NAME: Meet, Rahul, Raj	**DATE**: 03/04/2024
Roll Number: A058, A059, A072	**BATCH**: A - 2
Subject: Database Administrator	**PROGRAME**: Btech IT, 3rd year, 6th sem

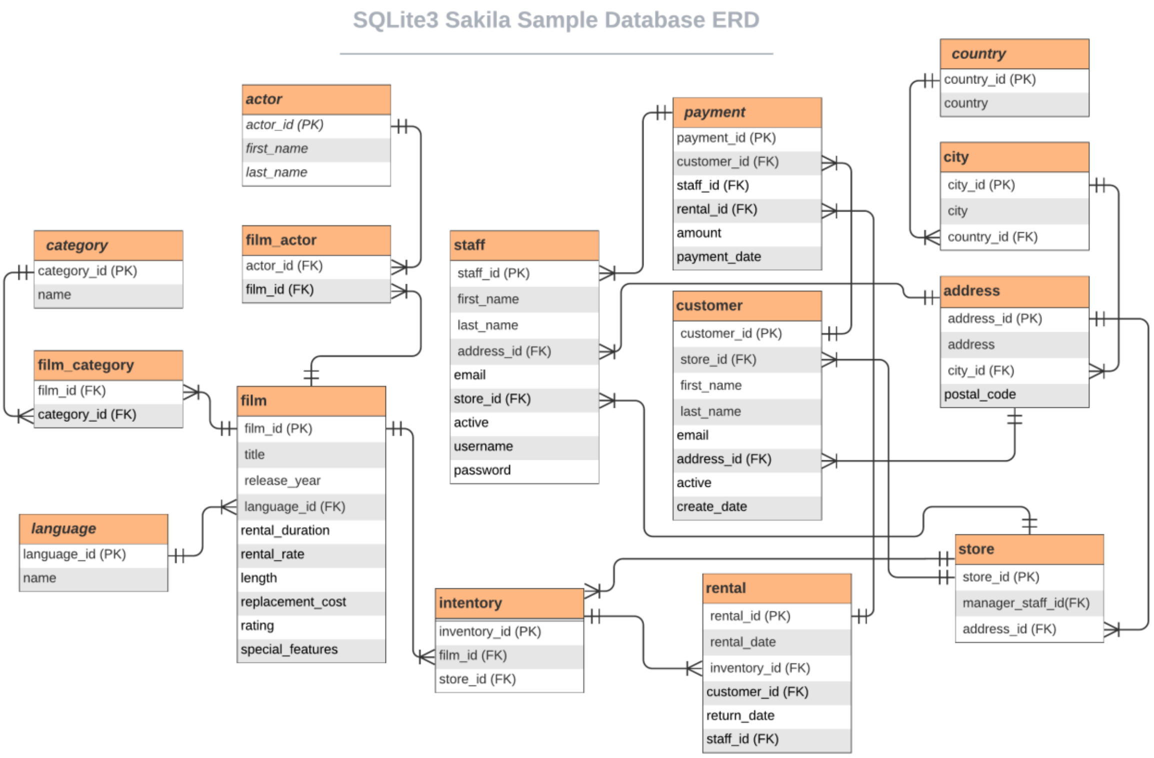
Tasks done:

- Data migration
- User management
- Password policy
- Backup recovery policy

What's our database? [Sakila database]

The Sakila database is a nicely normalised schema modelling a DVD rental store, featuring things like films, actors, film-actor relationships, and a central inventory table that connects films, stores, and rentals.

Schema:



Tables:

The actor Table – The actor table lists information for all actors. The actor table is joined to the film table by means of the film_actor table.

actor_id: A surrogate primary key used to uniquely identify each actor in the table.

first_name: The actor first name.

last_name: The actor last name.

last_update: When the row was created or most recently updated.

The address Table – The address table contains address information for customers, staff, and stores. The address table primary key appears as a foreign key in the customer, staff, and store tables.

address_id: A surrogate primary key used to uniquely identify each address in the table.

address: The first line of an address.

address2: An optional second line of an address.

district: The region of an address, this may be a state, province, prefecture, etc.

city_id: A foreign key pointing to the city table.

postal_code: The postal code or ZIP code of the address (where applicable).

phone: The telephone number for the address.

last_update: When the row was created or most recently updated.

location: A Geometry column with a spatial index on it.

The category Table – The category table lists the categories that can be assigned to a film. The category table is joined to the film table by means of the film_category table.

category_id: A surrogate primary key used to uniquely identify each category in the table.

name: The name of the category.

last_update: When the row was created or most recently updated.

The city Table – The city table contains a list of cities. The city table is referred to by a foreign key in the address table and refers to the country table using a foreign key.

city_id: A surrogate primary key used to uniquely identify each city in the table.

city: The name of the city.

country_id: A foreign key identifying the country that the city belongs to.

last_update: When the row was created or most recently updated.

The country Table – The country table contains a list of countries. The country table is referred to by a foreign key in the city table.

country_id: A surrogate primary key used to uniquely identify each country in the table.

country: The name of the country.

last_update: When the row was created or most recently updated.

The customer Table – The customer table contains a list of all customers. The customer table is referred to in the payment and rental tables and refers to the address and store tables using foreign keys.

customer_id: A surrogate primary key used to uniquely identify each customer in the table.

store_id: A foreign key identifying the customer “home store.” Customers are not limited to renting only from this store, but this is the store at which they generally shop.

first_name: The customer first name.

last_name: The customer last name.

email: The customer email address.

address_id: A foreign key identifying the customer address in the address table.

active: Indicates whether the customer is an active customer. Setting this to FALSE serves as an alternative to deleting a customer outright. Most queries should have a WHERE active = TRUE clause.

create_date: The date the customer was added to the system. This date is automatically set using a trigger during an INSERT.

last_update: When the row was created or most recently updated.

The film Table - The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table. The film table refers to the language table and is referred to by the film_category, film_actor, and inventory tables.

film_id: A surrogate primary key used to uniquely identify each film in the table.

title: The title of the film.

description: A short description or plot summary of the film.

release_year: The year in which the movie was released.

language_id: A foreign key pointing at the language table; identifies the language of the film.

original_language_id: A foreign key pointing at the language table; identifies the original language of the film. Used when a film has been dubbed into a new language.

rental_duration: The length of the rental period, in days.

rental_rate: The cost to rent the film for the period specified in the rental_duration column.

length: The duration of the film, in minutes.

replacement_cost: The amount charged to the customer if the film is not returned or is returned in a damaged state.

rating: The rating assigned to the film. Can be one of: G, PG, PG-13, R, or NC-17.

special_features: Lists which common special features are included on the DVD. Can be zero or more of: Trailers, Commentaries, Deleted Scenes, Behind the Scenes.

last_update: When the row was created or most recently updated.

The film_actor Table – The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing the actor and film. The film_actor table refers to the film and actor tables using foreign keys.

actor_id: A foreign key identifying the actor.

film_id: A foreign key identifying the film.

last_update: When the row was created or most recently updated.

The film_category Table - The film_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the film_category table listing the category and film. The film_category table refers to the film and category tables using foreign keys.

film_id: A foreign key identifying the film.

category_id: A foreign key identifying the category.

last_update: When the row was created or most recently updated.

The film_text Table – The film_text table contains the film_id, title and description columns of the film table, with the contents of the table kept in synchrony with the film table by means of triggers on film table INSERT, UPDATE and DELETE operations (see Section 5.5, "Triggers").

Before MySQL server 5.6.10, the film_text table was the only table in the Sakila sample database that used the MyISAM storage engine. This is because full-text search is used for titles and descriptions of films listed in the film table. MyISAM was used because full-text search support with InnoDB was not available until MySQL server 5.6.10.

film_id: A surrogate primary key used to uniquely identify each film in the table.

title: The title of the film.

description: A short description or plot summary of the film.

The inventory Table – The inventory table contains one row for each copy of a given film in a given store. The inventory table refers to the film and store tables using foreign keys and is referred to by the rental table.

inventory_id: A surrogate primary key used to uniquely identify each item in inventory.

film_id: A foreign key pointing to the film this item represents.

store_id: A foreign key pointing to the store stocking this item.

last_update: When the row was created or most recently updated.

The language Table - The language table is a lookup table listing the possible languages that films can have for their language and original language values. The language table is referred to by the film table.

language_id: A surrogate primary key used to uniquely identify each language.

name: The English name of the language.

last_update: When the row was created or most recently updated.

The payment Table – The payment table records each payment made by a customer, with information such as the amount and the rental being paid for (when applicable). The payment table refers to the customer, rental, and staff tables.

payment_id: A surrogate primary key used to uniquely identify each payment.

customer_id: The customer whose balance the payment is being applied to. This is a foreign key reference to the customer table.

staff_id: The staff member who processed the payment. This is a foreign key reference to the staff table.

rental_id: The rental that the payment is being applied to. This is optional because some payments are for outstanding fees and may not be directly related to a rental.

amount: The amount of the payment.

payment_date: The date the payment was processed.

last_update: When the row was created or most recently updated.

The rental Table - The rental table contains one row for each rental of each inventory item with information about who rented what item, when it was rented, and when it was returned. The rental table refers to the inventory, customer, and staff tables and is referred to by the payment table.

rental_id: A surrogate primary key that uniquely identifies the rental.

rental_date: The date and time that the item was rented.

inventory_id: The item being rented.

customer_id: The customer renting the item.

return_date: The date and time the item was returned.

staff_id: The staff member who processed the rental.

last_update: When the row was created or most recently updated.

The staff Table – The staff table lists all staff members, including information for email address, login information, and picture. The staff table refers to the store and address tables using foreign keys, and is referred to by the rental, payment, and store tables.

staff_id: A surrogate primary key that uniquely identifies the staff member

first_name: The first name of the staff member.

last_name: The last name of the staff member.

address_id: A foreign key to the staff member address in the address table.

picture: A BLOB containing a photograph of the employee.

email: The staff member email address.

store_id: The staff member “home store.” The employee can work at other stores but is generally assigned to the store listed.

active: Whether this is an active employee. If employees leave, their rows are not deleted from this table; instead, this column is set to FALSE.

username: The user name used by the staff member to access the rental system.

password: The password used by the staff member to access the rental system. The password should be stored as a hash using the SHA2() function.

last_update: When the row was created or most recently updated.

The store Table - The store table lists all stores in the system. All inventory is assigned to specific stores, and staff and customers are assigned a “home store”. The store table refers to the staff and address tables using foreign keys and is referred to by the staff, customer, and inventory tables.

store_id: A surrogate primary key that uniquely identifies the store.

manager_staff_id: A foreign key identifying the manager of this store.

address_id: A foreign key identifying the address of this store.

last_update: When the row was created or most recently updated.

Views:

The actor_info View - The actor_info view provides a list of all actors, including the films in which they have performed, broken down by category. The staff_list view incorporates data from the film, actor, category, film_actor, and film_category tables

The customer_list View - The customer_list view provides a list of customers, with first name and last name concatenated together and address information combined into a single view. The customer_list view incorporates data from the customer, address, city, and country tables.

The film_list View - The film_list view contains a formatted view of the film table, with a comma-separated list of actors for each film. The film_list view incorporates data from the film, category, film_category, actor, and film_actor tables.

The nicer_but_slower_film_list View - The nicer_but_slower_film_list view contains a formatted view of the film table, with a comma-separated list of the film's actors. The nicer_but_slower_film_list view differs from the film_list view in the list of actors. The lettercase of the actor names is adjusted so that the first letter of each name is capitalized, rather than having the name in all-caps. As indicated in its name, the nicer_but_slower_film_list view performs additional processing and therefore takes longer to return data than the film_list view. The nicer_but_slower_film_list view incorporates data from the film, category, film_category, actor, and film_actor tables

The sales_by_film_category View - The sales_by_film_category view provides a list of total sales, broken down by individual film category. Because a film can be listed in multiple categories, it is not advisable to calculate aggregate sales by totalling the rows of this view. The sales_by_film_category view incorporates data from the category, payment, rental, inventory, film, film_category, and category tables.

The sales_by_store View - The sales_by_store view provides a list of total sales, broken down by store. The view returns the store location, manager name, and total sales. The sales_by_store view incorporates data from the city, country, payment, rental, inventory, store, address, and staff tables.

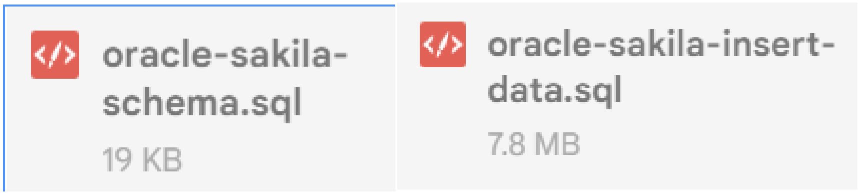
The staff_list View - The staff_list view provides a list of all staff members, including address and store information. The staff_list view incorporates data from the staff and address tables.

DATA MIGRATION

Open sqlplus and connect to the database as sysdba

Then execute the below two files with the syntax: @path\filename.sql

We have 2 files initially one is called as oracle-sakila-schema.sql and the other oracle-sakila-insert-data.sql, wherein upon running the first itll create all the tables and views and when we execute the second file it will insert the data into the tables and views created.



oracle-sakila-schema.sql
19 KB

oracle-sakila-insert-data.sql
7.8 MB

Now our database with all its table is successfully created.

IDENTIFYING STAKEHOLDERS:

- Management: This includes executives and managers who oversee the business operations and use data from the Sakila database for decision-making.
- Customers: These are individuals or entities who rent DVDs from the fictional DVD rental store represented in the Sakila database. Understanding customer behavior and preferences is crucial for the business.
- Employees: This refers to staff members working at the DVD rental store, including sales associates, managers, and administrative personnel. They use the database for day-to-day operations such as managing rentals, processing payments, and maintaining inventory.
- Data Analysts/ (company owner): These professionals analyze the data in the Sakila database to derive insights, trends, and patterns that can help improve business strategies, marketing campaigns, and customer satisfaction.
- Vendors/Partners: Suppliers, business partners, or vendors who interact with the DVD rental store and may require access to certain data or collaborate on projects involving the Sakila database.

