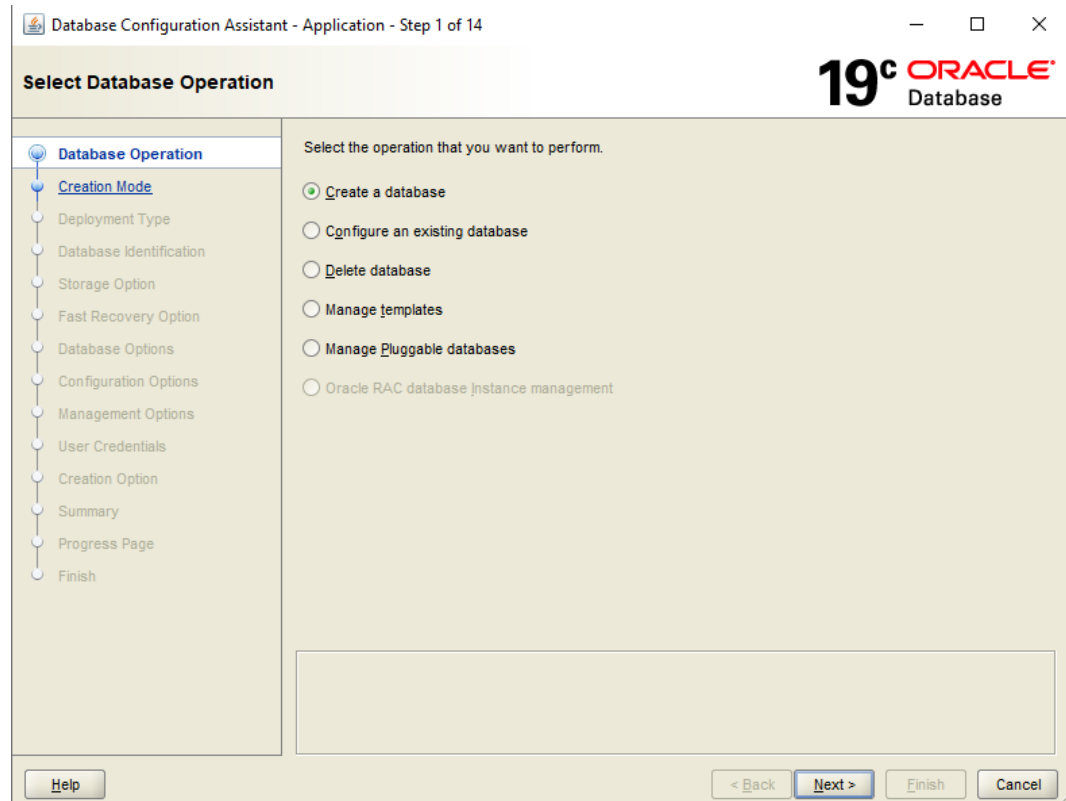


LAB 1

1. Create a Database using DBCA.

Steps Involved:

- Open cmd, type *dbca* then a screen will be opened.
- Follow the instructions in the screen as:



2. Create table customers with attributes first_name, last_name, phone_number and customer_id as a primary key.

Query:

```
create table customers (  
    customer_id int not null PRIMARY Key,  
    first_name varchar(25) not null,  
    last_name varchar(25) not null,  
    phone_number int not null  
);
```

3. Create table orders with attributes items, price, order_id as primary key and customer_id as a foreign key.

Query:

```
create table orders (  
    order_id int not null PRIMARY Key,  
    items varchar(25) not null,  
    price NUMBER(8,2) not null,  
    customer_id INT not NULL,  
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

4. Insert 5 rows in customer table and 5 rows in order table.

Query:

To insert data in customers table

```
insert into customers (customer_id, first_name, last_name, phone_number) values (1, 'Bijay', 'Shrestha',  
9876543210);
```

```
insert into customers (customer_id, first_name, last_name, phone_number) values (2, 'Dipak', 'Thapa  
Magar', 9860558458);
```

```
insert into customers (customer_id, first_name, last_name, phone_number) values (3, 'Ramesh',  
'Neupane', 9873333210);
```

```
insert into customers (customer_id, first_name, last_name, phone_number) values (4, 'Gaurav', 'Poudel',  
9876549990);
```

```
insert into customers (customer_id, first_name, last_name, phone_number) values (5, 'Sailesh', 'Karki',  
9876541470);
```

To insert data in orders table

```
insert into orders (order_id, items, price, customer_id) values (1, 'Apple', 120, 2);
```

```
insert into orders (order_id, items, price, customer_id) values (2, 'Banana', 220, 1);
```

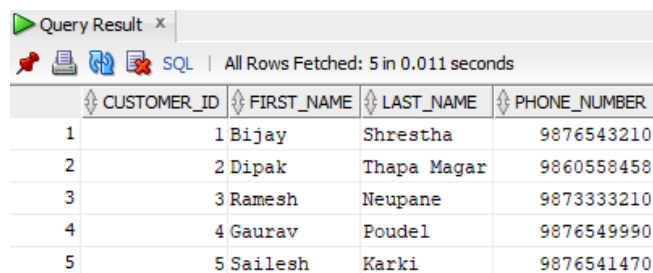
```
insert into orders (order_id, items, price, customer_id) values (3, 'Cream', 124, 3);
```

```
insert into orders (order_id, items, price, customer_id) values (4, 'Biscute', 20, 5);
```

```
insert into orders (order_id, items, price, customer_id) values (5, 'Keyboard', 420, 4);
```

To view customers table

```
select * from customers;
```

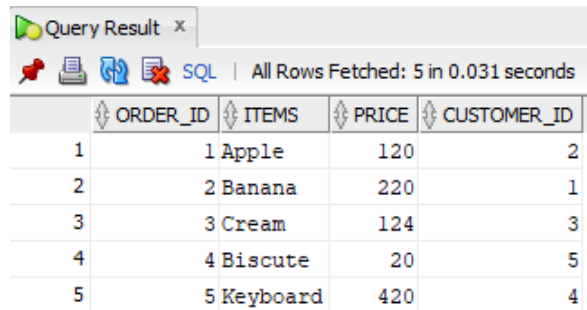


The screenshot shows a database query result window titled "Query Result". It displays the results of the query "select * from customers;". The window indicates that all rows were fetched in 0.011 seconds. The results are shown in a table with 5 rows and 4 columns: CUSTOMER_ID, FIRST_NAME, LAST_NAME, and PHONE_NUMBER.

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
1	1	Bijay	Shrestha	9876543210
2	2	Dipak	Thapa Magar	9860558458
3	3	Ramesh	Neupane	9873333210
4	4	Gaurav	Poudel	9876549990
5	5	Sailesh	Karki	9876541470

To view orders table

select * from orders;



Query Result x

SQL | All Rows Fetched: 5 in 0.031 seconds

ORDER_ID	ITEMS	PRICE	CUSTOMER_ID
1	1 Apple	120	2
2	2 Banana	220	1
3	3 Cream	124	3
4	4 Biscute	20	5
5	5 Keyboard	420	4

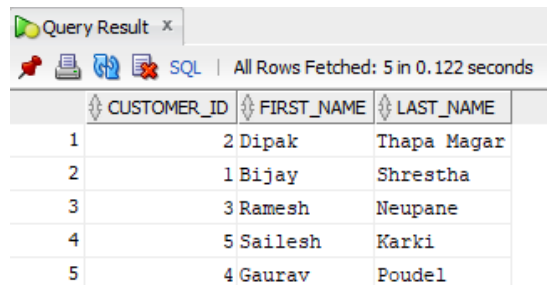
5. Create View to display the list of customers who have placed orders

Query:

