREFS_VALID_VALUES_TAB	NESTED T...	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_PREFS_LOOKUP	TABLE	0.063	0.063	1	1	2147483645
HR	REGIONS	TABLE	0.063	0.063	1	1	2147483645
HR	LOCATIONS	TABLE	0.063	0.063	1	1	2147483645
HR	DEPARTMENTS	TABLE	0.063	0.063	1	1	2147483645
HR	JOBS	TABLE	0.063	0.063	1	1	2147483645
HR	EMPLOYEES	TABLE	0.063	0.063	1	1	2147483645
HR	JOB_HISTORY	TABLE	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006063	INDEX	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006064	INDEX	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006066	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_INTERNAL_TAGS_PK	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	DOC_TYPES_PK	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	DOC_TYPES_UK	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_I_DOCS_PK	INDEX	0.063	0.063	1	1	2147483645

Figure 6: Tablespace taking most space

Using the filter provided by SQL Developer also shows there is one common size between all the files.



Owner	Object Name	Object Type	Size	Initial Extent (MB)	Next Extent (MB)	Num. Extents	Max. Extents
ORDSYS	SI_IMAGE_FORMATS_TAB	TABLE	0.063	0.063	1	1	2147483645
HR	LOC_COUNTRY_IX	INDEX	0.063	0.063	1	1	2147483645
ORDSYS	SI_VALUES_TAB	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_INTERNAL_TAGS	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_DOC_TYPES	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_INSTALL_DOCS	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_VR_DT_MAP	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_ANON_RULE_TYPES	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_ANON_ACTION_TYPES	TABLE	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_PREFS_DEF_VALUES_TAB	NESTED T...	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_PREFS_VALID_VALUES_TAB	NESTED T...	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_PREFS_LOOKUP	TABLE	0.063	0.063	1	1	2147483645
HR	REGIONS	TABLE	0.063	0.063	1	1	2147483645
HR	LOCATIONS	TABLE	0.063	0.063	1	1	2147483645
HR	DEPARTMENTS	TABLE	0.063	0.063	1	1	2147483645
HR	JOBS	TABLE	0.063	0.063	1	1	2147483645
HR	EMPLOYEES	TABLE	0.063	0.063	1	1	2147483645
HR	JOB_HISTORY	TABLE	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006063	INDEX	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006064	INDEX	0.063	0.063	1	1	2147483645
ORDSYS	SYS_C006066	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_INTERNAL_TAGS_PK	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	DOC_TYPES_PK	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	DOC_TYPES_UK	INDEX	0.063	0.063	1	1	2147483645
ORDDATA	ORDDCM_I_DOCS_PK	INDEX	0.063	0.063	1	1	2147483645

Figure 7: SQL developer filter

Question 3: : Write `CREATE TABLESPACE` command that implements the requirements for STOCK tablespace and **explain how the values that you provided for the various parameters of this command implement each of the requirement.**

- `CREATE TABLESPACE STOCK`

We are using a data definition language command called “CREATE” then we are specifying it's a tablespace then giving a name to the tablespace.

- Datafile

Tablespaces contain one or more physical datafiles. A Datafile can be related to only one table space and only one database. In this create statement for the stock we are connecting the Stock tablespace with a datafile that is already created called stock01.dbf. (Docs.oracle.com, 2019)

Datafile ``/u01/app/oracle/oradata/stock01.dbf``

- **SIZE 5Mb**

This requirement specifies the amount of data can be stored in the specific tablespace.

- Auto extend

This requirement it enables tablespace to extend itself in terms of size when it reaches the size limit. However, in our case we are turning this feature off, which in this case we get alerted when the database table reaches the size limit. We can respond then to the alert by increasing the tablespace size. This can be done by either increasing the size of tablespace data file or adding a new data files to the tablespace . (Docs.oracle.com, 2019)

- extent management local

This requirement above it enables the table space to be managed locally by specifying local at the end of the statement. The requirement above misses the two clauses either `AUTOALLOCATE` clause or `UNIFORM` clause. Making the database to manage extents automatically then choosing the `AUTOALLOCATE` would be the perfect choice and this the default clause if in the create statement not mentioned. In our case we have been required to set up the extend management to be local. In oracle website mentioned that if you are expecting tablespaces to have different sizes then `autoallocate` would be the perfect option and if you are not worried about having a control over space allocation and deallocation as it this simplifies table space management. If you want control over space used, then using `uniform` clause would be perfect choice as it makes sure that there will be never unusable space in tablespace. (Docs.oracle.com, 2019)