USER MANAGEMENT:

Admin User: [THAKKAR]

Role: Database Administrator

Privileges:

- Full access to all tables, views, and stored procedures.
- Ability to create, modify, and drop tables and indexes.
- Grant and revoke privileges to other users.
- Backup and restore database operations.
- Monitor and manage database performance.

Manager User: [RAHUL]

Role: Manager

Privileges:

- Read and write access to customer information, rental history, and sales data.
- Ability to generate reports and perform analytics on customer behavior.

- Read access to financial data (e.g., revenue reports).
- Cannot modify the schema or anything about the structure of the database.

Sales User: [PARTH]

Role: Sales Associate

Privileges:

- Read and write access to customer information and rental transactions.
- Process rentals, returns, and payments.
- View inventory status and availability.
- Cannot access sensitive financial data.
- Cannot modify the schema or anything about the structure of the database.

Data Analyst User: [RAJ]

Role: Data Analyst

Privileges:

- Read access to all tables and views for performing data analysis.
- Execute stored procedures for generating reports and analyzing trends.
- Create temporary tables or views for data manipulation.
- Cannot modify or delete existing data, tables, or views.

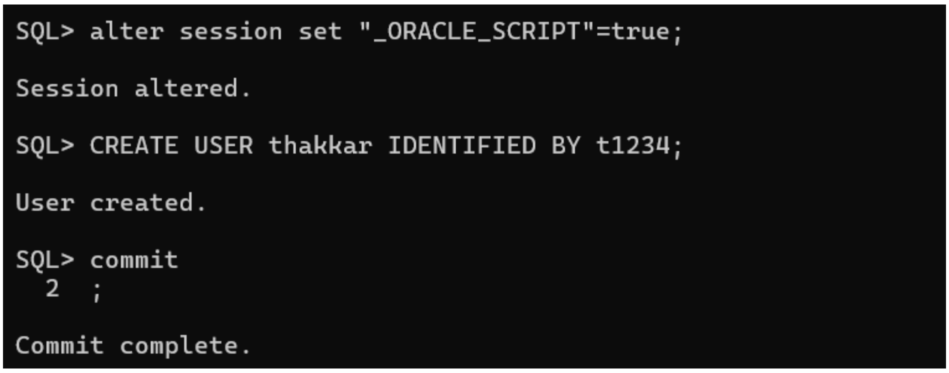
Recovery and Maintenance User: [NIKHIL]

Role: Database Maintenance Specialist

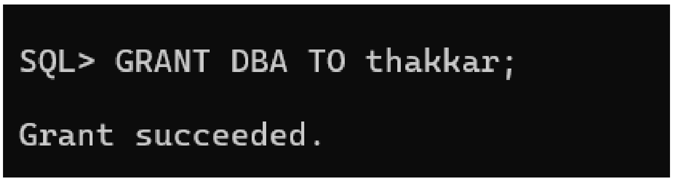
Privileges:

- Backup and restore database operations.
- Perform routine database maintenance tasks such as reindexing, updating statistics, and optimizing queries.
- Identify and troubleshoot issues related to database corruption or performance degradation.
- Access to system logs and diagnostic tools for analyzing database errors and failures.
- Cannot modify database structure, grant privileges to other users, or access sensitive financial or customer data unless required for recovery or maintenance purposes.

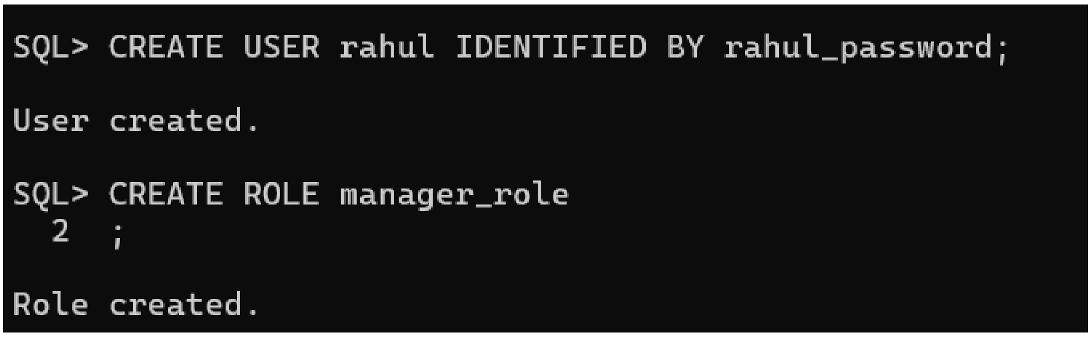
CREATING USER:



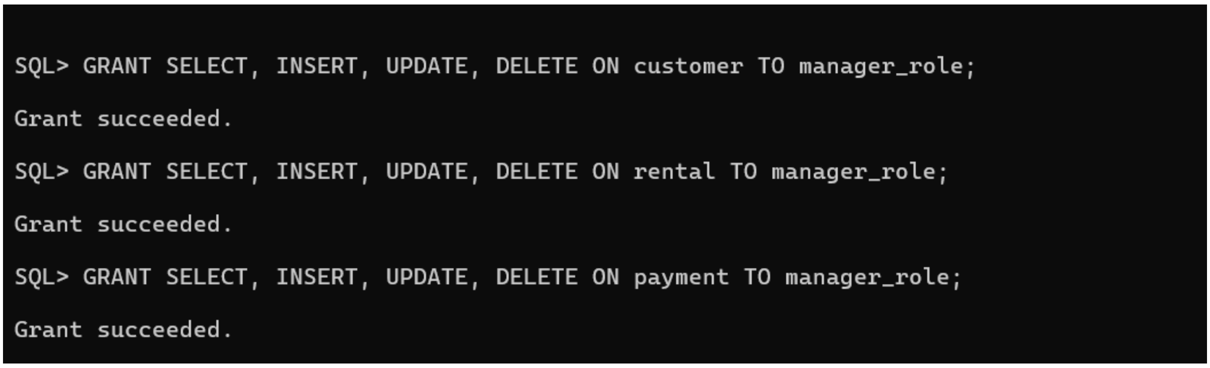
```
SQL> alter session set "_ORACLE_SCRIPT"=true;

Session altered.

SQL> CREATE USER thakkar IDENTIFIED BY t1234;

User created.

SQL> commit
  2  ;

Commit complete.
```



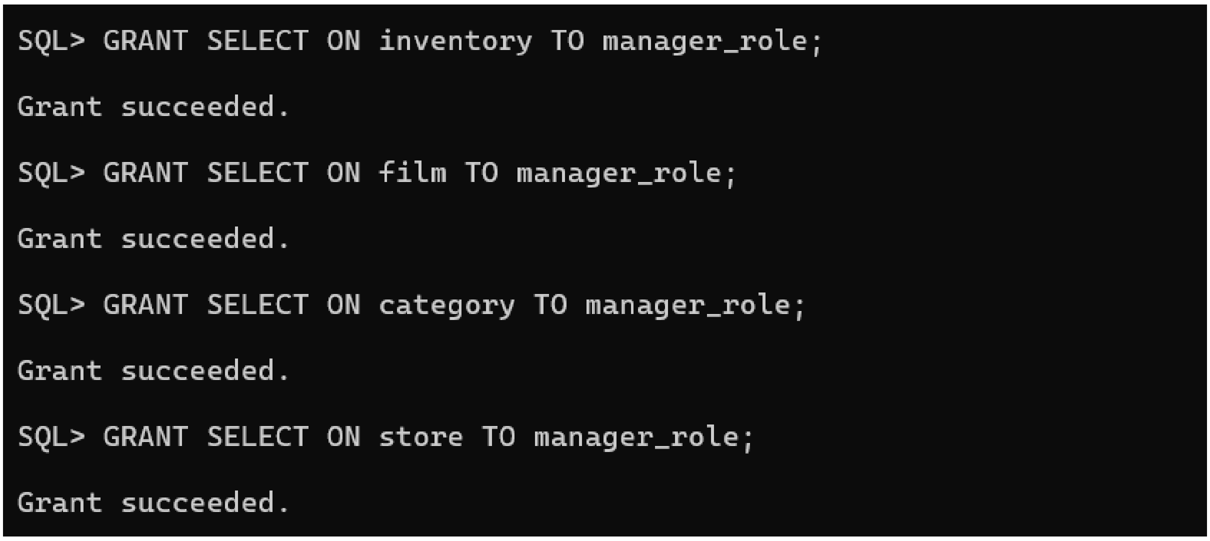
```
SQL> GRANT DBA TO thakkar;

Grant succeeded.
```



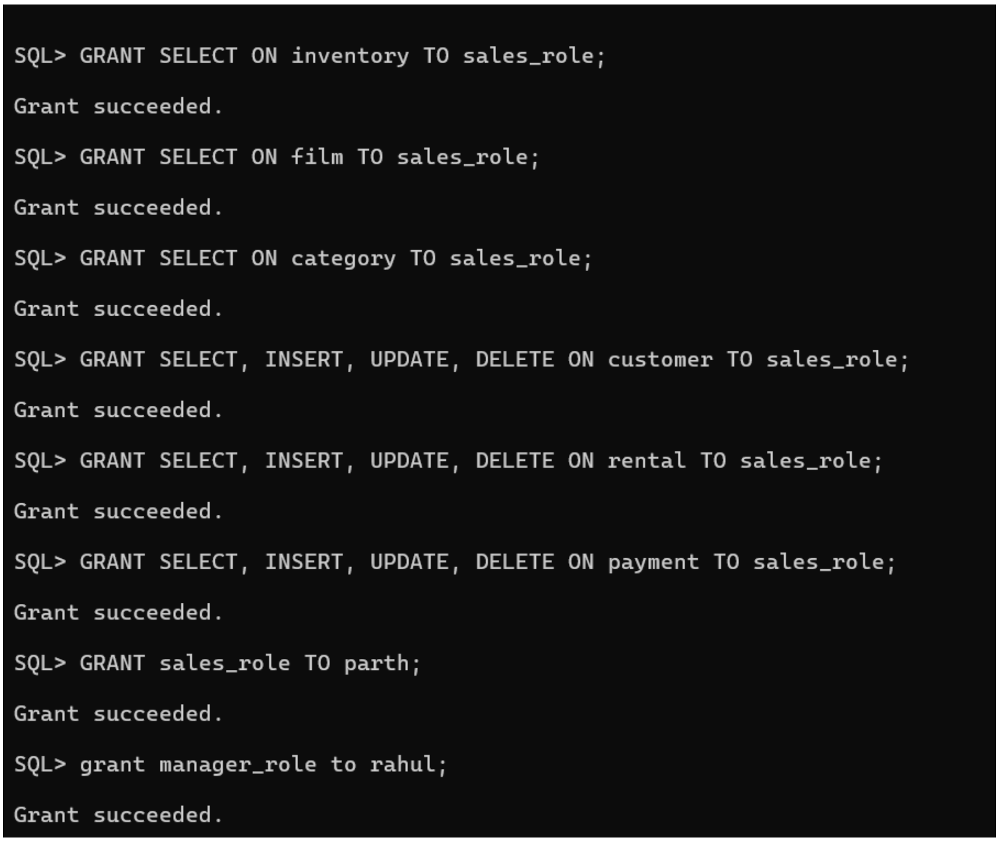
```
SQL> CREATE USER rahul IDENTIFIED BY rahul_password;

User created.

SQL> CREATE ROLE manager_role
  2  ;

Role created.
```



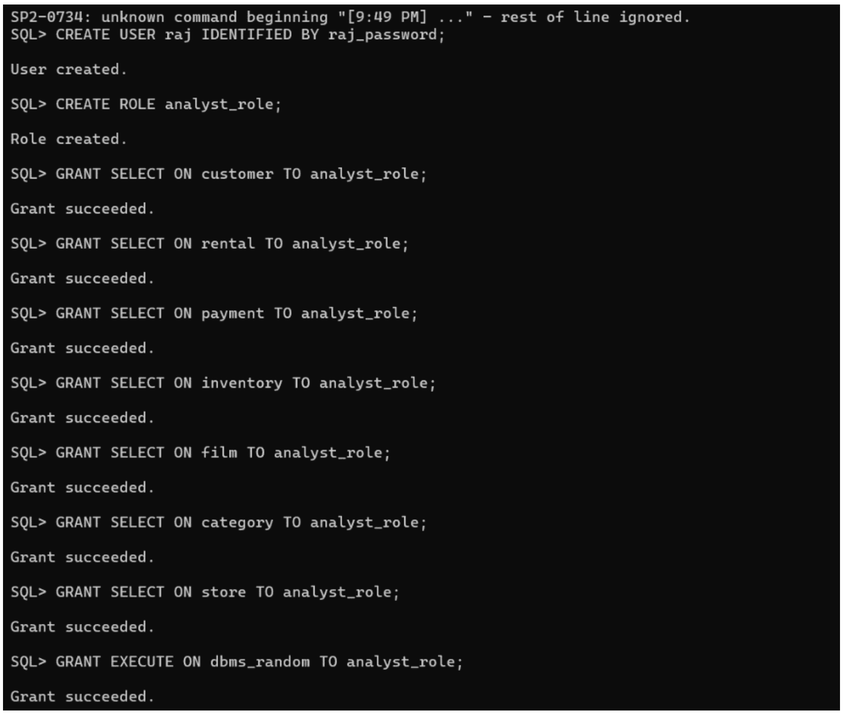
```
SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON customer TO manager_role;

Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON rental TO manager_role;

Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON payment TO manager_role;

Grant succeeded.
```