```
CREATE VIEW customer_list AS  
SELECT c.customer_id,c.first_name, c.last_name  
FROM customers c  
INNER JOIN orders o  
ON c.customer_id = o.customer_id;
```

To view customer list table

select * from customer_list;



Query Result x

SQL | All Rows Fetched: 5 in 0.122 seconds

CUSTOMER_ID	FIRST_NAME	LAST_NAME
1	2 Dipak	Thapa Magar
2	1 Bijay	Shrestha
3	3 Ramesh	Neupane
4	5 Sailesh	Karki
5	4 Gaurav	Poudel

Conclusion and Discussion:

Hence, we created a database using DBCA and created various tables, inserted data in tables, became familiarized with primary key and foreign key concepts and were able to create views with inner join.

Lab 2

1. Create user identified by password

Query:

```
CREATE USER dipak IDENTIFIED BY dipak123;  
CREATE USER bijay IDENTIFIED BY bijay123;  
CREATE USER ramesh IDENTIFIED BY ramesh123;
```

2. Find the list of users with OPEN status

Query:

```
SELECT username from dba_users where account_status = "OPEN";
```

3. Grant create session to user

Query:

```
GRANT CREATE SESSION TO dipak;  
GRANT CREATE SESSION TO bijay;
```

4. Create user with password expire

Query:

```
CREATE USER ramesh IDENTIFIED By ramesh123 PASSWORD EXPIRE;
```

5. Grant SELECT privilege on Table to user

Query:

```
SELECT * FROM customers;  
GRANT SELECT ON customers TO dipak;  
GRANT SELECT ON customer_list TO dipak;  
SELECT * FROM customers;
```

6. Grant All privileges on table to user

Query:

```
GRANT ALL ON customers TO bijay;
```

7. Revoke Select privilege from table

Query:

```
REVOKE SELECT ON customers FROM dipak;
```

8. Revoke all privileges from table

Query:

```
REVOKE ALL ON customers FROM bijay;
```

9. Drop User

Query:

```
DROP USER ramesh;
```

Conclusion and Discussion:

Hence, we are able to create user along with password, select username from database users where status is OPEN, grant session to user, expire user password, grant SELECT and ALL operation on various tables, revoke given operation and even drop user.

Lab 3

1) Create roles student.

Query:

```
CREATE ROLES student;
```

Output:

```
Role STUDENT created.
```

2) Grant select, insert, update privileges on customers table to role student.

Query:

```
GRANT SELECT, INSERT, UPDATE ON customers TO student;
```

Output:

```
Grant succeeded.
```

3) Grant student role to user.

Query:

```
GRANT student TO ramesh;
```

Output:

```
Grant succeeded.
```

4) Set role.

Query:

```
SET ROLE student;
```

Output:

```
| Role STUDENT succeeded.
```

5) View role.

Query:

```
SELECT * FROM session_roles;
```

```
SELECT role, owner, table_name from role_tab_privs WHERE role='STUDENT';
```

Output:

	ROLE	OWNER	TABLE_NAME
1	STUDENT	SYSTEM	CUSTOMERS
2	STUDENT	SYSTEM	CUSTOMERS
3	STUDENT	SYSTEM	CUSTOMERS

Output:

	ROLE
1	STUDENT

6) Revoke student role from user.

Query:

```
REVOKE student FROM ramesh;
```

Output:

```
Revoke succeeded.
```

7) Drop role.

Query:

```
DROP ROLE student;
```

Output:

```
Role STUDENT dropped.
```

Conclusion and Discussion:

Hence, we are able to create roles, grant SELECT, INSERT, UPDATE privileges to that create role in various tables, grant role to user, set role, view role, revoke role from user and even drop role in this lab session.

Lab 4

1) Data Pump: Export and import table data and schema using Data Pump

Processes and query along with screenshots:

- At first, connection was created to database then we run below query:

Query:

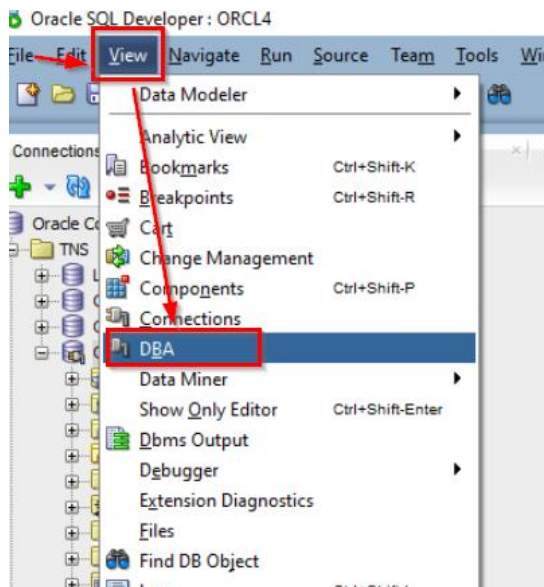
```
CREATE OR REPLACE DIRECTORY dba_backup AS  
'G:\BscCsit\sem7\DBA\DBABackup';
```

Output:

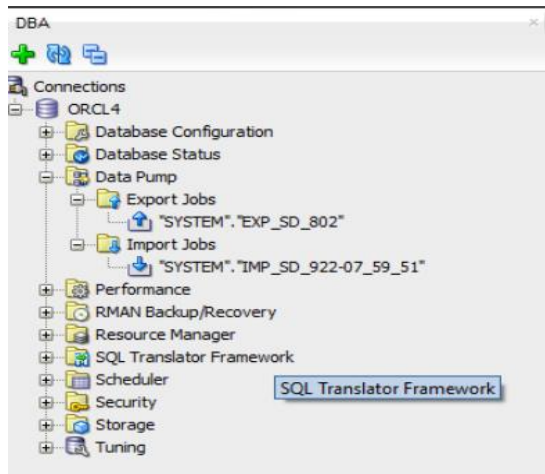
```
Directory DBA_BACKUP created.
```

To export:

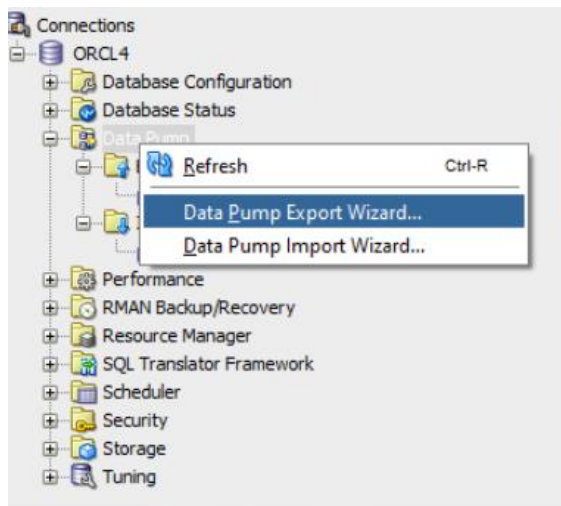
- We need to go to menu bar and press view then DBA as show in screenshot below:



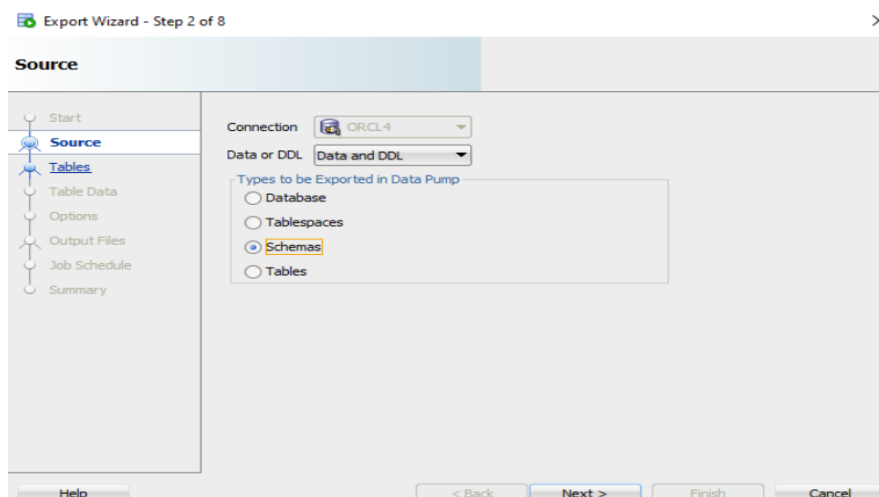
- We can see DBA on left end corner of screen and new Connection was selected and connection to our database was made and we can expand it where we can find Data Pump as in screenshot below:



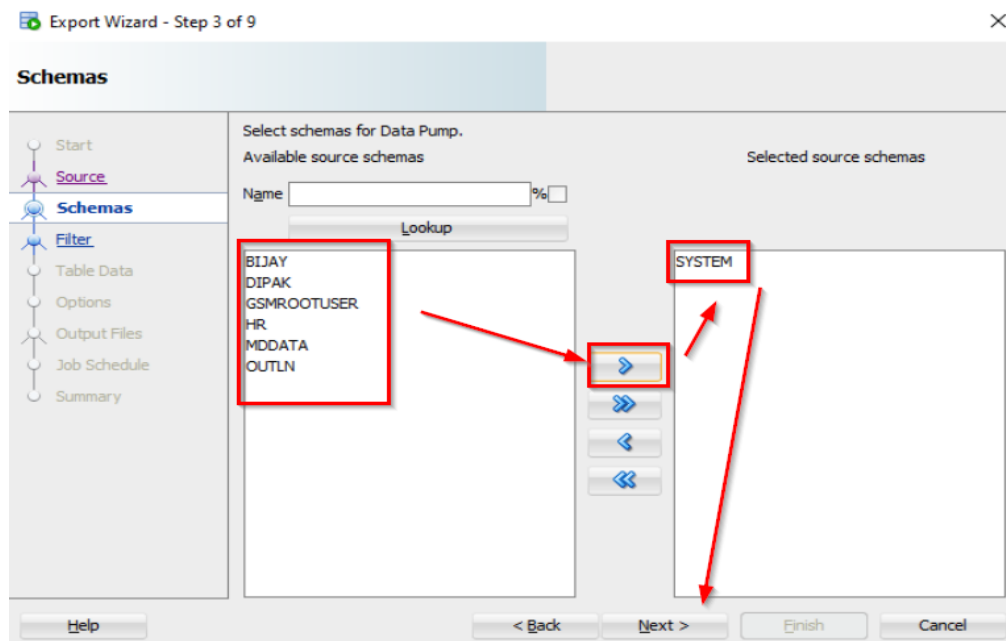
- Right click Data Pump folder gives us option as show in screenshot below:



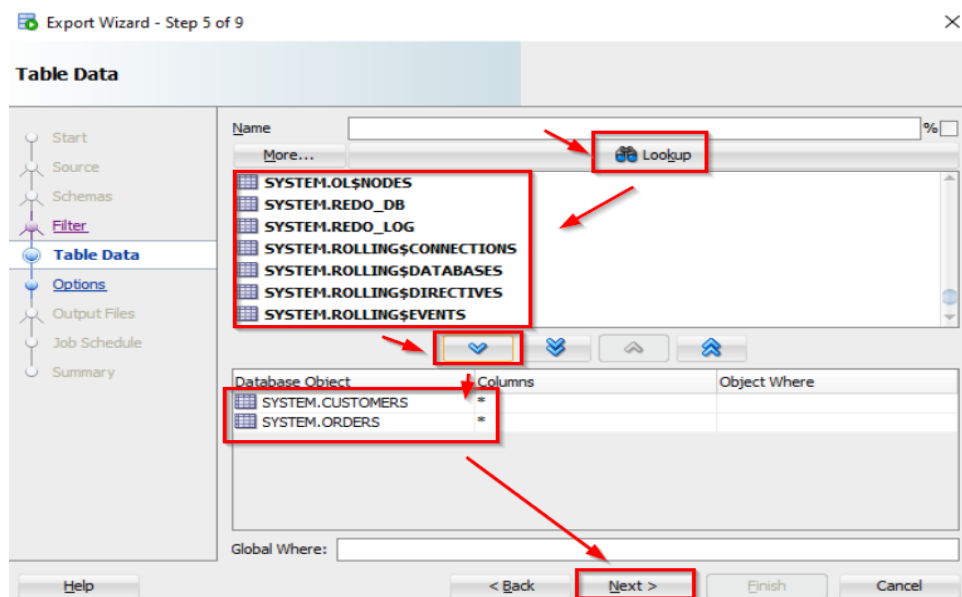
- To export data we need to select and click Data Pump Export Wizard. On click it we get below screen:



- To export schemas we select Schemas and to export tables we select Tables and rest of the step is same for all step. We can also set Data or DDL to prefer option, which in our case we set it to Data and DDL and click next.
- In next step, we click lookup to select schemas/tables and select the preferred one from Available source schemas/tables and send it to Selected source schemas/tables as below and click next:



- In next step, we lookup the table data and select the table, schemas and so on to export and click next



- In next step, we select the log file in which we are trying to export the data and the click next

Export Wizard - Step 6 of 9

Options

Thread Number: 1

Estimate: Blocks Calculate (Not Calculated yet)

☒ Enable Logging

Log File: DBA_BACKUP EXPDAT.LOG

☐ Export read-consistent view of data

By SCN ☒ By Date ☐

VERSION: COMPATIBLE

☐ Delete Master table

Help < Back Next > Finish Cancel

- In next step, we select the directories with same name as above step and click necessary file actions then click next

Export Wizard - Step 7 of 9

Output Files

Choose Output Files

Directories	File Names	Size (M)
DBA_BACKUP	EXPDAT%U.DMP	

Add Row Remove Row

File Actions

☐ Delete Existing Dump Files

☒ Append Timestamp to Dump, Log and Job names

☒ Compression

☐ Encryption Password: Transparent

☐ Copy files to Oracle Object Storage Service (OSS)

OSS Transfer

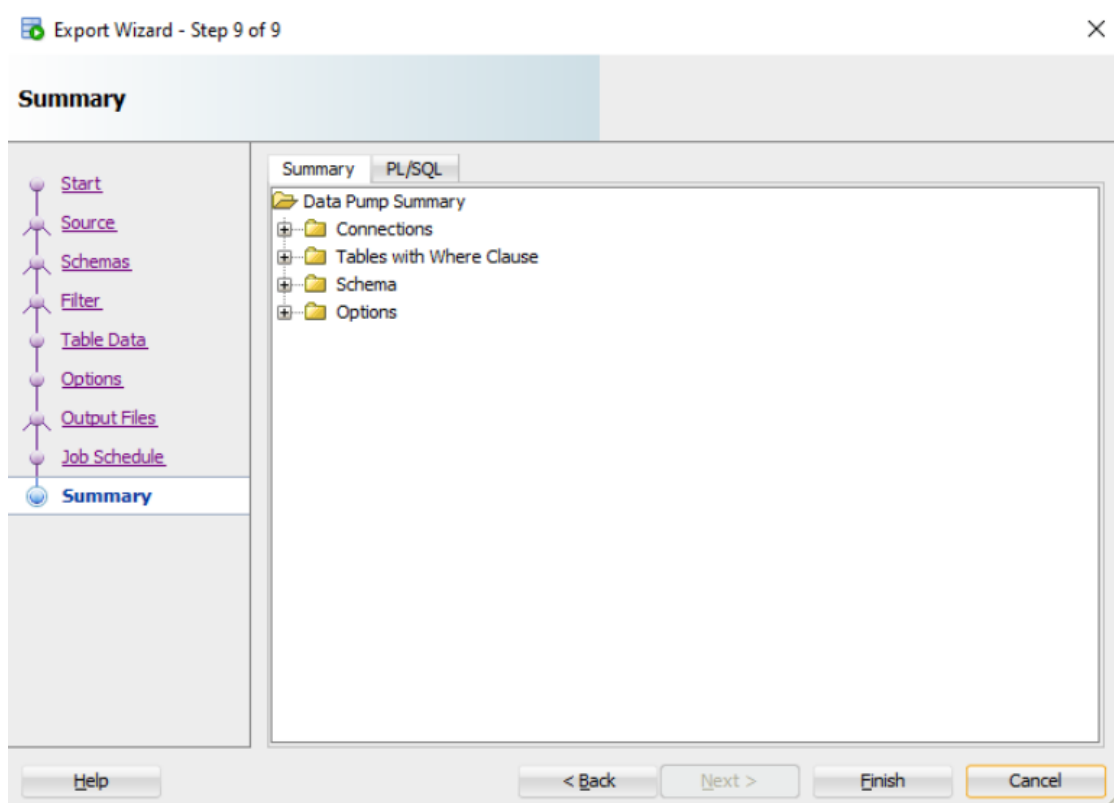
☐ Oracle Directory path override: DBA\admin\orcl\dpdump/

Data Pump Export Job Name: EXP_SD_1164

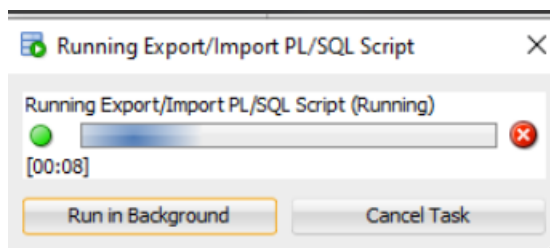
☐ Proceed to summary.

Help < Back Next > Finish Cancel

- For last 2 step we can simply click next without changing any field and then finish



- On clicking finish, we will get follow popup in screen

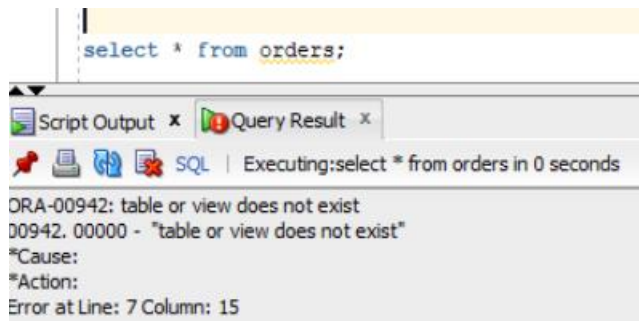


- When we go to our folder (i.e. 'G:\BscCsit\sem7\DBA\DBABackup'), we can see our export or backup

Local Disk (G:) > BscCsit > sem7 > DBA > DBABackup

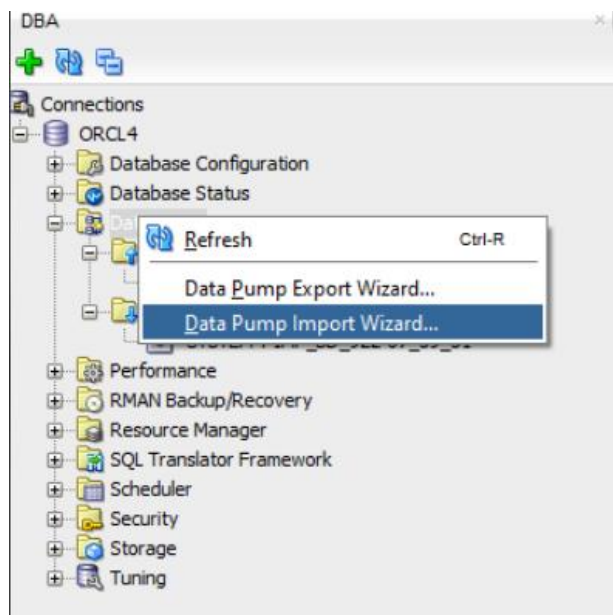