- segment space management auto

This clause only can be set when the tablespace have been set to be available in locally managed tablespaces. This simpler and more efficient for managing space within a segment. This avoid us of having to specify and tune the freelists, freelist groups storage parameters for schema objects created in the table space. (Burleson, 2019)

Answer:

CREATE TABLESPACE STOCK

Datafile `/u01/app/oracle/oradata/stock01.dbf`

Size 5m

Autoextend off

extent management local

segment space management auto;

Screen-shot (if required): We have asked to change the size later on the lab task from 5mb to 71mb.

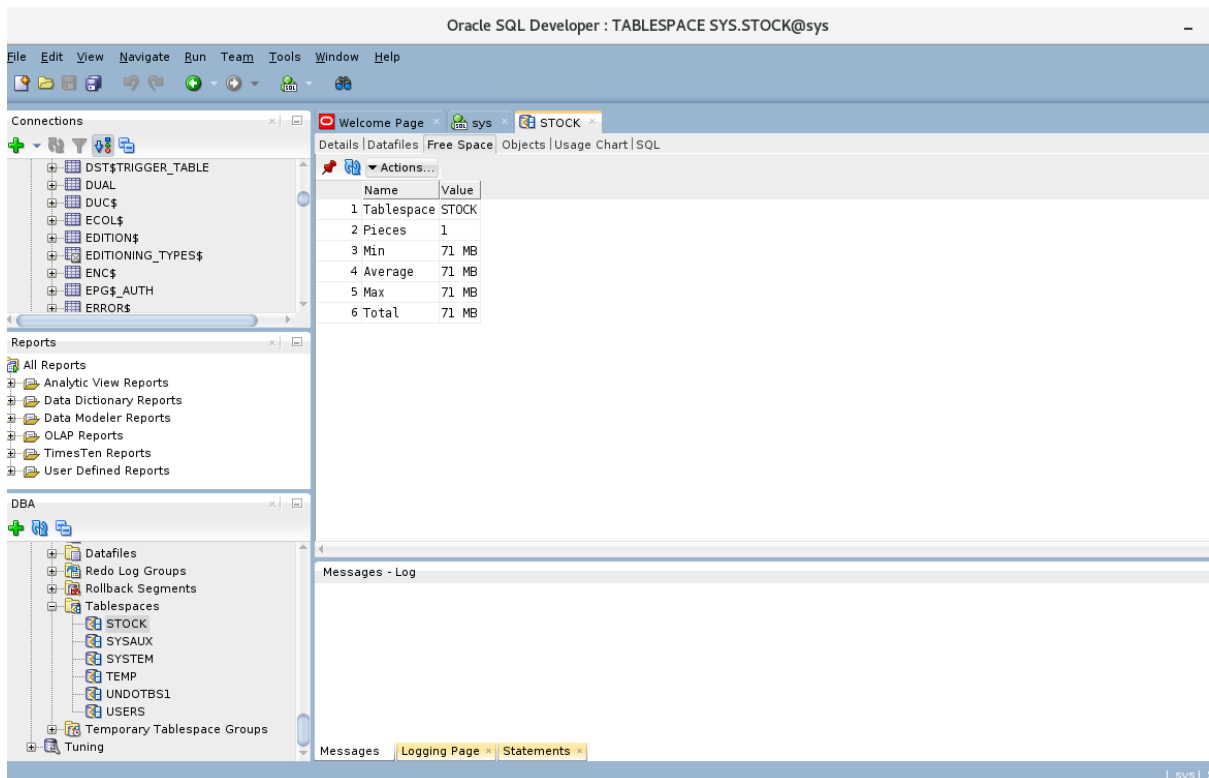


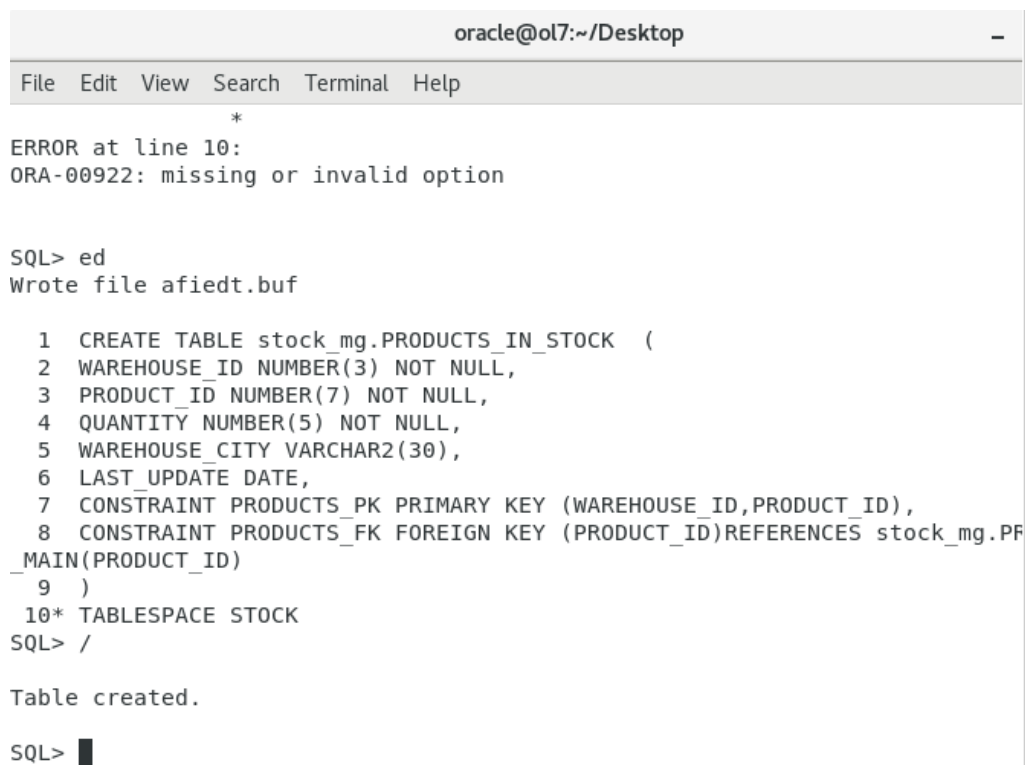
Figure 8: Tablespace size

Lab6: Managing schema objects and data

Question 1: What SQL statement you should execute in order to create PRODUCTS_IN_STOCK table in SQL*Plus? Make sure all constraints have user-defined names. Write SQL statement and execute it.

Answer:

```