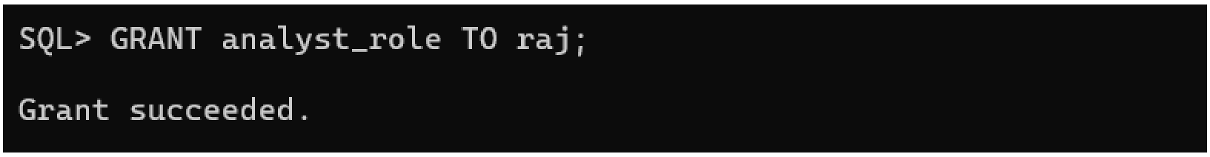
```
SQL> GRANT SELECT ON inventory TO manager_role;

Grant succeeded.

SQL> GRANT SELECT ON film TO manager_role;

Grant succeeded.

SQL> GRANT SELECT ON category TO manager_role;

Grant succeeded.

SQL> GRANT SELECT ON store TO manager_role;

Grant succeeded.
```



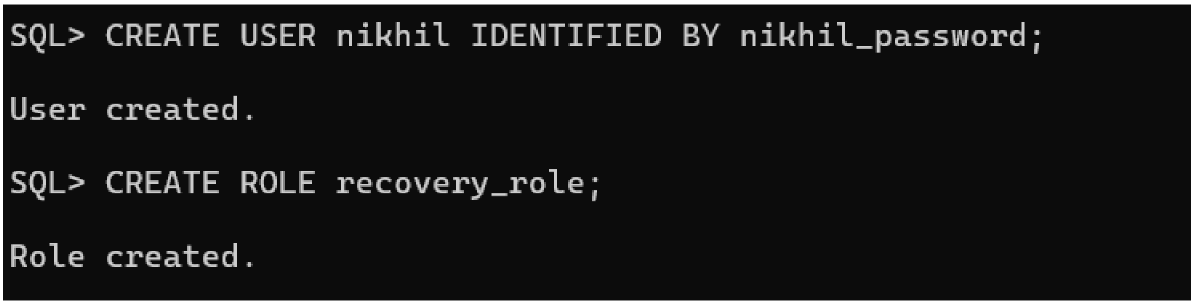
```
SQL> GRANT SELECT ON inventory TO sales_role;

Grant succeeded.

SQL> GRANT SELECT ON film TO sales_role;

Grant succeeded.

SQL> GRANT SELECT ON category TO sales_role;

Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON customer TO sales_role;

Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON rental TO sales_role;

Grant succeeded.

SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON payment TO sales_role;

Grant succeeded.

SQL> GRANT sales_role TO parth;

Grant succeeded.

SQL> grant manager_role to rahul;

Grant succeeded.
```



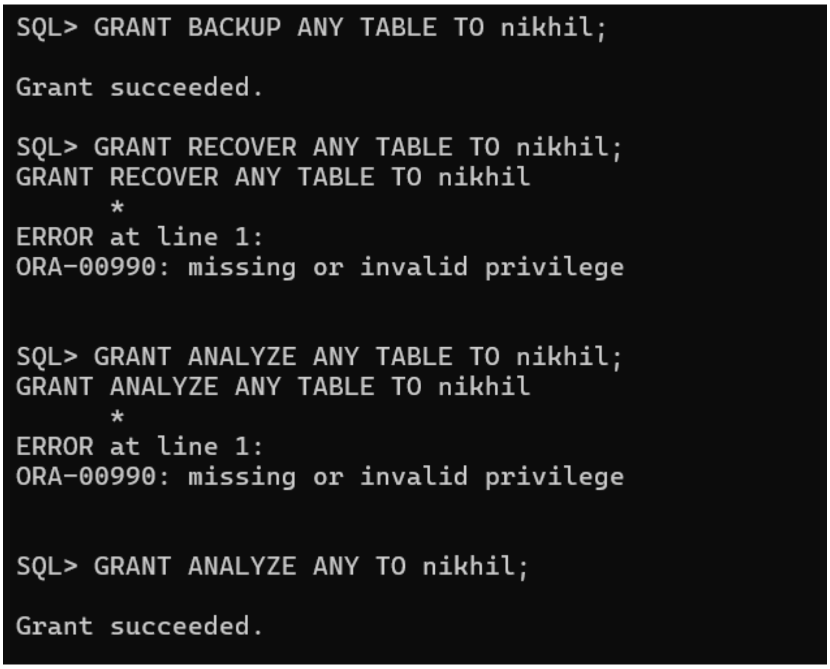
```
SP2-0734: unknown command beginning "[9:49 PM] ..." - rest of line ignored.
SQL> CREATE USER raj IDENTIFIED BY raj_password;

User created.

SQL> CREATE ROLE analyst_role;

Role created.

SQL> GRANT SELECT ON customer TO analyst_role;

Grant succeeded.

SQL> GRANT SELECT ON rental TO analyst_role;

Grant succeeded.

SQL> GRANT SELECT ON payment TO analyst_role;

Grant succeeded.

SQL> GRANT SELECT ON inventory TO analyst_role;

Grant succeeded.

SQL> GRANT SELECT ON film TO analyst_role;

Grant succeeded.

SQL> GRANT SELECT ON category TO analyst_role;

Grant succeeded.

SQL> GRANT SELECT ON store TO analyst_role;

Grant succeeded.

SQL> GRANT EXECUTE ON dbms_random TO analyst_role;

Grant succeeded.
```



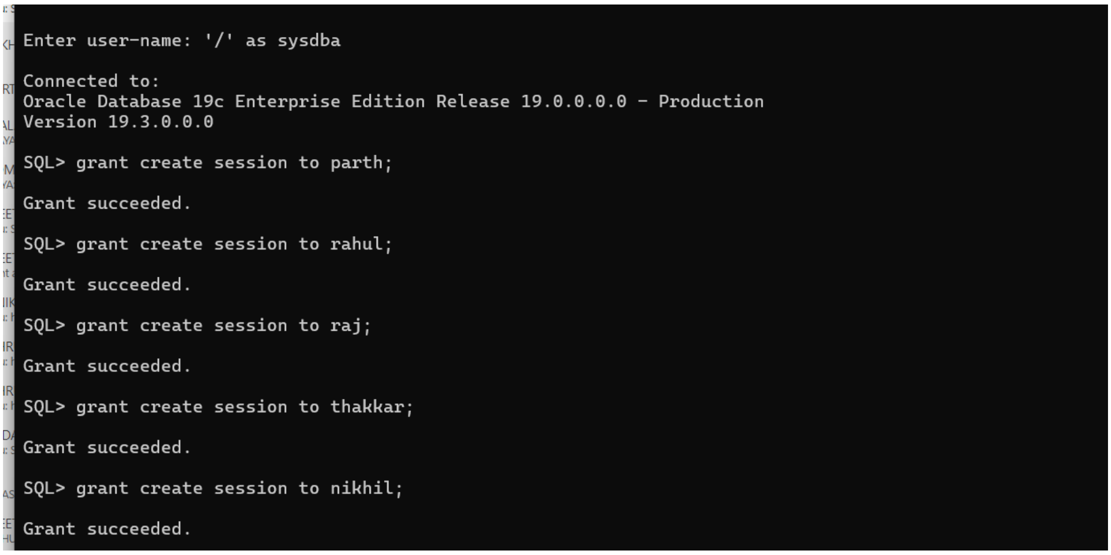
```
SQL> GRANT analyst_role TO raj;

Grant succeeded.
```



```
SQL> CREATE USER nikhil IDENTIFIED BY nikhil_password;

User created.

SQL> CREATE ROLE recovery_role;

Role created.
```

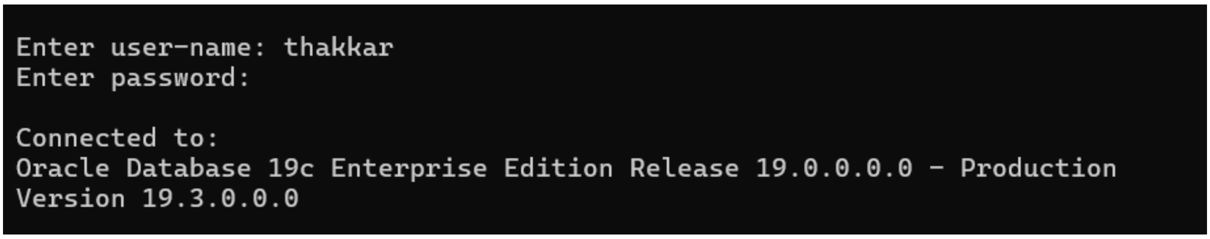


```
SQL> GRANT BACKUP ANY TABLE TO nikhil;

Grant succeeded.

SQL> GRANT RECOVER ANY TABLE TO nikhil;
GRANT RECOVER ANY TABLE TO nikhil
      *
ERROR at line 1:
ORA-00990: missing or invalid privilege


SQL> GRANT ANALYZE ANY TABLE TO nikhil;
GRANT ANALYZE ANY TABLE TO nikhil
      *
ERROR at line 1:
ORA-00990: missing or invalid privilege


SQL> GRANT ANALYZE ANY TO nikhil;

Grant succeeded.
```

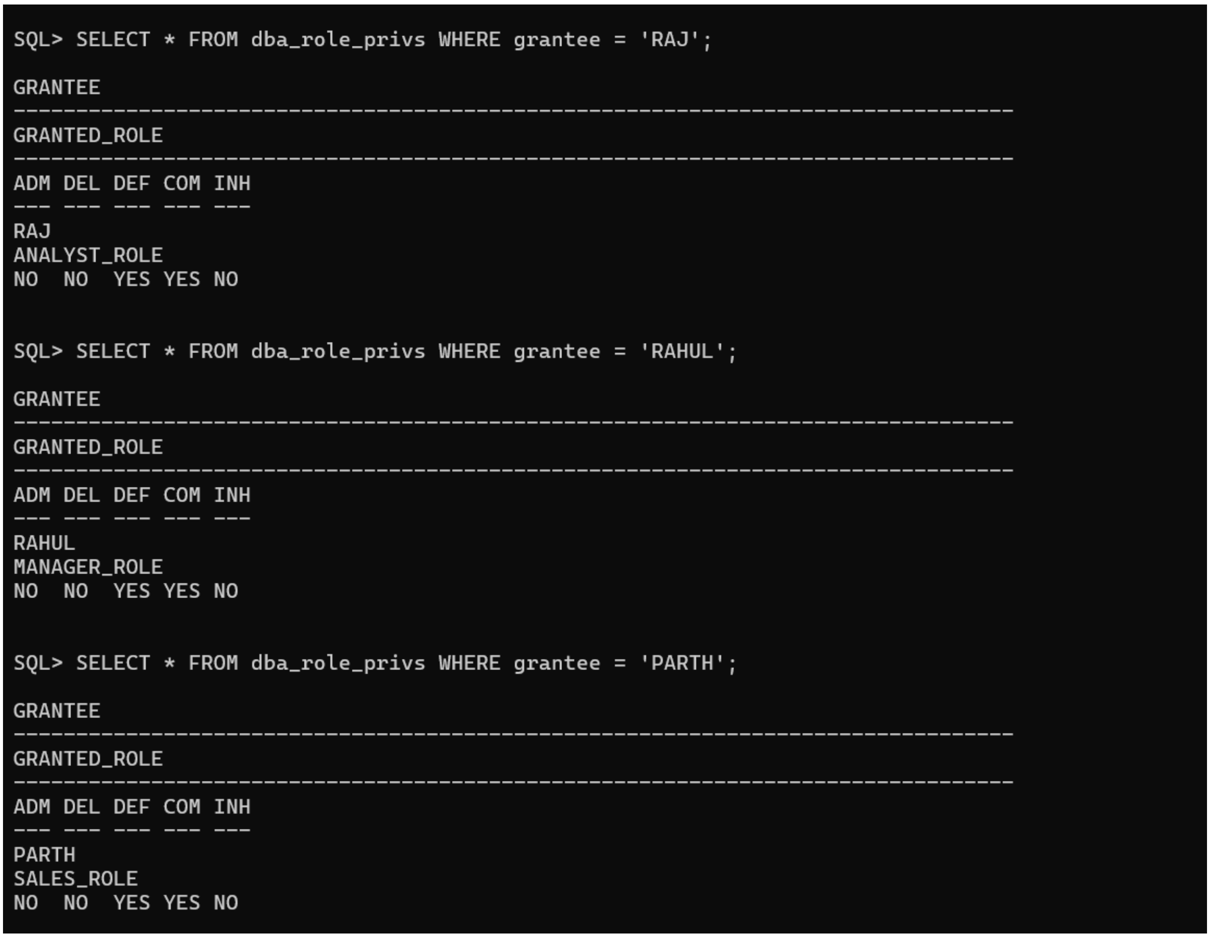


```
Enter user-name: '/' as sysdba

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> grant create session to parth;

Grant succeeded.

SQL> grant create session to rahul;

Grant succeeded.

SQL> grant create session to raj;

Grant succeeded.

SQL> grant create session to thakkar;

Grant succeeded.

SQL> grant create session to nikhil;

Grant succeeded.
```