Name	Date modified	Type	Size
EXPDAT01-11_49_04.DMP	12/12/2021 11:50 AM	Memory Dump File	60 KB
EXPDAT-11_49_03	12/12/2021 11:51 AM	Text Document	1 KB

Suppose we drop Orders table from our database and we like to get it back using our export file then we can perform import operation.

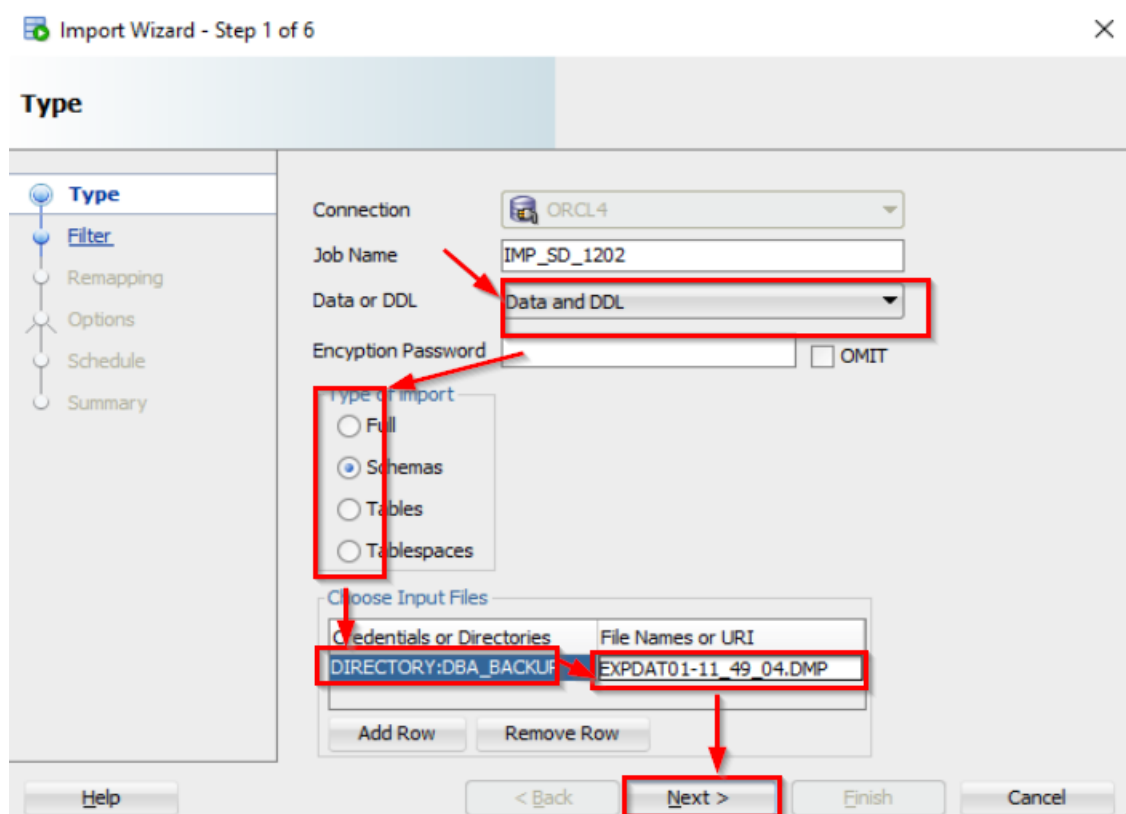


To import:

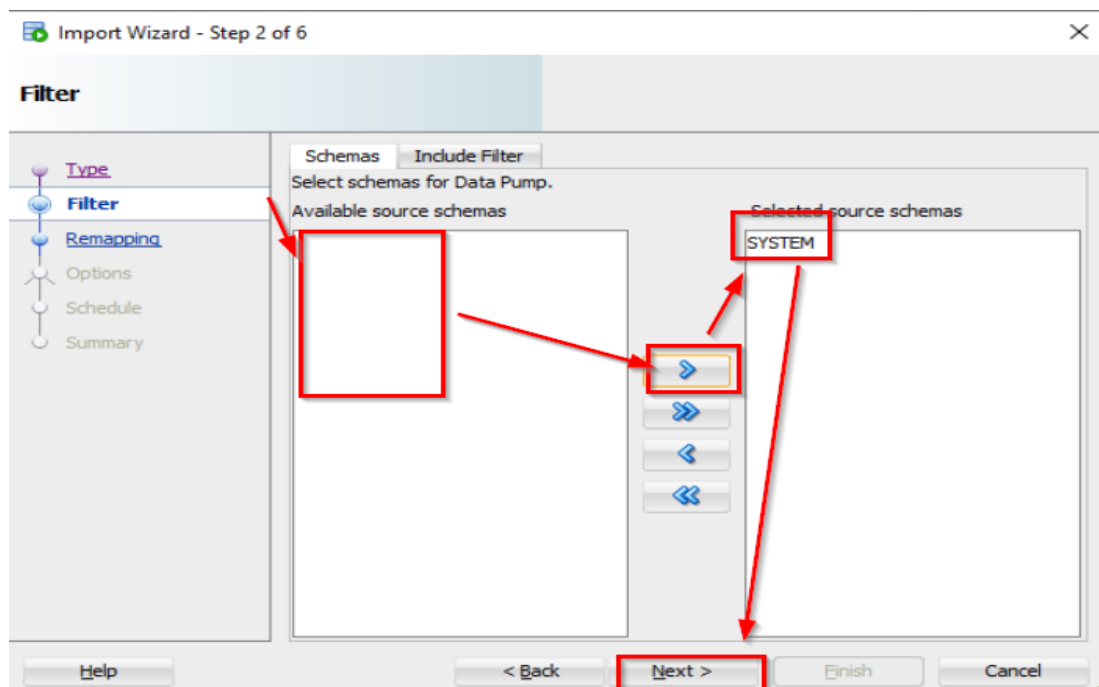
- Same as in export, from connection we right click Data Pump and click Data Pump Import Wizard



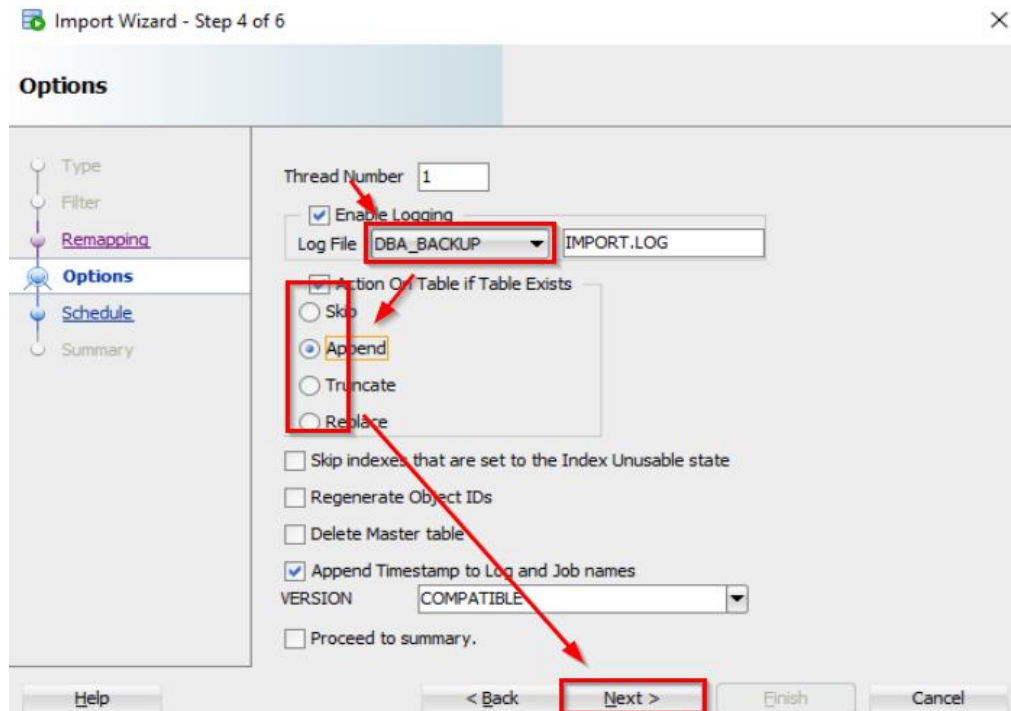
- Same as in export, to import schemas we select schemas and to import tables we select tables along with preferred Data or DDL and then select directory on which we had exported our data and we copy the dump file name from the location where we had exported our data (i.e. 'G:\BscCsit\sem7\DBA\DBABackup') and paste it in file names or URI section and click next



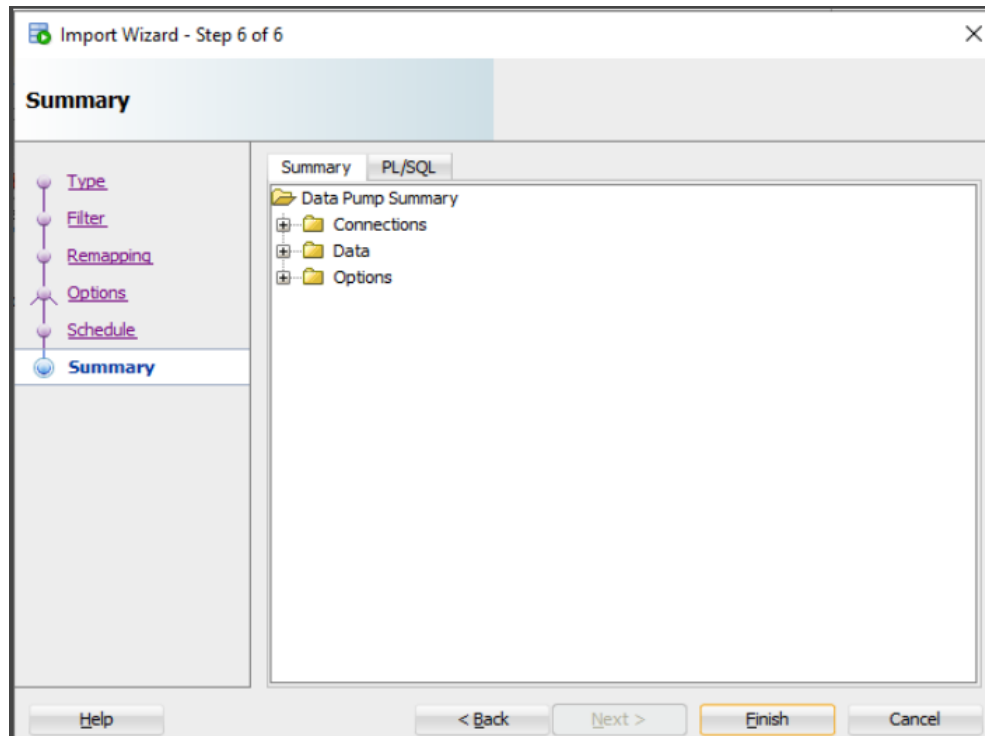
- We select Available source schemas and pass it to Selected source schemas and then click next



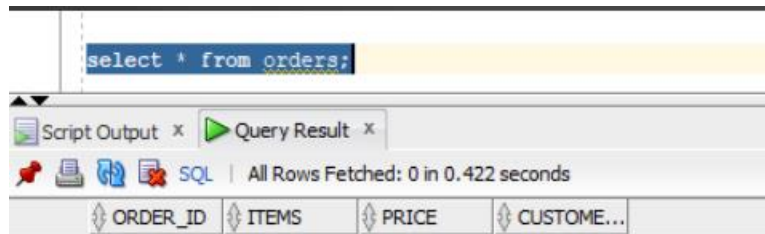
- In next step, we do not need to change anything so simply click next step and in next step we select the log file on which we had exported data and select necessary action on table (in this case as we had deleted orders table so to get order tables we append it) and click next.



- We do not need to change anything in next two step simply click next and then finish.



- Now the orders table as well as all other data in export file is being appended to our database. When we run `SELECT * FROM orders;` query we are able to get orders table but initially we were getting an error.



To drop backup directory

Query:

```
DROP DIRECTORY dba_backup;
```

Output:

```
Directory DBA_BACKUP dropped.
```

Conclusion and Discussion:

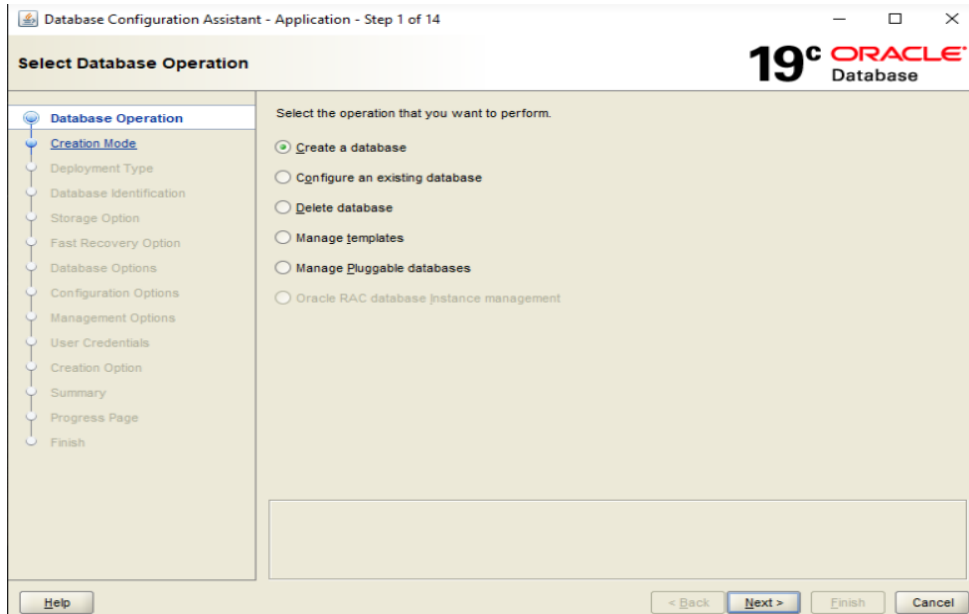
Hence, we are able to create backup directory using sqlconnector and able to import and export table data and schemas using data pump.

Lab 5

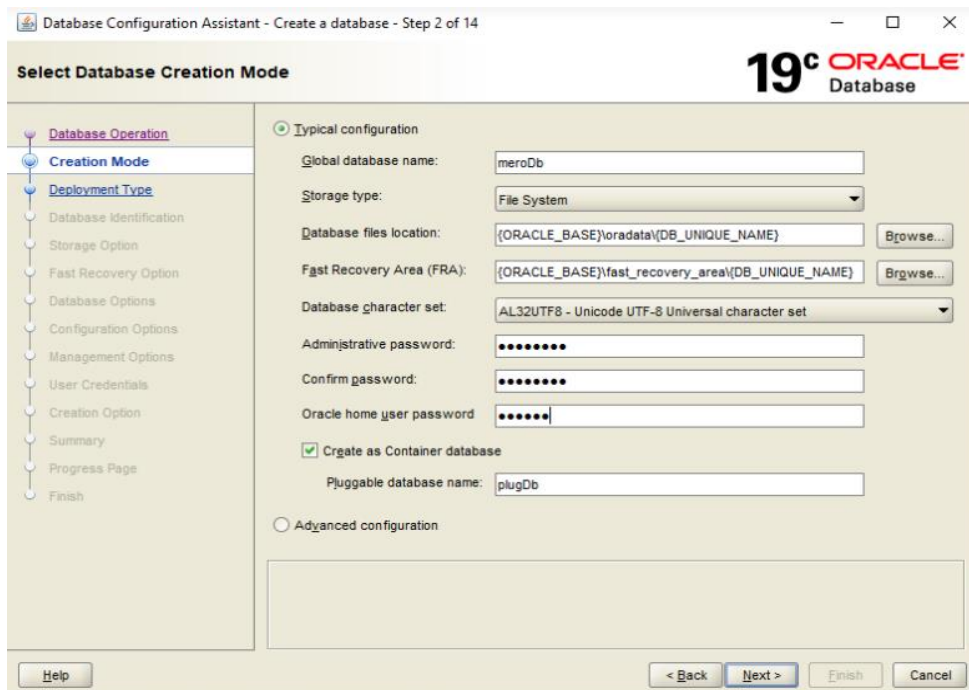
Create a pluggable database using DBCA.

Steps while creating pluggable database using DBCA:

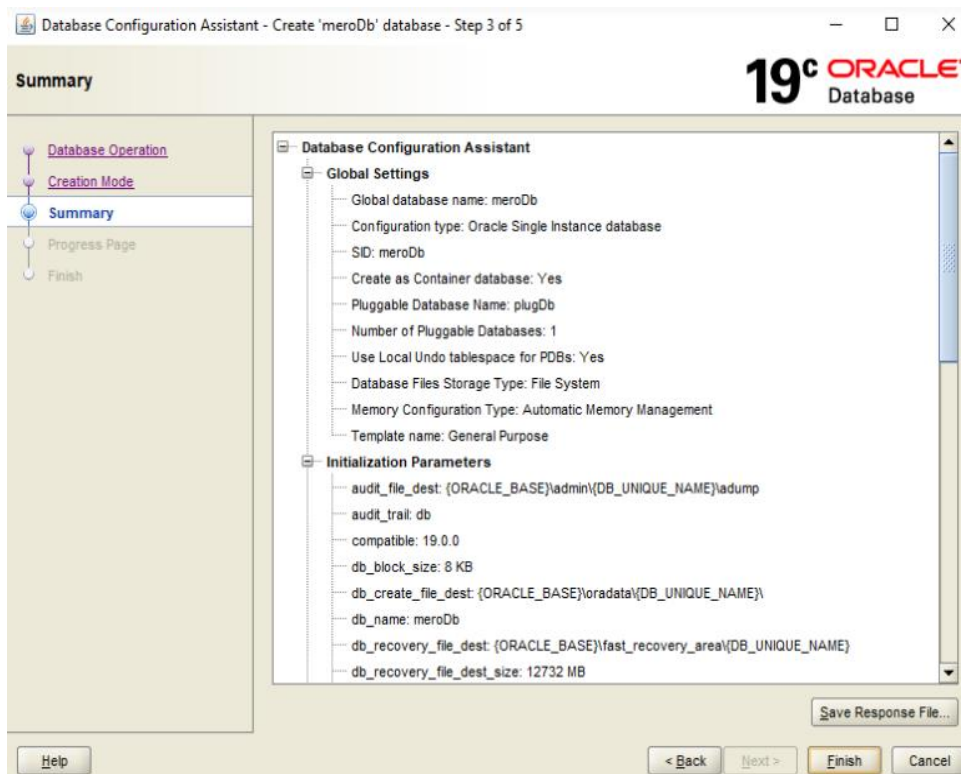
Step 1: Create a database is selected and click next.



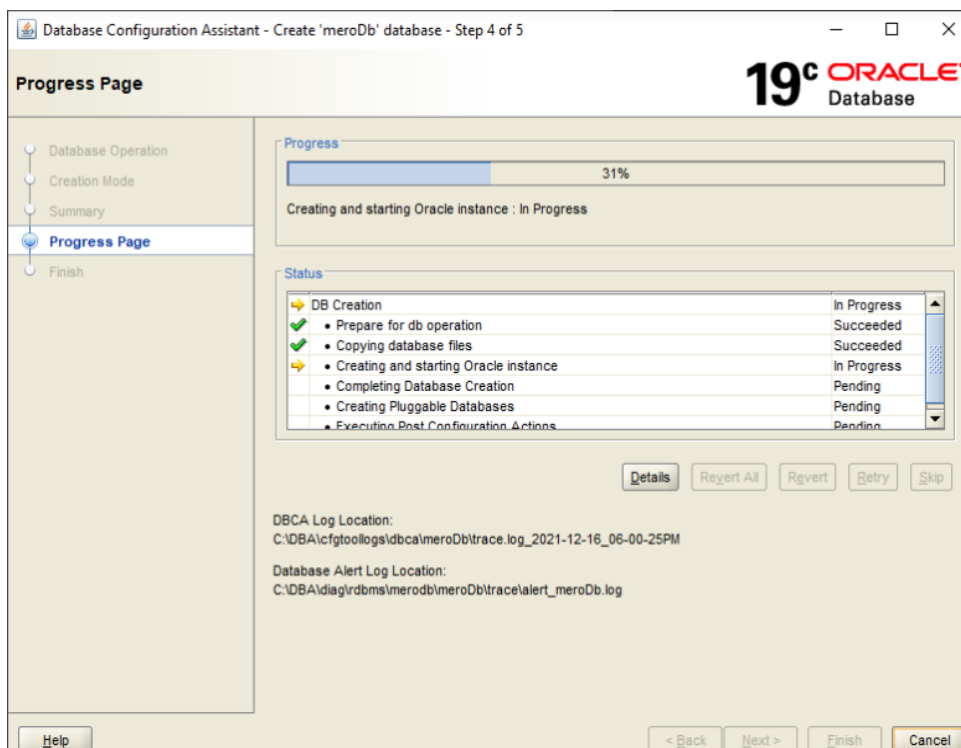
Step 2: Global name for database is given and required password along with oracle home user password is given. Then we tick create as container database and give name for pluggable database.



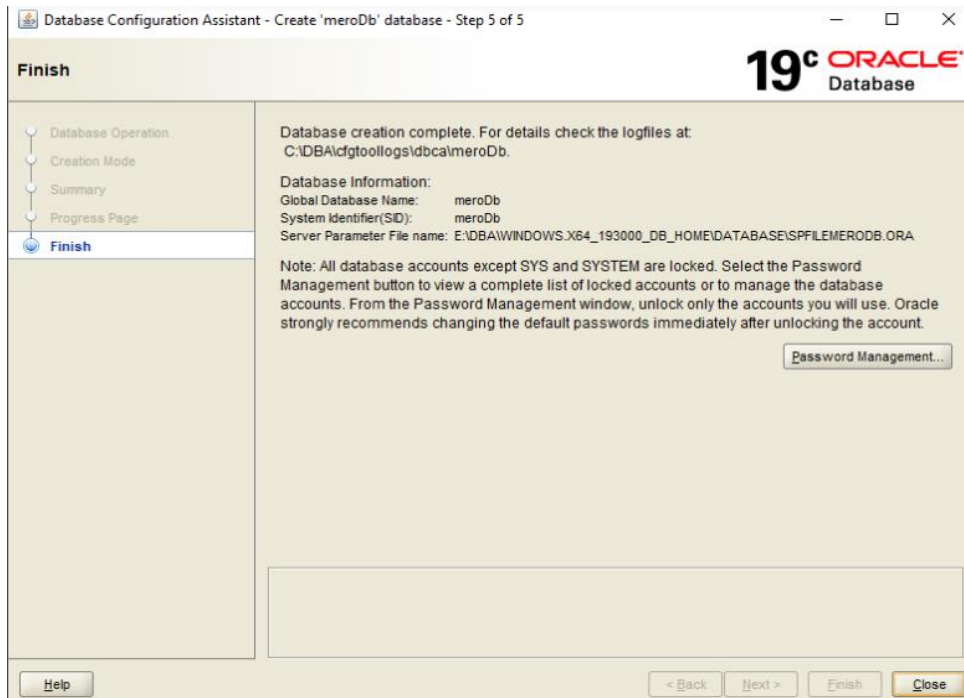
Step 3: In this step, we simply click finish.



Step 4: The necessary process to create and configure database is carried out by the system here.



Step 5: After progress is completed, we can see various information of created database. We can then click close process to finish the creating process of pluggable database using DBCA



Lab 6

1) Backup using rman.

Steps involved while using rman for back:

Step 1: Goto bin folder of setup file and open cmd from that location.

Step 2: Give command as below as show in screenshot

...\bin>rman target/