CREATE TABLE stock_mg.PRODUCTS_IN_STOCK (  
  WAREHOUSE_ID NUMBER(3) NOT NULL,  
  PRODUCT_ID NUMBER(7) NOT NULL,  
  QUANTITY NUMBER(5) NOT NULL,  
  WAREHOUSE_CITY VARCHAR2(30),  
  LAST_UPDATE DATE,  
  CONSTRAINT PRODUCTS_PK PRIMARY KEY (WAREHOUSE_ID,PRODUCT_ID),  
  CONSTRAINT PRODUCTS_FK FOREIGN KEY (PRODUCT_ID)REFERENCES  
  stock_mg.PRODUCTS_MAIN(PRODUCT_ID)  
)  
TABLESPACE STOCK;
```



```
oracle@ol7:~/Desktop
File Edit View Search Terminal Help
*
ERROR at line 10:
ORA-00922: missing or invalid option

SQL> ed
Wrote file afiedt.buf

 1 CREATE TABLE stock_mg.PRODUCTS_IN_STOCK (
 2 WAREHOUSE_ID NUMBER(3) NOT NULL,
 3 PRODUCT_ID NUMBER(7) NOT NULL,
 4 QUANTITY NUMBER(5) NOT NULL,
 5 WAREHOUSE_CITY VARCHAR2(30),
 6 LAST_UPDATE DATE,
 7 CONSTRAINT PRODUCTS_PK PRIMARY KEY (WAREHOUSE_ID,PRODUCT_ID),
 8 CONSTRAINT PRODUCTS_FK FOREIGN KEY (PRODUCT_ID)REFERENCES stock_mg.PF
 9 _MAIN(PRODUCT_ID)
10 )
11* TABLESPACE STOCK
SQL> /

Table created.

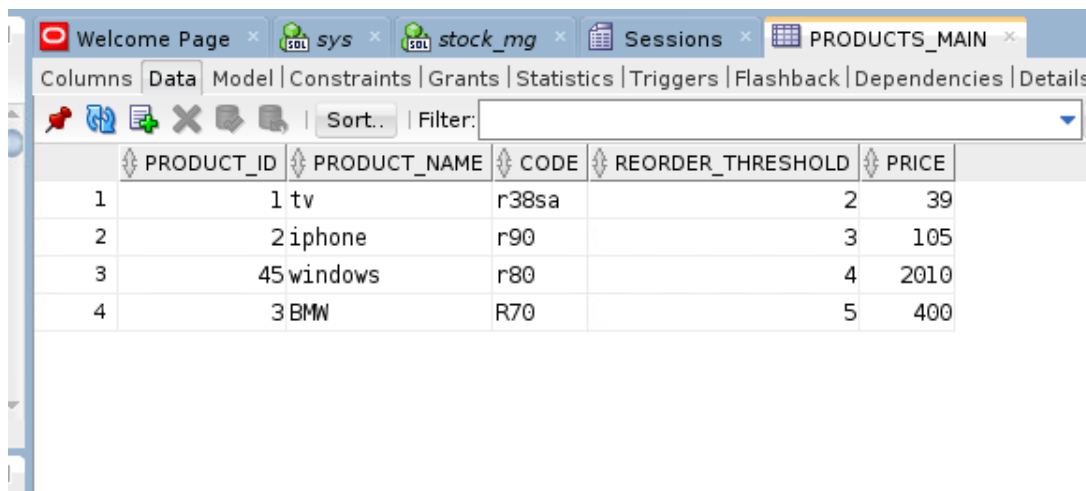
SQL> █
```