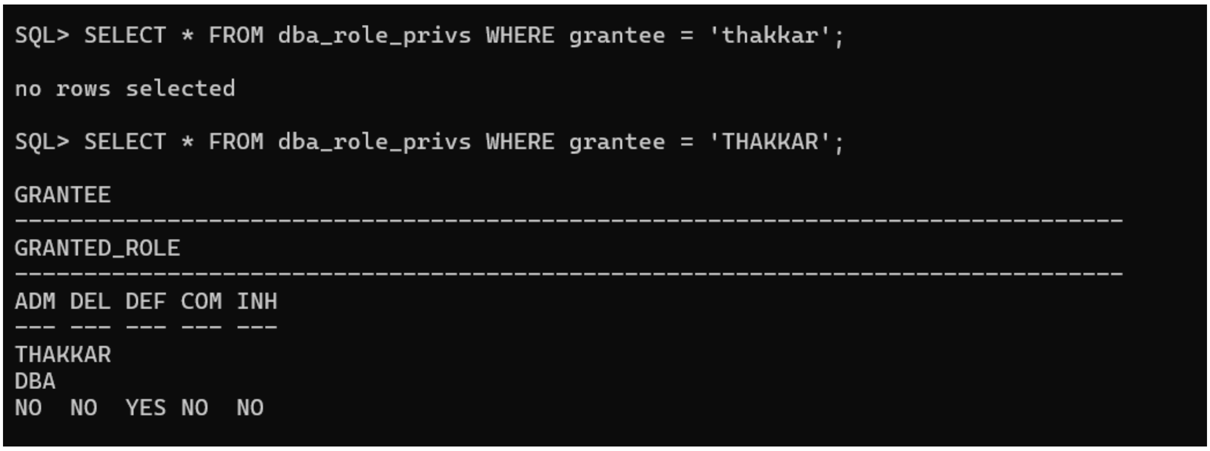
Connecting via a created user:



```
Enter user-name: thakkar
Enter password:

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

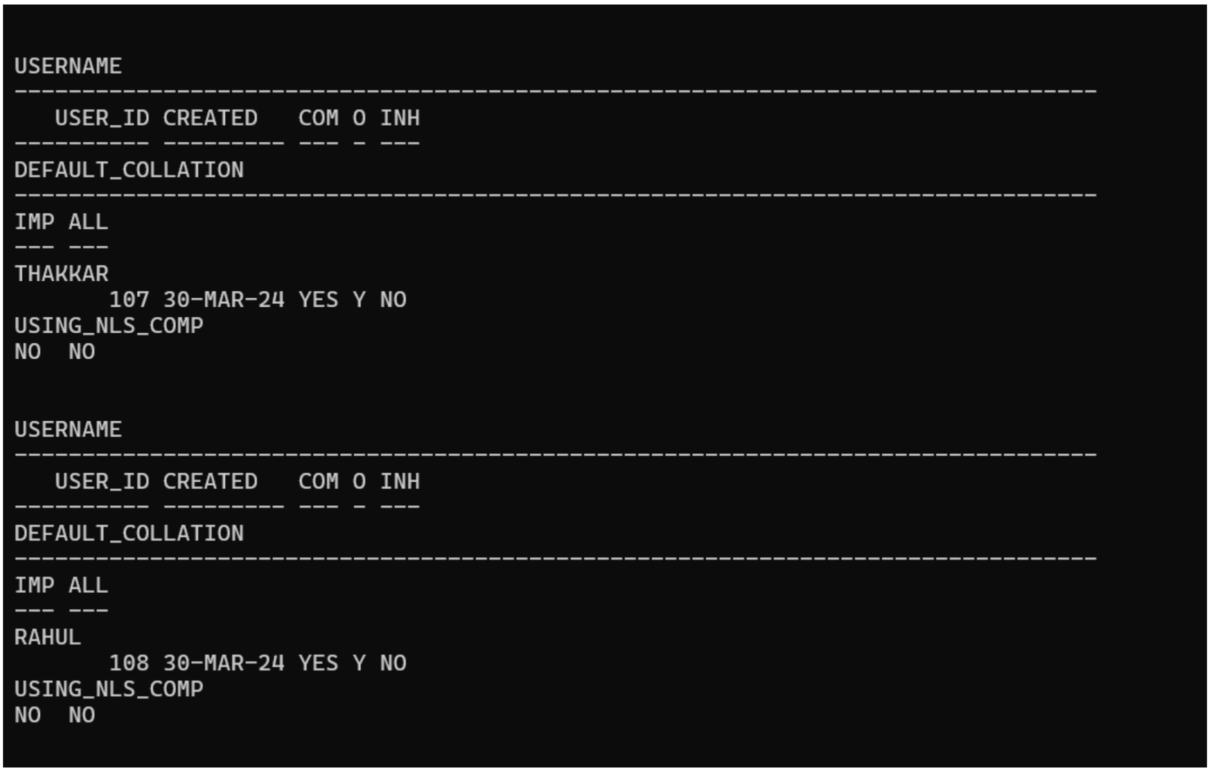


```
SQL> SELECT * FROM dba_role_privs WHERE grantee = 'RAJ';

GRANTEE
--------------------------------------------------------------------------------
GRANTED_ROLE
--------------------------------------------------------------------------------
ADM DEL DEF COM INH
--- --- --- --- ---
RAJ
ANALYST_ROLE
NO  NO  YES YES NO


SQL> SELECT * FROM dba_role_privs WHERE grantee = 'RAHUL';

GRANTEE
--------------------------------------------------------------------------------
GRANTED_ROLE
--------------------------------------------------------------------------------
ADM DEL DEF COM INH
--- --- --- --- ---
RAHUL
MANAGER_ROLE
NO  NO  YES YES NO


SQL> SELECT * FROM dba_role_privs WHERE grantee = 'PARTH';

GRANTEE
--------------------------------------------------------------------------------
GRANTED_ROLE
--------------------------------------------------------------------------------
ADM DEL DEF COM INH
--- --- --- --- ---
PARTH
SALES_ROLE
NO  NO  YES YES NO
```

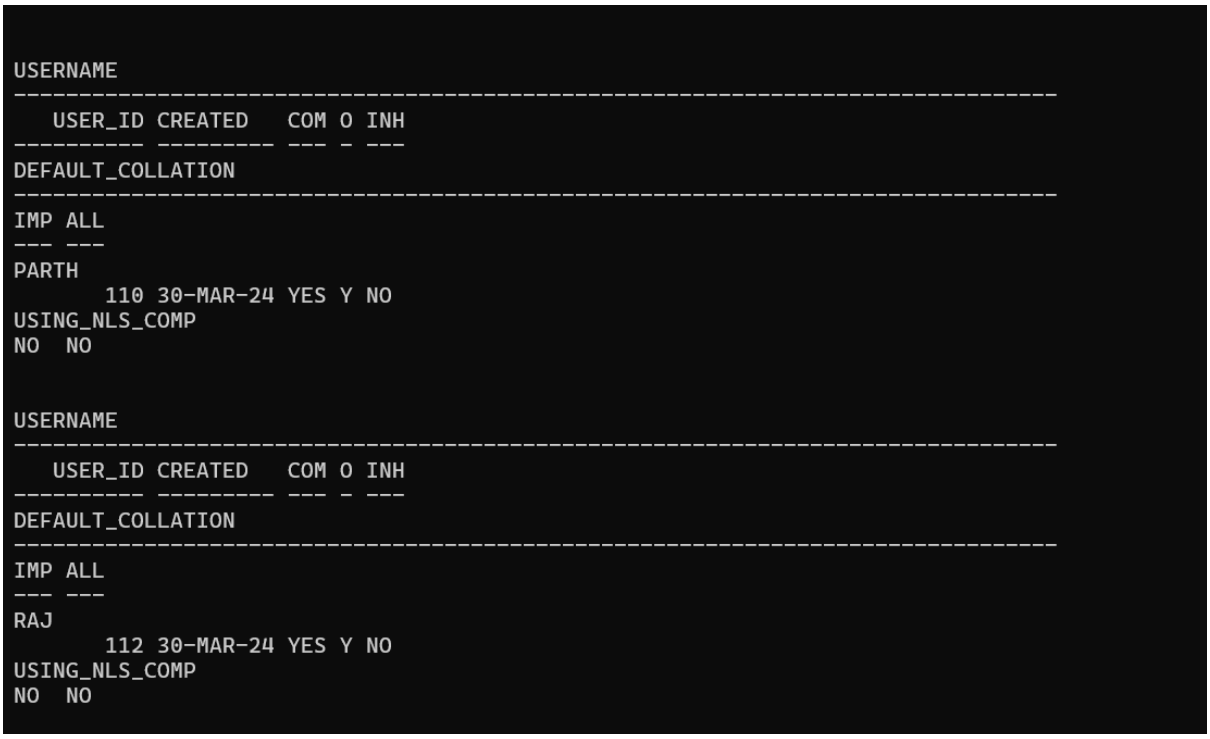


```
SQL> SELECT * FROM dba_role_privs WHERE grantee = 'thakkar';

no rows selected

SQL> SELECT * FROM dba_role_privs WHERE grantee = 'THAKKAR';

GRANTEE
--------------------------------------------------------------------------------
GRANTED_ROLE
--------------------------------------------------------------------------------
ADM DEL DEF COM INH
--- --- --- --- ---
THAKKAR
DBA
NO  NO  YES NO  NO
```

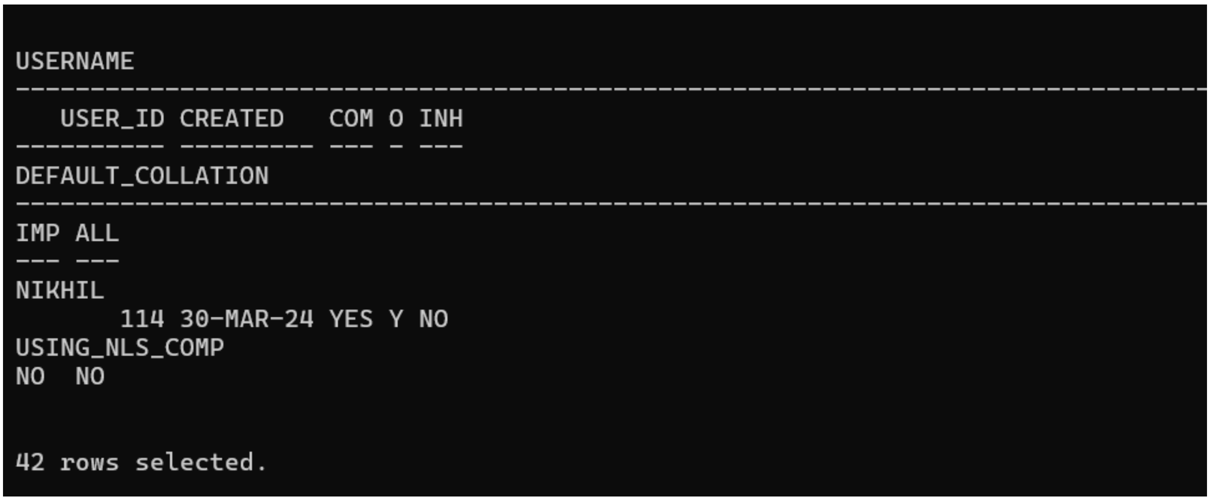
Select * from all_users;



```
USERNAME
--------------------------------------------------------------------------------
   USER_ID CREATED   COM O INH
---------- --------- --- - ---
DEFAULT_COLLATION
--------------------------------------------------------------------------------
IMP ALL
--- ---
THAKKAR
       107 30-MAR-24 YES Y NO
USING_NLS_COMP
NO  NO


USERNAME
--------------------------------------------------------------------------------
   USER_ID CREATED   COM O INH
---------- --------- --- - ---
DEFAULT_COLLATION
--------------------------------------------------------------------------------
IMP ALL
--- ---
RAHUL
       108 30-MAR-24 YES Y NO
USING_NLS_COMP
NO  NO
```



```
USERNAME
--------------------------------------------------------------------------------
   USER_ID CREATED   COM O INH
---------- --------- --- - ---
DEFAULT_COLLATION
--------------------------------------------------------------------------------
IMP ALL
--- ---
PARTH
       110 30-MAR-24 YES Y NO
USING_NLS_COMP
NO  NO


USERNAME
--------------------------------------------------------------------------------
   USER_ID CREATED   COM O INH
---------- --------- --- - ---
DEFAULT_COLLATION
--------------------------------------------------------------------------------
IMP ALL
--- ---
RAJ
       112 30-MAR-24 YES Y NO
USING_NLS_COMP
NO  NO
```



```
USERNAME
--------------------------------------------------------------------------------
   USER_ID CREATED   COM O INH
---------- --------- --- - ---
DEFAULT_COLLATION
--------------------------------------------------------------------------------
IMP ALL
--- ---
NIKHIL
       114 30-MAR-24 YES Y NO
USING_NLS_COMP
NO  NO


42 rows selected.
```

PASSWORD POLICY:

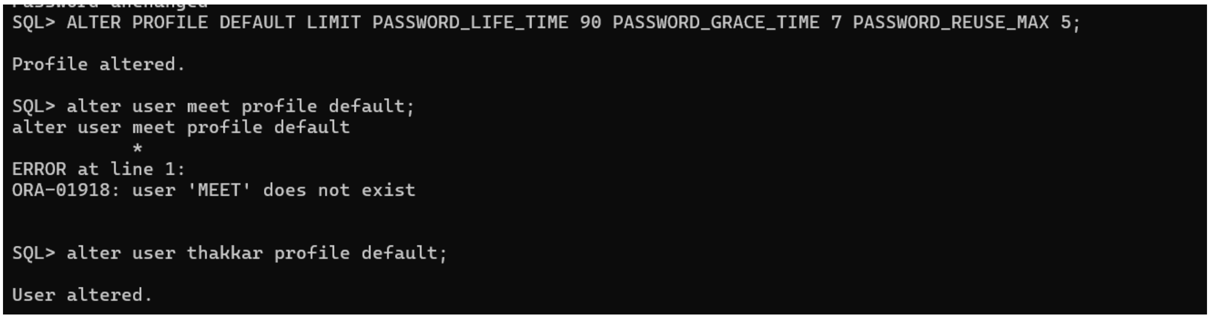
Admin User: [MEET]

Password Life Time: 90 days

Grace Period: 7 days

Reuse Max: 5

This user likely has high-level access and responsibilities, so a relatively shorter password life time and reuse limit are set to ensure frequent password changes and minimize the risk of compromised credentials.