```
E:\DBA\WINDOWS.X64_193000_db_home\bin>rman target/

Recovery Manager: Release 19.0.0.0.0 - Production on Fri Dec 17 07:33:40 2021
Version 19.3.0.0.0

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

connected to target database: MERODB (DBID=2207720884)
```

RMAN> backup database;

```
RMAN> backup database
2> ;

Starting backup at 17-DEC-21
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=267 device type=DISK
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of backup command at 12/17/2021 07:34:21
RMAN-06149: cannot BACKUP DATABASE in NOARCHIVELOG mode
```

If database is in NOARCHIVELOG mode than we need to shutdown it immediately using following command

RMAN> shutdown immediate;

```
RMAN> shutdown immediate;

database closed
database dismounted
Oracle instance shut down
```

RMAN> startup mount;

```
RMAN> startup mount;

connected to target database (not started)
Oracle instance started
database mounted

Total System Global Area      1677719744 bytes

Fixed Size                     9267392 bytes
Variable Size                  956301312 bytes
Database Buffers               704643072 bytes
Redo Buffers                   7507968 bytes
```

RMAN> backup database;

NOTE: we can also give some tag to backup data using command as backup database tag 'you_special_name_here';

```
RMAN> backup database;

Starting backup at 17-DEC-21
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=179 device type=DISK
channel ORA_DISK_1: starting full database backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=C:\ORADATA\HEROOD\DATAFILE\01_HF_SYSTEM_VPDRWB3_.DBF
input datafile file number=00003 name=C:\ORADATA\HEROOD\DATAFILE\01_HF_SYSAUX_VPD5R8Z_.DBF
input datafile file number=00004 name=C:\ORADATA\HEROOD\DATAFILE\01_HF_UNDOTBS1_VPD7SRF_.DBF
input datafile file number=00007 name=C:\ORADATA\HEROOD\DATAFILE\01_HF_USERS_VPD7V53_.DBF
channel ORA_DISK_1: starting piece 1 at 17-DEC-21
channel ORA_DISK_1: finished piece 1 at 17-DEC-21
piece handle=C:\ORAF\FAST_RECOVERY_AREA\HEROOD\BACKUPSET\2021_12_17\01_HF_RMANF_TAG20211217T073752_VQVCPWF_ .BKP tag=TAG20211217T073752 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
channel ORA_DISK_1: starting full database backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00009 name=C:\ORADATA\HEROOD\3818548F22064CD098A38282967361DF\DATAFILE\01_HF_SYSAUX_VPGF6P8_.DBF
input datafile file number=00011 name=C:\ORADATA\HEROOD\3818548F22064CD098A38282967361DF\DATAFILE\01_HF_SYSTEM_VPGF6P9_.DBF
input datafile file number=00012 name=C:\ORADATA\HEROOD\3818548F22064CD098A38282967361DF\DATAFILE\01_HF_UNDOTBS1_VPGF6PD_.DBF
input datafile file number=00013 name=C:\ORADATA\HEROOD\3818548F22064CD098A38282967361DF\DATAFILE\01_HF_USERS_VPGG4HC_.DBF
channel ORA_DISK_1: starting piece 1 at 17-DEC-21
channel ORA_DISK_1: finished piece 1 at 17-DEC-21
piece handle=C:\ORAF\FAST_RECOVERY_AREA\HEROOD\3818548F22064CD098A38282967361DF\BACKUPSET\2021_12_17\01_HF_RMANF_TAG20211217T073752_VQVQ6WR_ .BKP tag=TAG20211217T073752
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
channel ORA_DISK_1: starting full database backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00005 name=C:\ORADATA\HEROOD\DATAFILE\01_HF_SYSAUX_VPD5R8Z_.DBF
input datafile file number=00006 name=C:\ORADATA\HEROOD\DATAFILE\01_HF_SYSTEM_VPD5R8Z_.DBF
input datafile file number=00008 name=C:\ORADATA\HEROOD\DATAFILE\01_HF_UNDOTBS1_VPD7SRF_.DBF
channel ORA_DISK_1: starting piece 1 at 17-DEC-21
channel ORA_DISK_1: finished piece 1 at 17-DEC-21
piece handle=C:\ORAF\FAST_RECOVERY_AREA\HEROOD\51950038BED4F0193F02A60ABD22869\BACKUPSET\2021_12_17\01_HF_RMANF_TAG20211217T073752_VQVQOYF_ .BKP tag=TAG20211217T073752
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 17-DEC-21