Figure 9: Required table created

I have created a table inside stock_mg schema by basically identifying the schema name before the table name separated by dot. This privilege only works as DBA creating tables for users. The specification was all inserted inside the bracket which all conclude of the rows names with specification of data type and constraints. Both primary keys constraint were set in the statement to be not null but by default they are not null set when they are set to be a primary key. Note in the creation of foreign key referenced products_main table with field product_ID. At the end of the statement we referenced the table to a table space name STOCK.

Question 2: : Explain how you can test the use of the newly created index. Write SQL statements if necessary. **Answer:**

In order to test the index then some data needs to be inserted into the table,



The screenshot shows a database management tool interface with a tab for 'PRODUCTS_MAIN'. The 'Data' tab is selected, displaying a table with the following columns: PRODUCT_ID, PRODUCT_NAME, CODE, REORDER_THRESHOLD, and PRICE. The table contains four rows of data:

	PRODUCT_ID	PRODUCT_NAME	CODE	REORDER_THRESHOLD	PRICE
1	1	tv	r38sa	2	39
2	2	iphone	r90	3	105
3	45	windows	r80	4	2010
4	3	BMW	R70	5	400

Figure 10: Inserting data into table

In order to test the newly created index will be using the “EXPLAIN PLAN tool” to inform us the exeution plan for the requested SELECT, UPDATE,DELETE and INSERT statements in bracket “query performance”. The explain plan conculde table refrenced in the statament, ordering method with tables that has been accessed and data operations such as filiters or sort included in the statement. The plan also contain information about optimiaztion, parallel excuation and partitioning. More importantly, for this question explain plan will show the table access methods either full or Index, table join types and sorting.

Firstly, before creating the index, I wanted to check the peforamnce by running a select query that compares information against the product_name column with “Explain plan for” as we need to create the explain plann table first.

EXPLAIN PLAN FOR

SELECT * FROM PRODUCTS_MAIN WHERE PRODUCT_NAME='tv';

After submitting this query then I used this query to view the plan table

SELECT * FROM TABLE (dbms_xplan.display());

Those statement will show what are the execution performance without an index created for the table. See screenshot below.

```

1 explain plan for
2* select * from PRODUCTS_MAIN WHERE PRODUCT_NAME = 'tv'
SQL> /

Explained.

SQL> SELECT * FROM table(dbms_xplan.display());

PLAN_TABLE_OUTPUT
-----
Plan hash value: 4137556008

-----
---
| Id | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time
|
-----
---

```

```

PLAN_TABLE_OUTPUT
-----
| 0 | SELECT STATEMENT |              | 1 | 21 | 3 (0)| 00:00:0
1 |
|* 1 | TABLE ACCESS FULL| PRODUCTS_MAIN | 1 | 21 | 3 (0)| 00:00:0
1 |
-----
---

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT
-----
-----

```

Figure 11: Using explain plan for

In the plan table output it shows that the table access was “full” meaning it checked all the data in table to find the row requested. This means the table are not using index to find the requested data. The cost for CPU% shows 3 and it explained in the oracle that the higher cost, the lower performance we get.