```
SQL> ALTER PROFILE DEFAULT LIMIT PASSWORD_LIFE_TIME 90 PASSWORD_GRACE_TIME 7 PASSWORD_REUSE_MAX 5;

Profile altered.

SQL> alter user meet profile default;
alter user meet profile default
           *
ERROR at line 1:
ORA-01918: user 'MEET' does not exist


SQL> alter user thakkar profile default;

User altered.
```

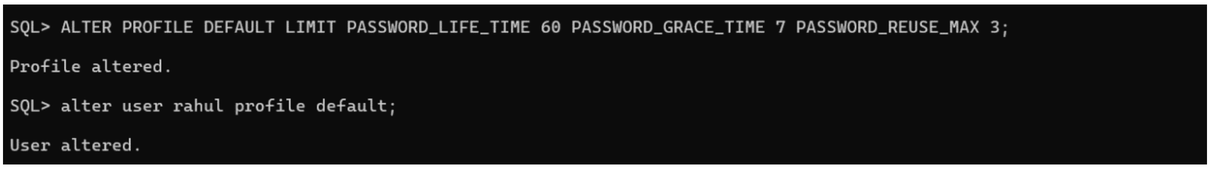
Manager User: [RAHUL]

Password Life Time: 60 days

Grace Period: 7 days

Reuse Max: 3

Managers typically have access to sensitive data and functions, but their password policies may not need to be as stringent as those for administrators. A shorter password life time and moderate reuse limit are set for security.



```
SQL> ALTER PROFILE DEFAULT LIMIT PASSWORD_LIFE_TIME 60 PASSWORD_GRACE_TIME 7 PASSWORD_REUSE_MAX 3;

Profile altered.

SQL> alter user rahul profile default;

User altered.
```

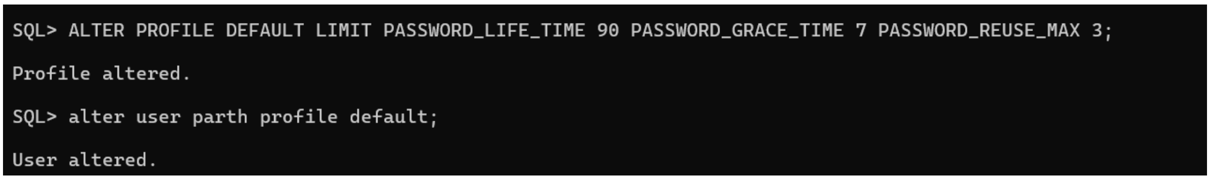
Sales User: [PARTH]

Password Life Time: 90 days

Grace Period: 7 days

Reuse Max: 3

Sales users often handle customer information and transactions. Their password policy is similar to the Manager's, balancing security with usability.



```
SQL> ALTER PROFILE DEFAULT LIMIT PASSWORD_LIFE_TIME 90 PASSWORD_GRACE_TIME 7 PASSWORD_REUSE_MAX 3;

Profile altered.

SQL> alter user parth profile default;

User altered.
```

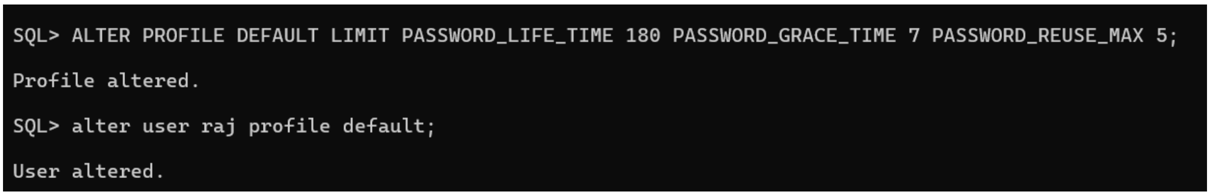
Data Analyst User: [RAJ]

Password Life Time: 180 days

Grace Period: 7 days

Reuse Max: 5

Data analysts typically require less frequent password changes, but still, need strong security. A longer password life time and higher reuse limit are set for convenience without compromising security.



```
SQL> ALTER PROFILE DEFAULT LIMIT PASSWORD_LIFE_TIME 180 PASSWORD_GRACE_TIME 7 PASSWORD_REUSE_MAX 5;

Profile altered.

SQL> alter user raj profile default;

User altered.
```

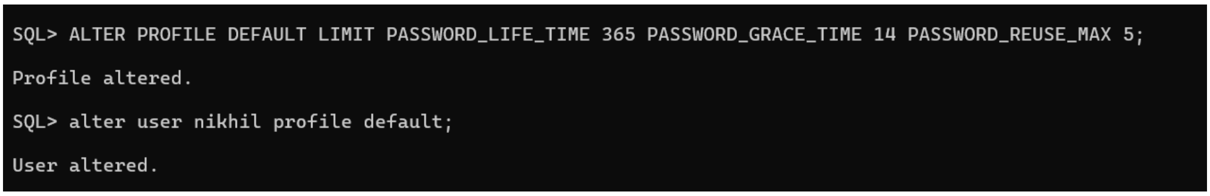
Recovery and Maintenance User: [NIKHIL]

Password Life Time: 365 days

Grace Period: 14 days

Reuse Max: 5

Users responsible for maintenance and recovery tasks may have longer password life times to avoid disruptions during critical operations. The grace period and reuse limit are also set to balance security and operational needs.



```
SQL> ALTER PROFILE DEFAULT LIMIT PASSWORD_LIFE_TIME 365 PASSWORD_GRACE_TIME 14 PASSWORD_REUSE_MAX 5;

Profile altered.

SQL> alter user nikhil profile default;

User altered.
```

COMMAND PARAMETERS DESCRIPTION:

PASSWORD_LIFE_TIME: Sets the password expiration time (UNLIMITED means passwords never expire).

PASSWORD_GRACE_TIME: Sets the grace period for password expiration (7 days in this example).

PASSWORD_REUSE_MAX: Sets the maximum number of password changes before a password can be reused.

Backup recovery policy

SAKILA DATASET BACKUP POLICY -

Daily Incremental Backup at 2 AM Local Time (IST):

Backup Type: Incremental backup.

Schedule: Daily at 2 AM local time (IST).

Retention Time: Retain incremental backups for 1 month.

Data Included: Transactional data (payment, rental, inventory) and audit/log data capturing changes made to the database schema, tables, and records.

Multiplexing: Multiplex backup files into at least 2 backup files located on separate disks.

This daily incremental backup captures changes made to transactional data like payment records, rental transactions, and inventory updates, ensuring that recent database modifications are protected. Additionally, audit and log data critical for monitoring database changes are also included.

Monthly Full Backup on the First Monday at 4 AM Local Time (IST):

Backup Type: Full backup.

Schedule: Monthly on the first Monday at 4 AM local time (IST).

Retention Time: Retain full backups for 6 months.

Data Included: Entire database backup, updating all tables/stores.

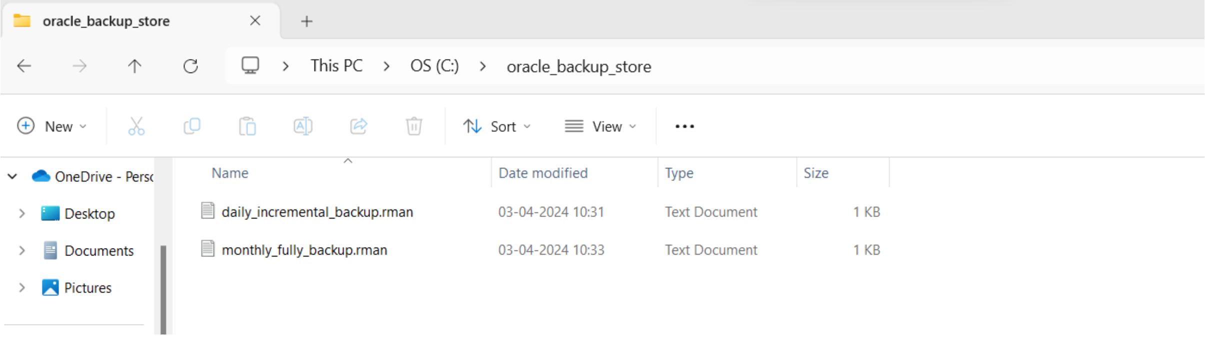
Multiplexing: Multiplex backup files into 4 backup files located on separate disks.

This monthly full backup ensures a comprehensive snapshot of the entire Sakila database, including all tables, schemas, and objects. By updating all stores and tables, it provides a complete backup for recovery purposes.

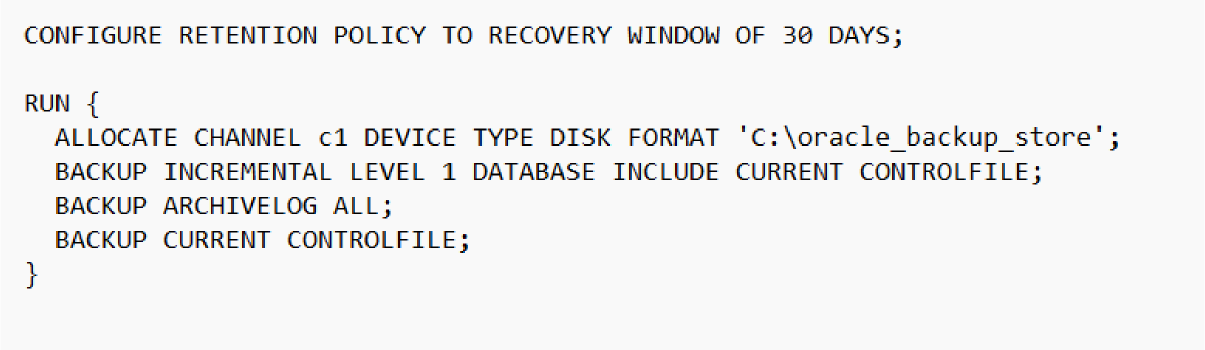
Time Zone: The time zone for backup schedules and retention times is set to Indian Standard Time (IST).

Multiplexing of Backup Files: For full backups, multiplex backup files into at least 2 backup files located on separate disks to ensure redundancy and fault tolerance. For example, configure multiplexing to create 4 backup files spread across different disks.

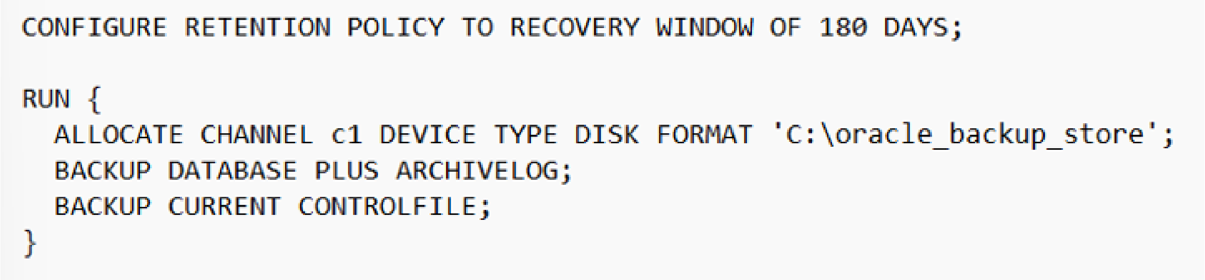
Creating scripts for daily and monthly backups.



Name	Date modified	Type	Size
daily_incremental_backup.rman	03-04-2024 10:31	Text Document	1 KB
monthly_fully_backup.rman	03-04-2024 10:33	Text Document	1 KB

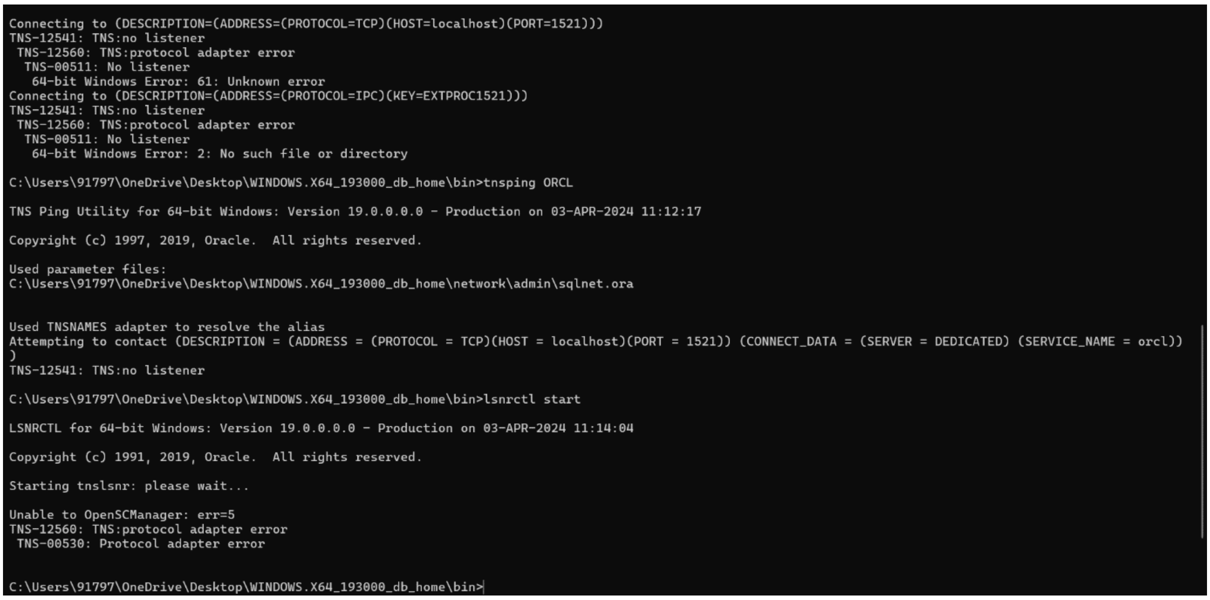