Starting Control File and SPFILE Autobackup at 17-DEC-21
piece handle=C:\ORAF\FAST_RECOVERY_AREA\HEROOD\AUTOBACKUP\2021_12_17\01_HF_5_10P1518519_VQVF6B3_ .BKP comment=NONE
Finished Control File and SPFILE Autobackup at 17-DEC-21
```

RMAN> list backup;

```
RMAN> list backup;

List of Backup Sets
=====

BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
1 Full 1.11G DISK 00:00:14 17-DEC-21
BP Key: 1 Status: AVAILABLE Compressed: NO Tag: TAG20211217T073752
Piece Name: C:\ORAF\FAST_RECOVERY_AREA\HEROOD\BACKUPSET\2021_12_17\01_HF_RMANF_TAG20211217T073752_VQVCPWF_ .BKP
List of Datafiles in backup set 1
File LV Type Ckp SCN Ckp Time Abs Fuz SCN Sparse Name
-----
1 Full 2003991 17-DEC-21 NO C:\ORADATA\HEROOD\DATAFILE\01_HF_SYSTEM_VPDRWB3_.DBF
3 Full 2003991 17-DEC-21 NO C:\ORADATA\HEROOD\DATAFILE\01_HF_SYSAUX_VPD5R8Z_.DBF
4 Full 2003991 17-DEC-21 NO C:\ORADATA\HEROOD\DATAFILE\01_HF_UNDOTBS1_VPD7SRF_.DBF
7 Full 2003991 17-DEC-21 NO C:\ORADATA\HEROOD\DATAFILE\01_HF_USERS_VPD7V53_.DBF

BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
2 Full 441.60M DISK 00:00:00 17-DEC-21
BP Key: 2 Status: AVAILABLE Compressed: NO Tag: TAG20211217T073752
Piece Name: C:\ORAF\FAST_RECOVERY_AREA\HEROOD\3818548F22064CD098A38282967361DF\BACKUPSET\2021_12_17\01_HF_RMANF_TAG20211217T073752_VQVQ6WR_ .BKP
List of Datafiles in backup set 2
Container ID: 3, POB Name: PLUGDB
File LV Type Ckp SCN Ckp Time Abs Fuz SCN Sparse Name
-----
9 Full 2007393 16-DEC-21 NO C:\ORADATA\HEROOD\3818548F22064CD098A38282967361DF\DATAFILE\01_HF_SYSTEM_VPGF6P9_.DBF
10 Full 2007393 16-DEC-21 NO C:\ORADATA\HEROOD\3818548F22064CD098A38282967361DF\DATAFILE\01_HF_SYSAUX_VPGF6P8_.DBF
11 Full 2007393 16-DEC-21 NO C:\ORADATA\HEROOD\3818548F22064CD098A38282967361DF\DATAFILE\01_HF_UNDOTBS1_VPGF6PD_.DBF
12 Full 2007393 16-DEC-21 NO C:\ORADATA\HEROOD\3818548F22064CD098A38282967361DF\DATAFILE\01_HF_USERS_VPGG4HC_.DBF
```

```
BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
3 Full 468.91M DISK 00:00:00 17-DEC-21
BP Key: 3 Status: AVAILABLE Compressed: NO Tag: TAG20211217T073752
Piece Name: C:\ORAF\FAST_RECOVERY_AREA\HEROOD\51950038BED4F0193F02A60ABD22869\BACKUPSET\2021_12_17\01_HF_RMANF_TAG20211217T073752_VQVQOYF_ .BKP
List of Datafiles in backup set 3
Container ID: 2, POB Name: PDB$SEED
File LV Type Ckp SCN Ckp Time Abs Fuz SCN Sparse Name
-----
5 Full 191512 16-DEC-21 NO C:\ORADATA\HEROOD\DATAFILE\01_HF_SYSTEM_VPD5R8Z_.DBF
6 Full 191512 16-DEC-21 NO C:\ORADATA\HEROOD\DATAFILE\01_HF_SYSAUX_VPD5R8Z_.DBF
8 Full 191512 16-DEC-21 NO C:\ORADATA\HEROOD\DATAFILE\01_HF_UNDOTBS1_VPD7SRF_.DBF

BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
4 Full 17.95M DISK 00:00:01 17-DEC-21
BP Key: 4 Status: AVAILABLE Compressed: NO Tag: TAG20211217T073841
Piece Name: C:\ORAF\FAST_RECOVERY_AREA\HEROOD\AUTOBACKUP\2021_12_17\01_HF_5_10P1518519_VQVF6B3_ .BKP
SPFILE Included: Modification time: 17-DEC-21
SPFILE db unique name: HEROOD
Control File Included: ckp SCN: 2003991 Ckp time: 17-DEC-21
```

RMAN> alter database open;

```
RMAN> alter database open;

Statement processed
```

2) To be familiarize with stored procedure.

Query:

```
CREATE OR REPLACE PROCEDURE greetings
AS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello world');
END;
/
```

Output:

```
Procedure GREETINGS compiled
```

Query to execute above procedure:

```
execute greetings;
```

Output:

```
Connecting to the database ORCL2.
Hello World
Process exited.
Disconnecting from the database ORCL2.
```

NOTE: we can also write exec in place of execute

Next example:

```
CREATE OR REPLACE PROCEDURE selecctedcustomers
IS
BEGIN
    UPDATE customers
    SET first_name='Ram'
    WEHRE customer_id=6;
END;
/
```

Output:

```
Procedure SELECTEDCUSTOMERS compiled
```

Query to execute above procedure:

```
exec selectedcustomers;
```

Conclusion and discussion:

Hence, we are able to backup database using rman and also become familiarize with stored procedure.

Lab 7

To be familiarize with trigger.

First we need to create table for which we will give privilege to store trigger

Query:

```
CREATE TABLE audits(  
    audit_id number GENERATED BY DEFAULT AS IDENTITY primary key,  
    table_name varchar(50),  
    transaction_name varchar(50),  
    by_user varchar(50),  
    transaction_date date  
);
```

Output:

```
Table AUDITS created.
```

Query:

```
CREATE OR REPLACE TRIGGER customers_audit_trg  
    AFTER  
    INSERT OR UPDATE OR DELETE  
    ON customers  
    FOR EACH ROW  
DECLARE  
    transaction_name :=CASE  
        WHEN INSERTING THEN 'INSERT'  
        WHEN UPDATING THEN 'UPDATE'  
        WHEN DELETING THEN 'DELETE'  
    END;  
    INSERT INTO audits (table_name, transaction_name, by_user, transaction_date)  
    VALUES('customers', transaction_name, USER, SYSDATE);  
END;
```


/

Output:

Trigger CUSTOMERS_AUDIT_TRG compiled

Now, if we insert or update or delete in customers table we can then see the information in audits table

Query:

UPDATE customers

SET last_name='karki'

WHERE customer_id = 6;

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
1	1	Bijay	Shrestha	9876543210
2	2	Dipak	Thapa Magar	9860558458
3	3	Ramesh	Neupane	9873333210
4	4	Gaurav	Poudel	9876549990
5	5	Sailesh	Karki	9876541470
6	6	Ram	karki	9876543210

SELECT * FROM audits;

	AUDIT_ID	TABLE_NAME	TRANSACTION_NAME	BY_USER	TRANSACTION_DATE
1	1	customers	UPDATE	SYSTEM	17-DEC-21

Conclusion and discussion:

Hence, we implement trigger and become familiarize about how it keep log file of insert, update and delete in given table.