After I created the index using the SQL developer tool or you can use SQL query like this .

```
CREATE INDEX PROD-NAME_IDX ON PRODUCTS_MAIN(PRODUCT_NAME);
```

Notice the type of index not mentioned in the query because the default index chosen by oracle is B-tree index. The reason I decided to keep the default index B-tree as Product-Name column can have many different products or can say its unique not like Bit-map used when there is repeated or duplicated data such as male or female.

I checked again the explain plan table to check on the performance within the index. First creating the plan table then re-viewing the table to check if there is any changes of the performance. See screenshot below.

```
SQL> Explain plan for
      2 select * from PRODUCTS_MAIN WHERE PRODUCT_NAME ='tv';
```

Explained.

```
SQL> select * from table(dbms_xplan.display());
```

PLAN_TABLE_OUTPUT

Plan hash value: 478802930

Id	Operation	Name	Rows	Bytes	Cost
----	-----------	------	------	-------	------

Index created is working

PLAN_TABLE_OUTPUT

0	SELECT STATEMENT		1	21	
2	(0)				
1	TABLE ACCESS BY INDEX ROWID BATCHED	PRODUCTS_MAIN	1	21	
2	(0)				
* 2	INDEX RANGE SCAN	PROD_NAME_IDX	1		
1	(0)				

PLAN_TABLE_OUTPUT

Figure 12: Prove of index created

After checking the plan table then I found the cost of CPU not 2 instead of 3 when it was full scan. The screenshot showed the index that was created is working perfectly and as Oracle website the lower cost of CPU is then it will deliver a better performance.

Question 3: How would you resolve the conflict if you don't have access to EM or SQL Developer, and only have SQL*Plus. Explain and include all necessary SQL commands.

Answer:

The first commands to use in SQL plus would be using V\$SESSION command that would list session information for each current session according to oracle documentation. The reason wanted to list session because I do not know what the serial number or SID for each session or its are not knowing to me who active user or inactive user in the server is. To use V\$SESSION command I need to have a DBA privilege, so this only works on a sys user account. There are a lot of information V\$SESSION command can display for us but for me as DBA to be able to kill a session or check status then will only select view information such as SID,

SERIAL#, USERNAME, STATUS.

SID – a session identifier

Serial – session serial number

Username – username that the session allocated to.

STATUS – active or inactive session

To kill inactive session must need SID and Serial number like this

```
ALTER SYSTEM KILL SESSION '76,45009';
```

Screen-shot (if required):

Firstly, I had to know what sessions are actives and note users sid and serial numbers for their sessions but more importantly the status. Below screenshots shows the operations.

```
SQL> ED
Wrote file afiedt.buf
```

```
1* SELECT SID,SERIAL#,USERNAME,STATUS FROM V$SESSION
SQL> /
```

SID	SERIAL#	USERNAME	STATUS
2	60042		ACTIVE
3	8119		ACTIVE

Figure 13: Finding out status and serial id

I identified user stock_mg has inactive session so I noted SID and serial number from screenshot blow.

```

-----
USERNAME
-----
STATUS
-----

ACTIVE

          76          45009
STOCK_MG
INACTIVE

```

Figure 14: Checking active sessions and taking SID and serial key to kill sessions

After making a note of stock_mg user SID and serial number then its time to put this into the query to kill the session as its inactive session. Below screenshot showing the query.

```

1* ALTER SYSTEM KILL SESSION '76,45009'
SQL> SELECT sid,serial#,username,status from V$SESSION;

          SID          SERIAL#
-----
USERNAME
-----
STATUS
-----

          2          60042

```

Figure 15: Query to kill the session

After running the query, I want back to run the same query for checking active sessions and now the status changed to inactive sessions.

```

-----
STATUS
-----

ACTIVE

          76          45009
STOCK_MG
KILLED

```

Figure 16: Session killed

Lab 7: Using flashback technology

Question 1: How many objects were dropped together with HR. DEPARTMENT table? Describe them (include the name and the type) and explain why they were dropped together with the table?