```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 30 DAYS;

RUN {
  ALLOCATE CHANNEL c1 DEVICE TYPE DISK FORMAT 'C:\oracle_backup_store';
  BACKUP INCREMENTAL LEVEL 1 DATABASE INCLUDE CURRENT CONTROLFILE;
  BACKUP ARCHIVELOG ALL;
  BACKUP CURRENT CONTROLFILE;
}
```

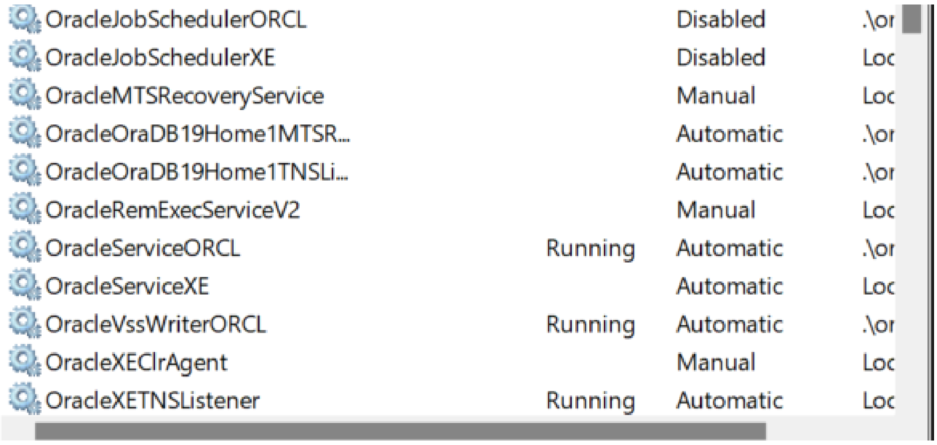


```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 180 DAYS;

RUN {
  ALLOCATE CHANNEL c1 DEVICE TYPE DISK FORMAT 'C:\oracle_backup_store';
  BACKUP DATABASE PLUS ARCHIVELOG;
  BACKUP CURRENT CONTROLFILE;
}
```

NOTE: While working with backup and recovery we faced with the issue of

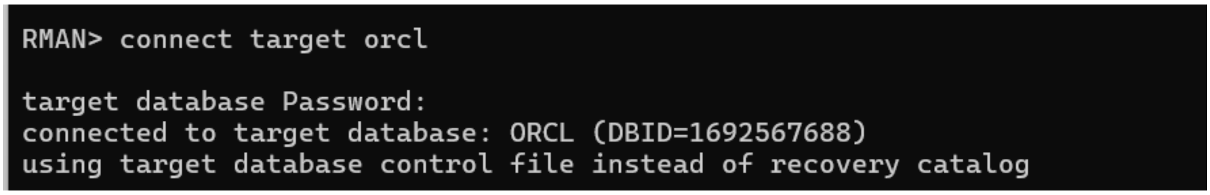


```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521)))
TNS-12541: TNS:no listener
 TNS-12560: TNS:protocol adapter error
  TNS-00511: No listener
   64-bit Windows Error: 61: Unknown error
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
TNS-12541: TNS:no listener
 TNS-12560: TNS:protocol adapter error
  TNS-00511: No listener
   64-bit Windows Error: 2: No such file or directory

C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin>tnsping ORCL

TNS Ping Utility for 64-bit Windows: Version 19.0.0.0.0 - Production on 03-APR-2024 11:12:17

Copyright (c) 1997, 2019, Oracle.  All rights reserved.

Used parameter files:
C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\network\admin\sqlnet.ora


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = orcl))
)
TNS-12541: TNS:no listener

C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin>lsnrctl start

LSNRCTL for 64-bit Windows: Version 19.0.0.0.0 - Production on 03-APR-2024 11:14:04

Copyright (c) 1991, 2019, Oracle.  All rights reserved.

Starting tnslsnr: please wait...

Unable to OpenSCManager: err=5
TNS-12560: TNS:protocol adapter error
 TNS-00530: Protocol adapter error


C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin>
```



OracleJobSchedulerORCL		Disabled	.\or
OracleJobSchedulerXE		Disabled	Loc
OracleMTSRecoveryService		Manual	Loc
OracleOraDB19Home1MTSR...		Automatic	.\or
OracleOraDB19Home1TNSLi...		Automatic	.\or
OracleRemExecServiceV2		Manual	Loc
OracleServiceORCL	Running	Automatic	.\or
OracleServiceXE		Automatic	Loc
OracleVssWriterORCL	Running	Automatic	.\or
OracleXEClrAgent		Manual	Loc
OracleXETNSListener	Running	Automatic	Loc

As a dba we explored various documentation for hours and then solved it by going to services and then turning the TNS listener restart again.

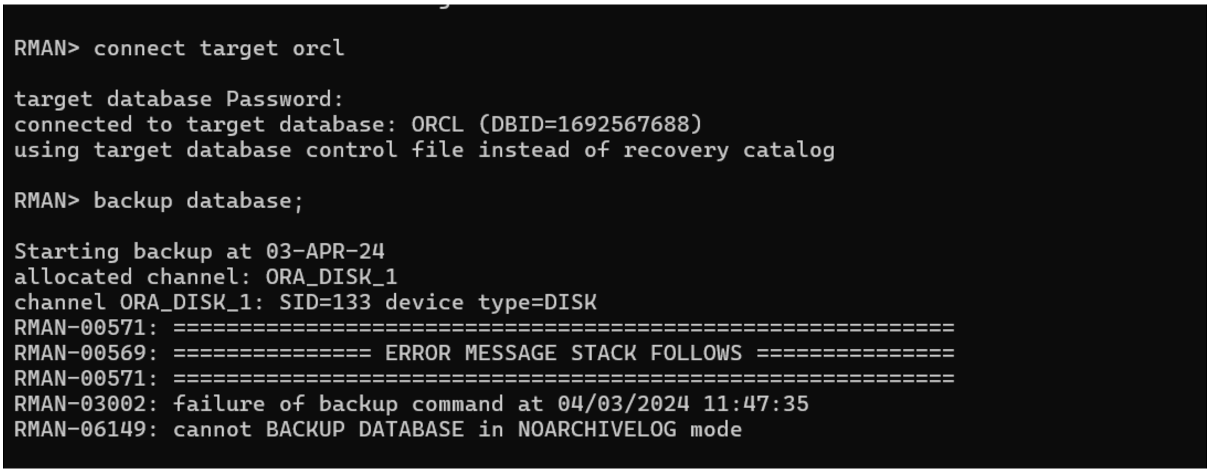


```
RMAN> connect target orcl

target database Password:
connected to target database: ORCL (DBID=1692567688)
using target database control file instead of recovery catalog
```

12345

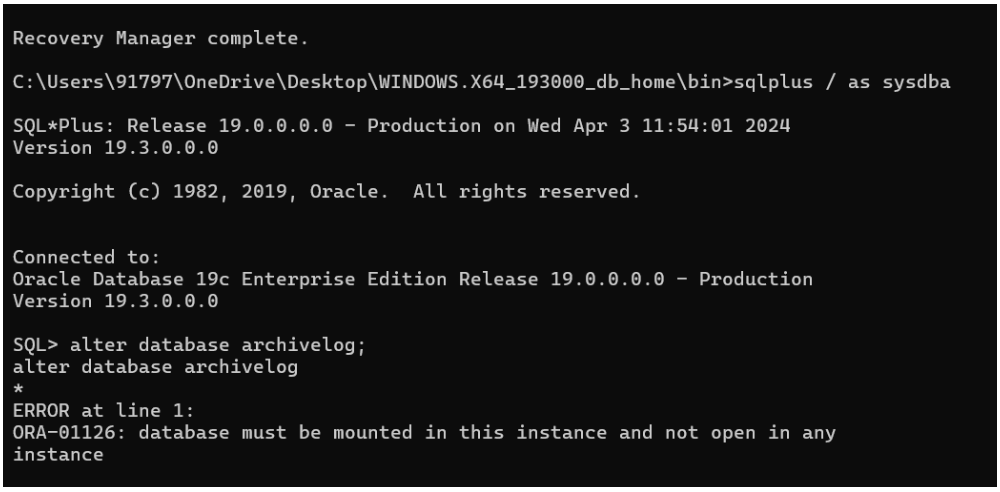
We did the above to connect to database via RMAN.

In order to take backup we had to connect to rman in noarchivelog mode for this we did the below

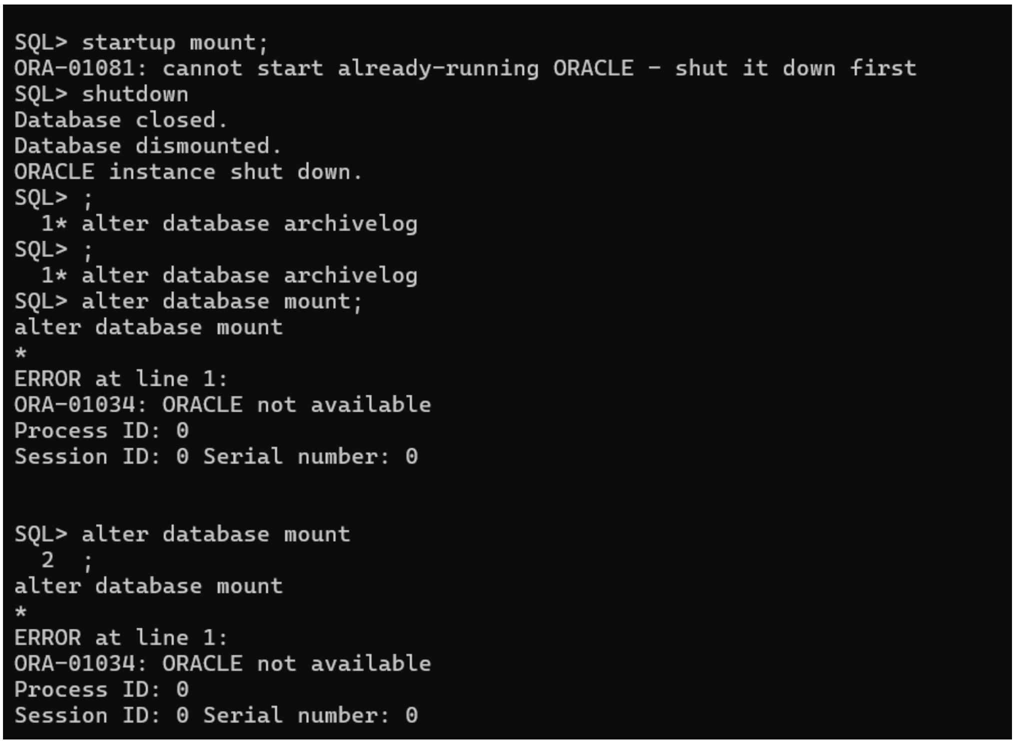


```
RMAN> connect target orcl

target database Password:
connected to target database: ORCL (DBID=1692567688)
using target database control file instead of recovery catalog

RMAN> backup database;

Starting backup at 03-APR-24
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=133 device type=DISK
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of backup command at 04/03/2024 11:47:35
RMAN-06149: cannot BACKUP DATABASE in NOARCHIVELOG mode
```

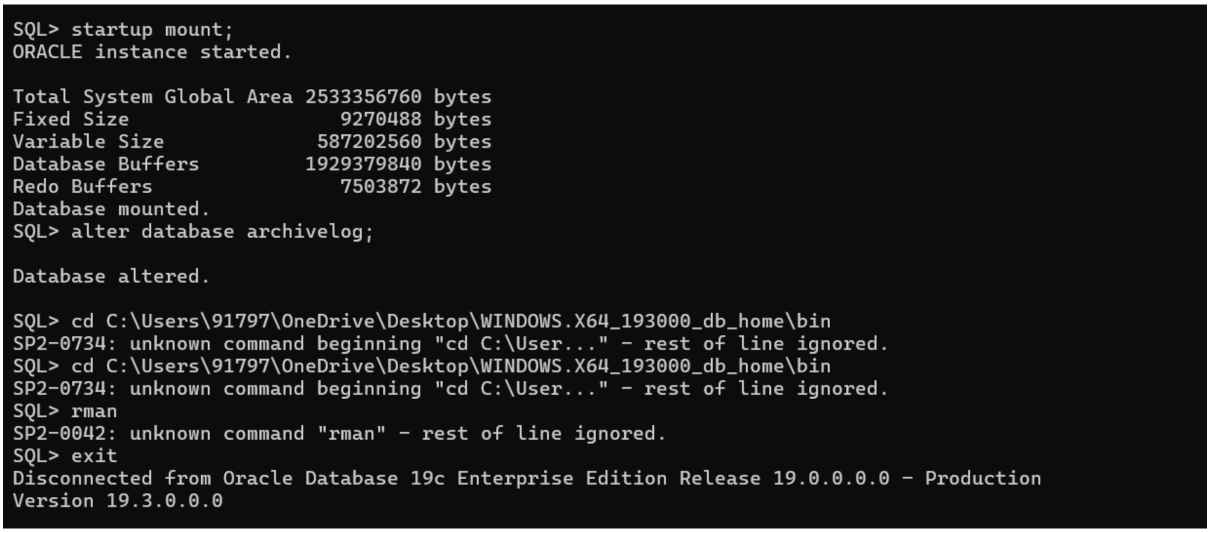
We didn’t know how that we had to connect to noarchivelog mode.



```
Recovery Manager complete.