Answer:

When I used the statement “drop table hr. Department cascade constraints;”, the table has been dropped by oracle. Specifying at the end of the query cascade constraints to drop all the referential integrity constraints that exist within the table such as primary and all the unique keys to be dropped within the table. If our table primary unique referenced as the foreign key in another table then without using cascade constraints at the end of the query, then Oracle will return this error “ORA-02449: unique/primary keys in table referenced by foreign keys”. The two objects that dropped within the table was the constraints such as primary key and the indexes because we specified them in calling cascade constraints. **Screenshot (if required):**

The screenshot shows I queried the database to understand what has been dropped by the query statement drop table. The bin showed three objects have been dropped such as primary key, index, and department table. It also specifies timing when it has been dropped.

```
SQL> select owner,object_name,original_name,droptime from dba_recyclebin;

OWNER
-----
OBJECT_NAME
-----
ORIGINAL_NAME
-----
DROPTIME
-----
HR
BIN$lsPxxylvE6HgVQAAAAAAQ==$0
DEPT_LOCATION_IX
2019-11-07:15:01:47

OWNER
-----
OBJECT_NAME
-----
ORIGINAL_NAME
-----
DROPTIME
-----
HR
BIN$lsPxxylvE6HgVQAAAAAAQ==$0
DEPT_ID_PK
2019-11-07:15:01:48

OWNER
-----
OBJECT_NAME
-----
ORIGINAL_NAME
-----
DROPTIME
-----
HR
BIN$lsPxxylwE6HgVQAAAAAAQ==$0
DEPARTMENTS
2019-11-07:15:01:48
```

Figure 17: Dropped objects



The bin shows the two objects dropped within the table was the constraint primary key dept_id_pk and the index dept_location_IX.

Figure 18: Bin with objects inside

Question 2:

- a) Provide a screenshot of the SQL command and the result of the flashback version query in step 7.3(1b) and explain.

Answer: In this question above, we have been asked to use the flashback version query. This type of query used to retrieve metadata and historical data for a specific time interval such as viewing all the rows of the table that ever existed during a given time interval. We will be querying the start as well and the end time because metadata include that as well.

I updated a value twice inside stock_mg. bouns table then I used the flash version query to show the history of the changes. The version row only exists if I used “commit “ after the query. I had to query the scn number before using the update query to be able to select the time that wants to check the changes between that will give me the absolute value that was there and the changes after.

In the query in the screenshot below, I had some select operation to get a specific result from the table sock_mg.bouns versions.

Versions_starttime = This command shows when the row version was created by given timestamp or SCM;

Versions_ENDTIME = This command shows when the row was expired.

Versions_xid = This command used as an identifier of the transaction that created the row version.

Versions_ operation = This command used to identify operation performed by the transaction such as insertion, deletion and for the update. This shows like the row after the insert operation and the row before delete operation or the row affected by the operation.

```
SQL> SELECT versions_starttime,versions_endtime,versions_xid,versions_operation
       2 ,bonus_amt from stock_mg.bonus versions between scn 4649588 and 4649792 where name ='Ford';
```

```
VERSIONS_STARTTIME
-----
VERSIONS_ENDTIME
-----
VERSIONS_XID      V  BONUS_AMT
-----
08-NOV-19 14.41.29

08000400370B0000 U          2300

08-NOV-19 14.40.49
08-NOV-19 14.41.29
07001A00410B0000 U          2100

VERSIONS_STARTTIME
-----
VERSIONS_ENDTIME
-----
VERSIONS_XID      V  BONUS_AMT
-----

08-NOV-19 14.38.58
08-NOV-19 14.40.49
03001100450B0000 I          3000

08-NOV-19 14.38.58
```

Figure 19: Executing Flashback version query

Benefits

- This query shows us the changes made to the database within the timestamp. This helps us to identify what type of version needs to be flashbacked using SCN start time that included in the flashback version query. In the lab before we were querying to get SCN before making the changes and afterward to be able to flashback. Using this flash version query show us all the information we need to be able to flashback to what information we need to get back. In the query above I have not selected version_startscn because I have recorded both before and after scan that helped me to get the result. In the future, if we do not have scn or timestamp query then would use this query to see the changes and be able to select the timestamp or scan of the version we would want back then run the query flashback table.
- The query can be used for auditing and analytical purposes.

Drawbacks

- Firstly, after restarting the database then this query would not work which in that case would not be able to get SCN number or to be able to analysis the changes made.
- Flashback version query only used to view past and current data and not undo anything. The information however from the query can be used such as timestamp or scan to retrieve data.
- You cannot specify versions in flashback queries for external, fixed and temporary tables or tables that are part of the cluster.