C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin>sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Apr 3 11:54:01 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> alter database archivelog;
alter database archivelog
*
ERROR at line 1:
ORA-01126: database must be mounted in this instance and not open in any
instance
```

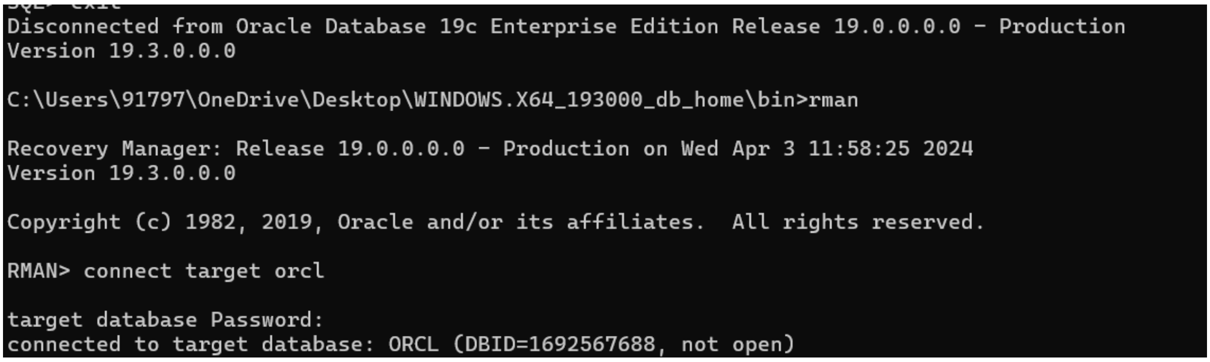


```
SQL> startup mount;
ORA-01081: cannot start already-running ORACLE - shut it down first
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ;
  1* alter database archivelog
SQL> ;
  1* alter database archivelog
SQL> alter database mount;
alter database mount
*
ERROR at line 1:
ORA-01034: ORACLE not available
Process ID: 0
Session ID: 0 Serial number: 0


SQL> alter database mount
  2  ;
alter database mount
*
ERROR at line 1:
ORA-01034: ORACLE not available
Process ID: 0
Session ID: 0 Serial number: 0
```

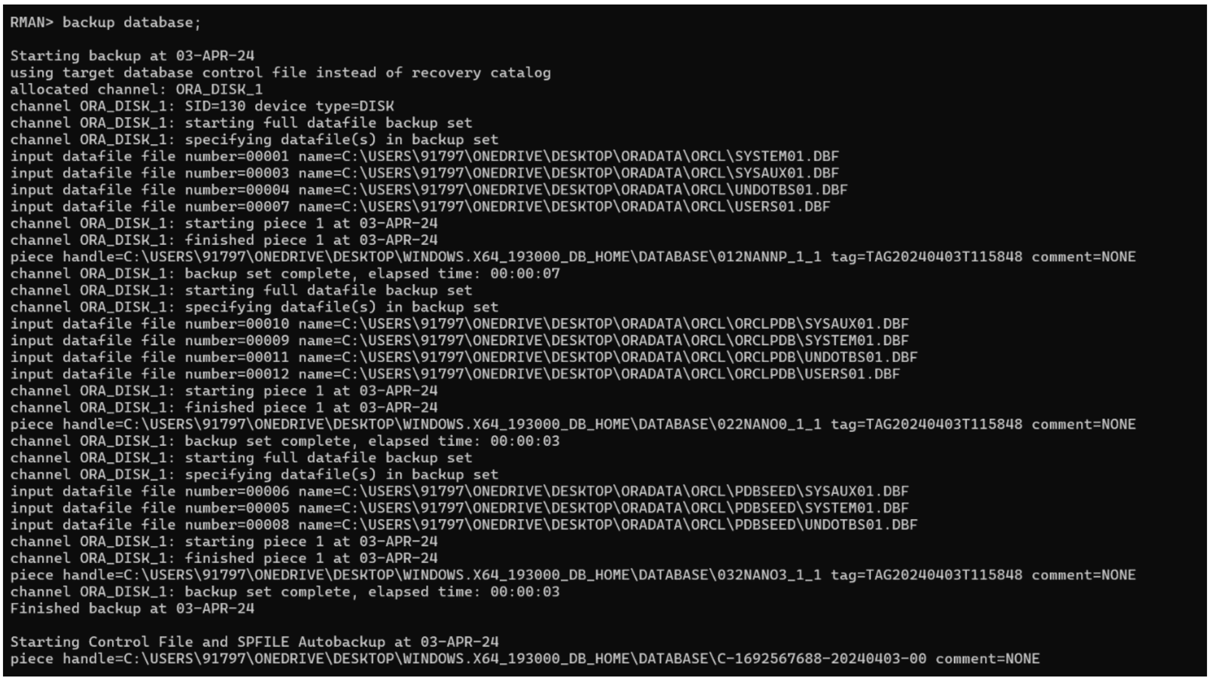


```
SQL> startup mount;
ORACLE instance started.

Total System Global Area 2533356760 bytes
Fixed Size                  9270488 bytes
Variable Size             587202560 bytes
Database Buffers         1929379840 bytes
Redo Buffers                7503872 bytes
Database mounted.
SQL> alter database archivelog;

Database altered.

SQL> cd C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin
SP2-0734: unknown command beginning "cd C:\User..." - rest of line ignored.
SQL> cd C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin
SP2-0734: unknown command beginning "cd C:\User..." - rest of line ignored.
SQL> rman
SP2-0042: unknown command "rman" - rest of line ignored.
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

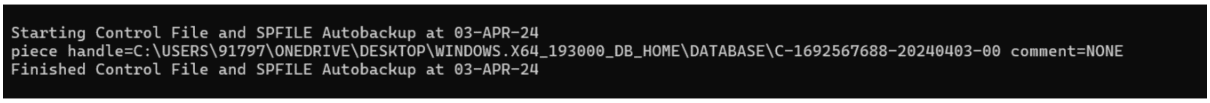
So we connect as sys user and then mounted the database and changed it to noarchivelog mode then we had to login via rman.



```
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin>rman

Recovery Manager: Release 19.0.0.0.0 - Production on Wed Apr 3 11:58:25 2024
Version 19.3.0.0.0

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

RMAN> connect target orcl

target database Password:
connected to target database: ORCL (DBID=1692567688, not open)
```

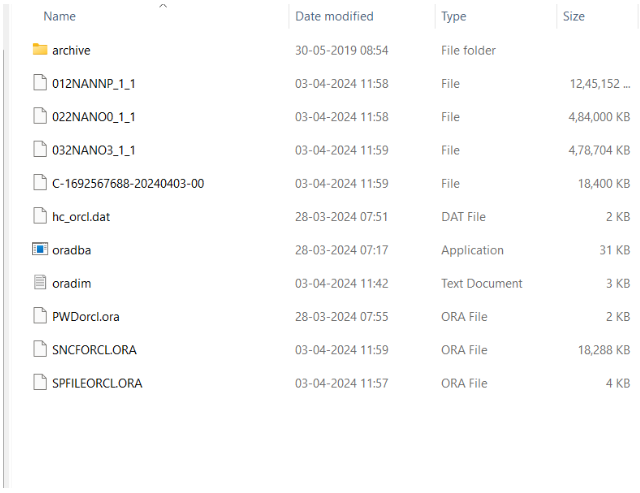


```
RMAN> backup database;

Starting backup at 03-APR-24
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=130 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\SYSTEM01.DBF
input datafile file number=00003 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\SYSAUX01.DBF
input datafile file number=00004 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\UNDOTBS01.DBF
input datafile file number=00007 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 03-APR-24
channel ORA_DISK_1: finished piece 1 at 03-APR-24
piece handle=C:\USERS\91797\ONEDRIVE\DESKTOP\WINDOWS.X64_193000_DB_HOME\DATABASE\012NANNP_1_1 tag=TAG20240403T115848 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00010 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\ORCLPDB\SYSAUX01.DBF
input datafile file number=00009 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\ORCLPDB\SYSTEM01.DBF
input datafile file number=00011 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\ORCLPDB\UNDOTBS01.DBF
input datafile file number=00012 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\ORCLPDB\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 03-APR-24
channel ORA_DISK_1: finished piece 1 at 03-APR-24
piece handle=C:\USERS\91797\ONEDRIVE\DESKTOP\WINDOWS.X64_193000_DB_HOME\DATABASE\022NANO0_1_1 tag=TAG20240403T115848 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\PDBSEED\SYSAUX01.DBF
input datafile file number=00005 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\PDBSEED\SYSTEM01.DBF
input datafile file number=00008 name=C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\PDBSEED\UNDOTBS01.DBF
channel ORA_DISK_1: starting piece 1 at 03-APR-24
channel ORA_DISK_1: finished piece 1 at 03-APR-24
piece handle=C:\USERS\91797\ONEDRIVE\DESKTOP\WINDOWS.X64_193000_DB_HOME\DATABASE\032NANO3_1_1 tag=TAG20240403T115848 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 03-APR-24

Starting Control File and SPFILE Autobackup at 03-APR-24
piece handle=C:\USERS\91797\ONEDRIVE\DESKTOP\WINDOWS.X64_193000_DB_HOME\DATABASE\C-1692567688-20240403-00 comment=NONE
```

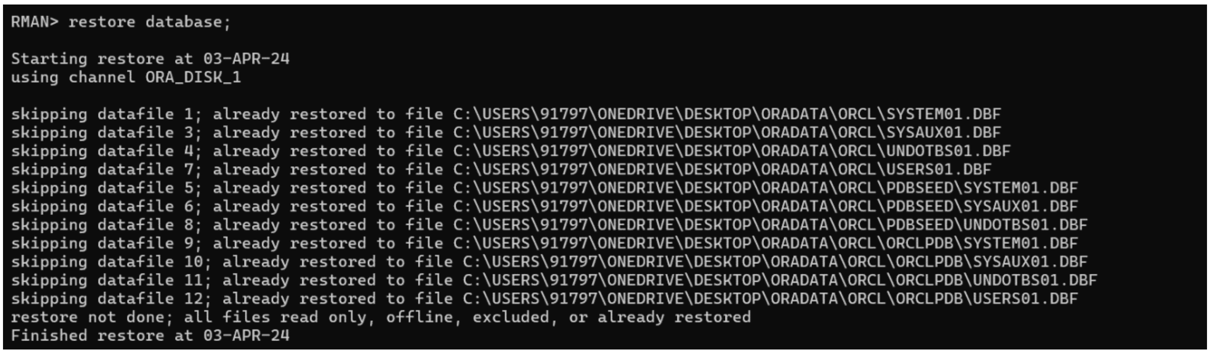


```
Starting Control File and SPFILE Autobackup at 03-APR-24
piece handle=C:\USERS\91797\ONEDRIVE\DESKTOP\WINDOWS.X64_193000_DB_HOME\DATABASE\C-1692567688-20240403-00 comment=NONE
Finished Control File and SPFILE Autobackup at 03-APR-24
```

Verifying if backup took place.



Name	Date modified	Type	Size
archive	30-05-2019 08:54	File folder	
012NANNP_1_1	03-04-2024 11:58	File	12,45,152 ...
022NANO0_1_1	03-04-2024 11:58	File	4,84,000 KB
032NANO3_1_1	03-04-2024 11:59	File	4,78,704 KB
C-1692567688-20240403-00	03-04-2024 11:59	File	18,400 KB
hc_orcl.dat	28-03-2024 07:51	DAT File	2 KB
oradba	28-03-2024 07:17	Application	31 KB
oradim	03-04-2024 11:42	Text Document	3 KB
PWDorcl.ora	28-03-2024 07:55	ORA File	2 KB
SNCFORCL.ORA	03-04-2024 11:59	ORA File	18,288 KB
SPFILEORCL.ORA	03-04-2024 11:57	ORA File	4 KB

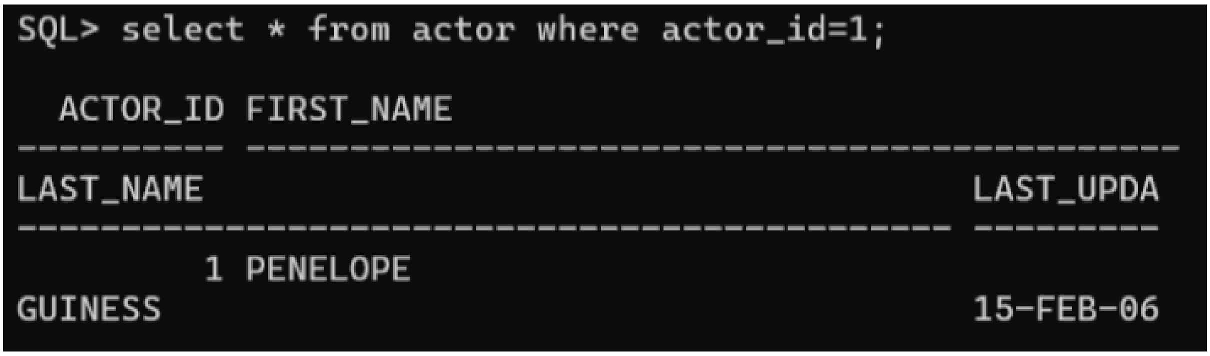


```
RMAN> restore database;

Starting restore at 03-APR-24
using channel ORA_DISK_1

skipping datafile 1; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\SYSTEM01.DBF
skipping datafile 3; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\SYSAUX01.DBF
skipping datafile 4; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\UNDOTBS01.DBF
skipping datafile 7; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\USERS01.DBF
skipping datafile 5; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\PDBSEED\SYSTEM01.DBF
skipping datafile 6; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\PDBSEED\SYSAUX01.DBF
skipping datafile 8; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\PDBSEED\UNDOTBS01.DBF
skipping datafile 9; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\ORCLPDB\SYSTEM01.DBF
skipping datafile 10; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\ORCLPDB\SYSAUX01.DBF
skipping datafile 11; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\ORCLPDB\UNDOTBS01.DBF
skipping datafile 12; already restored to file C:\USERS\91797\ONEDRIVE\DESKTOP\ORADATA\ORCL\ORCLPDB\USERS01.DBF
restore not done; all files read only, offline, excluded, or already restored
Finished restore at 03-APR-24
```

RECOVERY SCNEARIO:

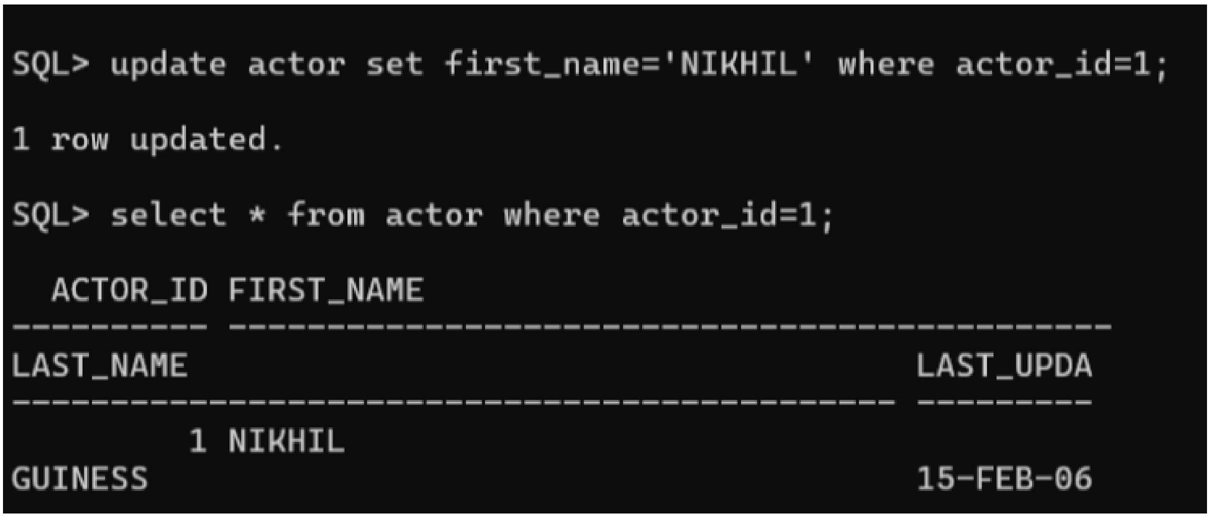
Let’s assume a simple scenario in which the actors table first row that is actor_id with value equals to 1 is altered so that his/her is changed to Nikhil but it is abruptly shutdown without commiting. Our job as the DBA is to recover the database to the last commited stage.

Step 1: checking the value of actor_id=1, name before alteration



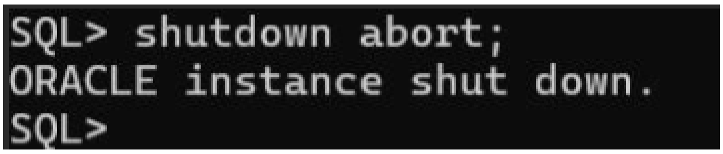
```
SQL> select * from actor where actor_id=1;

  ACTOR_ID FIRST_NAME
---------- ---------------------------------------------
LAST_NAME                                     LAST_UPDA
--------------------------------------------- ---------
         1 PENELOPE
GUINESS                                       15-FEB-06
```

Step 2: Updating the value to Nikhil.



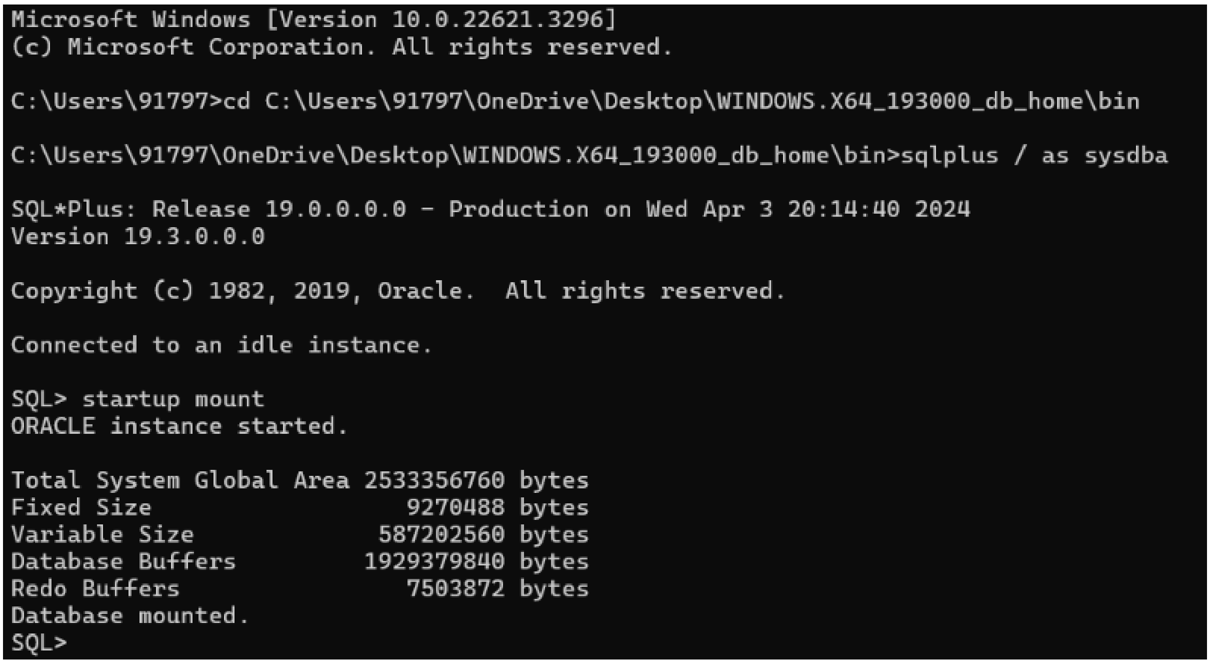
```
SQL> update actor set first_name='NIKHIL' where actor_id=1;

1 row updated.

SQL> select * from actor where actor_id=1;

  ACTOR_ID FIRST_NAME
---------- ---------------------------------------------
LAST_NAME                                     LAST_UPDA
--------------------------------------------- ---------
         1 NIKHIL
GUINESS                                       15-FEB-06
```

Step 3: Shutting down the database without commiting



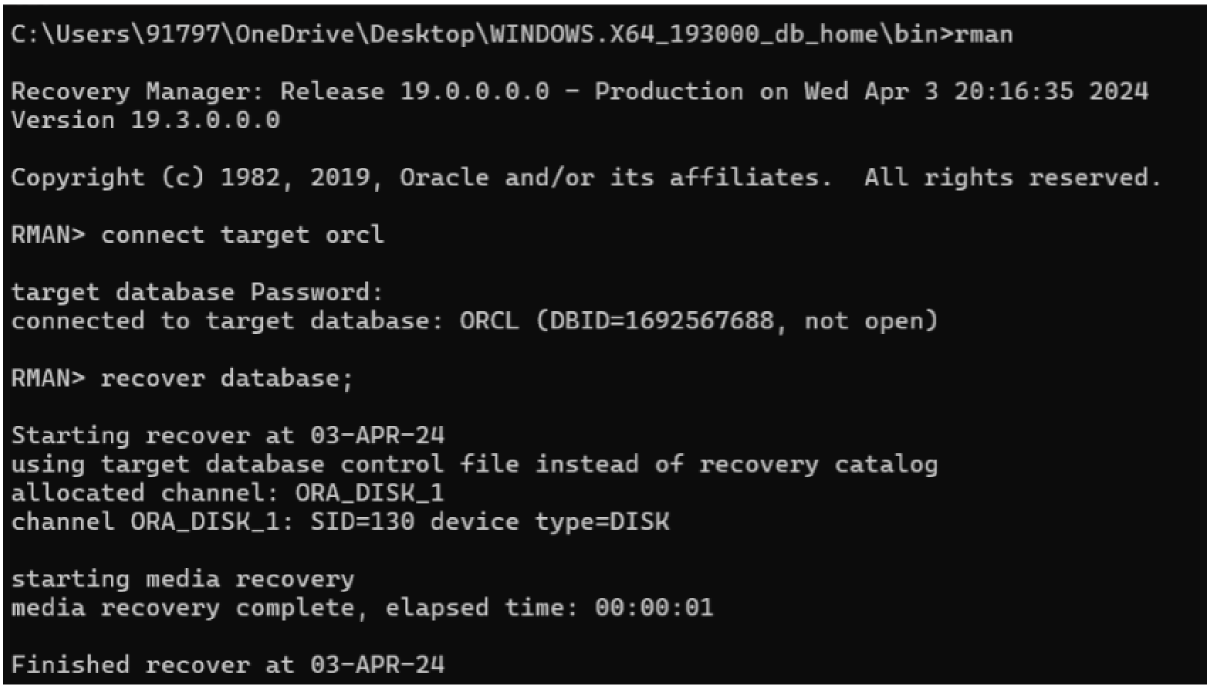
```
SQL> shutdown abort;
ORACLE instance shut down.
SQL>
```

Step 4: In the cmd, logging as dba and starting up in mount. This means the database will be open, but it won't be accessible for transactions.



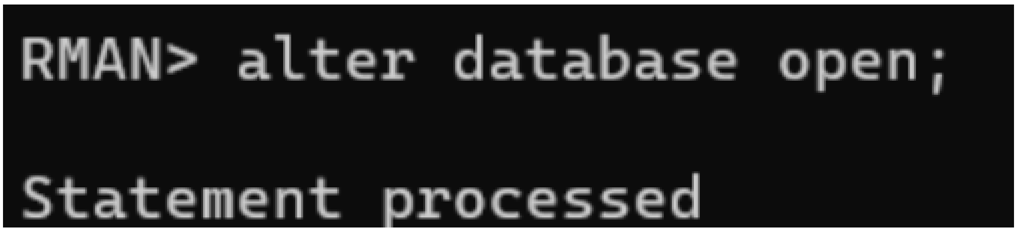
```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91797>cd C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin

C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin>sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Apr 3 20:14:40 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup mount
ORACLE instance started.

Total System Global Area 2533356760 bytes
Fixed Size                  9270488 bytes
Variable Size             587202560 bytes
Database Buffers         1929379840 bytes
Redo Buffers                7503872 bytes
Database mounted.
SQL>
```

Step 5: Now connecting as RMAN, and recovering the database.



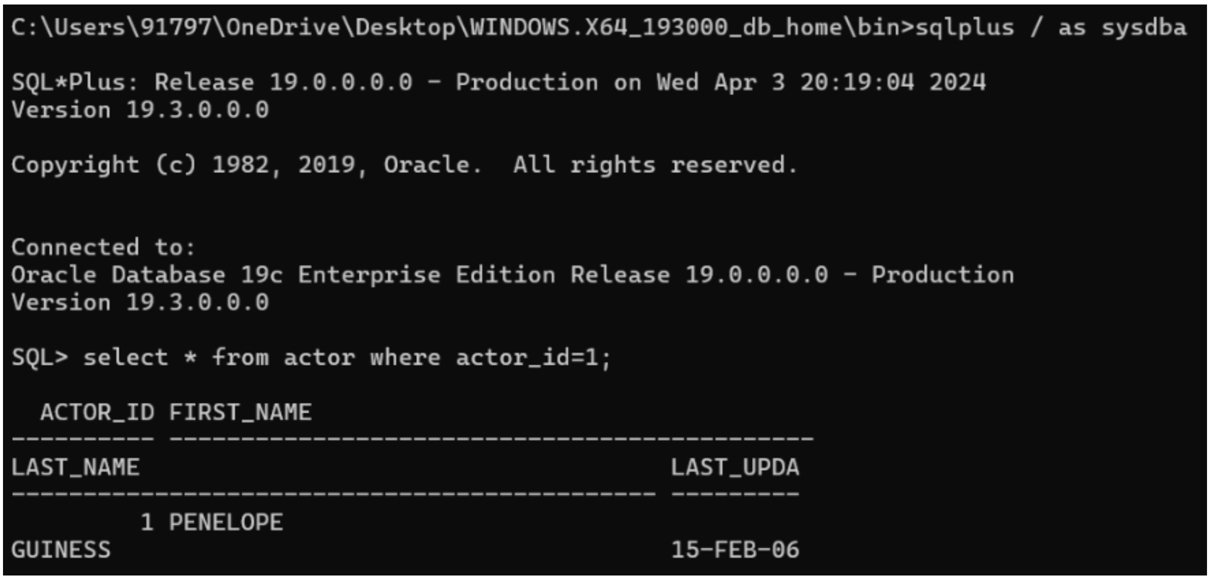
```
C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin>rman

Recovery Manager: Release 19.0.0.0.0 - Production on Wed Apr 3 20:16:35 2024
Version 19.3.0.0.0

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

RMAN> connect target orcl

target database Password:
connected to target database: ORCL (DBID=1692567688, not open)

RMAN> recover database;

Starting recover at 03-APR-24
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=130 device type=DISK

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 03-APR-24
```

Step 6: Starting the database in open mode, After the recovery completes, open the database to make it accessible for users.



```
RMAN> alter database open;

Statement processed
```

Step 7: In SqlPlus logging in as dba again and checking if the database is recovered successfully.



```
C:\Users\91797\OneDrive\Desktop\WINDOWS.X64_193000_db_home\bin>sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Apr 3 20:19:04 2024
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select * from actor where actor_id=1;

  ACTOR_ID FIRST_NAME
---------- ---------------------------------------------
LAST_NAME                                     LAST_UPDA
--------------------------------------------- ---------
         1 PENELOPE
GUINESS                                       15-FEB-06
```

In the above screenshot we can see that the database is successfully recovered to the old stage that is last committed.

CONCLUSION: Thus, we have successfully implemented all the vital DBA tasks ranging from creating users all the way to recovering our *SAKILA* database.