Question 3: What pre-requisites must be set before you can flashback the database to point in time? Explain and provide the necessary SQL commands.

Answer:

- You must be a system administrator (sysdba).
- The fast recovery area must be prepared for the database.
- The database must be put in the flashback mode using this query “Alter Database Flashback on”.
- The database must be mounted but no opened with the current control file.
- The database must run in ARCHIVELOG mode.
- The database must contain no online tablespace for which flashback functionality was disabled using the SQL statement “Alter tablespace flashback off.
- After ensuring all the points above have been met such as preparing a fast recovery area or enable database flashback mode and opened the database then you can use this command,

SHUTDOWN IMMEDIATE

STARTUP MOUNT

FLASHBACK DATABASE TO TIMESTAMP SYSDATE-1; This will flash the database back to one day.

Lab 8: Moving data between systems

Question 1:: Explain why you need to log in as the user `SYSTEM` and not `SYS` in order to move data and why you need to log in using `Normal` role and not `sysdba` role. **Discuss the principle that is implemented here and its benefits.**

Answer:

Sys with role `sysdba` doesn't have a privilege to move data between systems. Sys is a powerful administrative account designed for managing instances, users and so on. Moving data in this situation you have to log in with user `system` and role `normal`. This is good for businesses to separate concerns by having a user that performs data movements logged in as a system and an administrator such `sys` with role `sysdba` to do administrative tasks.

Question 2: From the logical storage point of view we can say that table `admin.employees` belong to the `admin` schema. What is the physical storage of the table `admin.employees`? Provide a screen-shot and explain.

Answer:

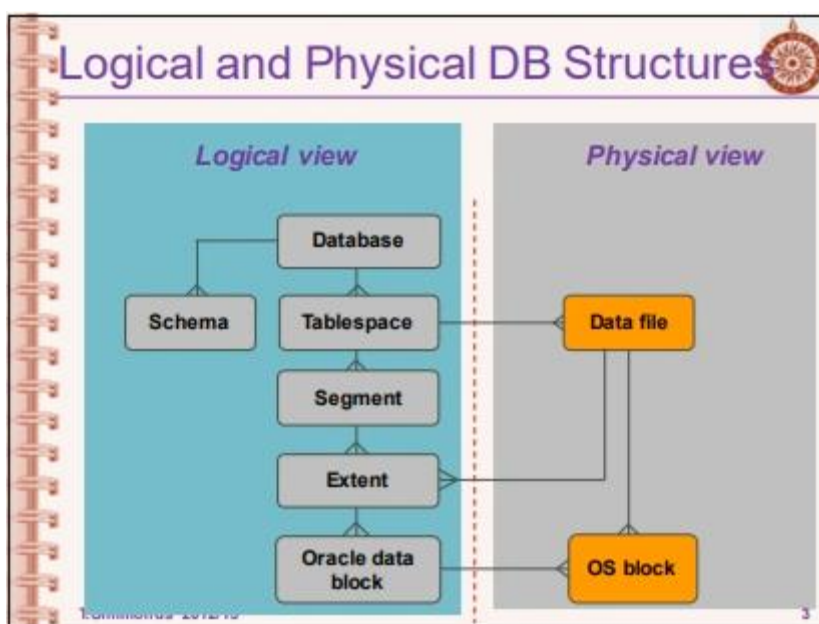


Figure 20: Logical VS physical

What is the physical storage?

The physical storage from the screenshot above is data files and operating system data blocks.

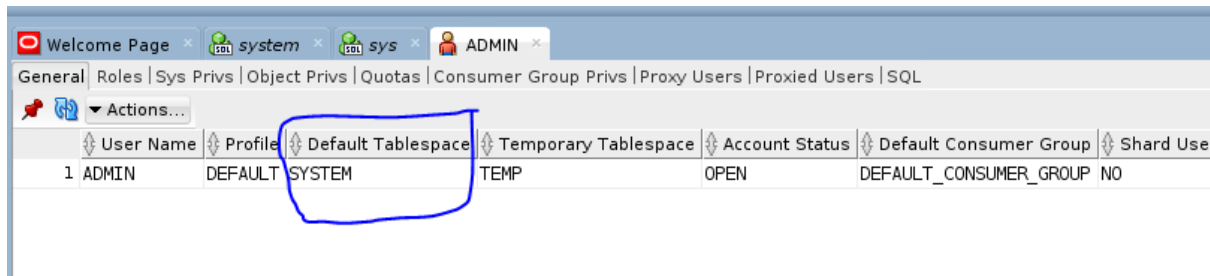
What is logical storage:

The screenshot above shows that the logical structure is tablespace, segment, extent, oracle data block, and schemas.

To answer the question, we must find where the schemas where is stored?

Oracle database stores the schemas logically within the tablespace of the database. The data of each object is physically inside one or more of the tablespace's datafiles.

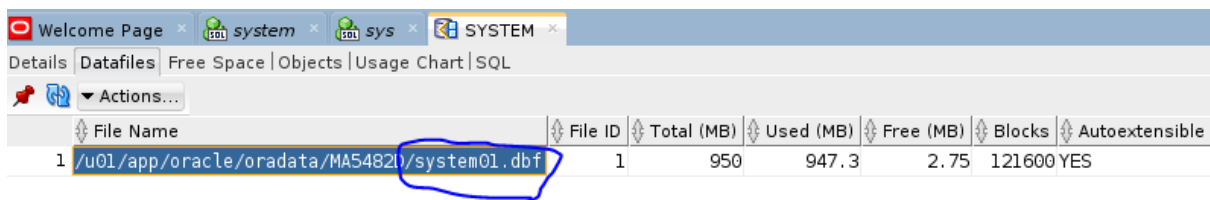
What I got from the explanation above that I need to find what tablespace allocated for admin. After finding out the tablespace then finding what tablespace datafile name is that admin. Employees stored inside it.



The screenshot shows the Oracle Enterprise Manager interface with the 'ADMIN' user selected. The 'General' tab is active, displaying a table of user details. A blue circle highlights the 'Default Tablespace' column, which contains the value 'SYSTEM' for the 'ADMIN' user.

	User Name	Profile	Default Tablespace	Temporary Tablespace	Account Status	Default Consumer Group	Shard Use
1	ADMIN	DEFAULT	SYSTEM	TEMP	OPEN	DEFAULT_CONSUMER_GROUP	NO

Figure 21: Tablespace allocated for admin



The screenshot shows the Oracle Enterprise Manager interface with the 'SYSTEM' tablespace selected. The 'Datafiles' tab is active, displaying a table of datafile details. A blue circle highlights the 'File Name' column, which contains the value '/u01/app/oracle/oradata/MA54821/system01.dbf' for the first datafile.

	File Name	File ID	Total (MB)	Used (MB)	Free (MB)	Blocks	Autoextensible
1	/u01/app/oracle/oradata/MA54821/system01.dbf	1	950	947.3	2.75	121600	YES

Figure 22: Datafile physical structure location where the tablespace stored and this means the physical storage is system01.dbf for table admin .employees.

Lab 9: Managing backup and recovery

Question 1: What prerequisites must be met to create a valid backup of a database without shutting it down? Explain.

Answer:

The prerequisite with or without RMAN, the database must be configured in archive log mode and few additional recoveries related database initialization parameters. The archive log files show all the history happened in the database and used to prevent the loss of data with the media failure. [Ensuring that](#) tablespaces in backup mode and back up the rest of files using the operating system copy command or use Oracle tool provided called RMAN that created for [backup](#) and recovery. RMAN technology uses the database control file to keep its catalogue or DBA set up a schema for each database in a sperate database for RMAN catalogues. DBAS need to understand the different types of backup such as full, incremental and differential backups that are set using RMAN scripts. The DBA must also put the database in archive mode and backup all the archived redo logs. Lastly, [it's](#) very important that the DBA must not forget to back up at the end of each backup the RMAN catalogue. The DBA can do the export backup of the RMAN catalogue schema. Backup with the database open and the archive off cannot be utilized for recovery.

Question 2: What are the differences between logical and physical backup?

Answer:

Backup in oracle usually either physical or logical. The physical back up its usually the backup of physical database files such as data files and control files and so on into a media service. However, logical backup is using SQL statements and consists of exporting the schema objects into a binary file. Both terms export and import in logical backup used to move data in and out of the Oracle schema. Export used to write data into binary system files and the export files created will be to store information about the schema objects such as tables and stored procedure. The import utility used to read the export files and restores them back into the existing database.

