Ex no:	8- Creating of Other Database Objects
Date	12-03-24

Aim:

To manage data within a database, utilizing various database objects to facilate data storage, retrieval, manipulation, security.

Description:

Database objects plays a crucial role in structuring and optimizing the performance of a database system these objects include tables, views, index, sequences, and synonyms.

1. Table- Basic unit of storage; composed of rows and columns.
2. View- Logically represents subsets of data from one or more tables.
3. Sequence- Generates primary key values.
4. Index- Improves the performance of some queries.
5. Synonym- Alternative name for an object

View

You embed a subquery within the CREATE VIEW statement.

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view[(alias[, alias]...)]

 AS subquery

[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

Removing a View

```
DROP VIEW view_name;
```

Inline Views

An inline view is a subquery with an alias (or correlation name) that you can use within a SQL statement.

Top-N Analysis

The high-level structure of a Top-N analysis query is:

```
SELECT ROWNUM as RANK, last_name, salary FROM (SELECT last_name,salary FROM employees
ORDER BY salary DESC) WHERE ROWNUM <= 3;
```

Sequence

Define a sequence to generate sequential numbers automatically:

```
CREATE SEQUENCE sequence
            [INCREMENT BY n]
            [START WITH n]
            [{MAXVALUE n | NOMAXVALUE}]
            [{MINVALUE n | NOMINVALUE}]
          [{CYCLE | NOCYCLE}]      [{CACHE n | NOCACHE}];
```

Insert:

```
INSERT INTO departments(department_id, department_name, location_id)

  VALUES (dept_deptid_seq.NEXTVAL,'Support', 2500);
```

Removing a Sequence

```
DROP SEQUENCE dept_deptid_seq;
```

Index

Automatically: A unique index is created automatically when you define a PRIMARY KEY or UNIQUE constraint in a table definition.

Manually: Users can create nonunique indexes on columns to speed up access to the rows.

```
CREATE INDEX index ON table (column[, column]...);
```

Removing an Index

```
DROP INDEX index;
```

Synonyms

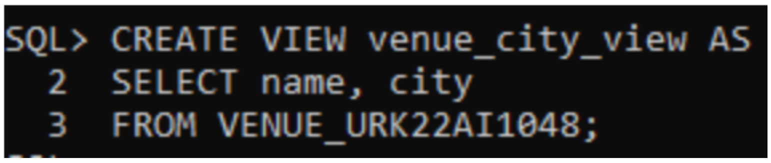
```
CREATE SYNONYM d_sum FOR dept_sum_vu;
```

Removing a synonym

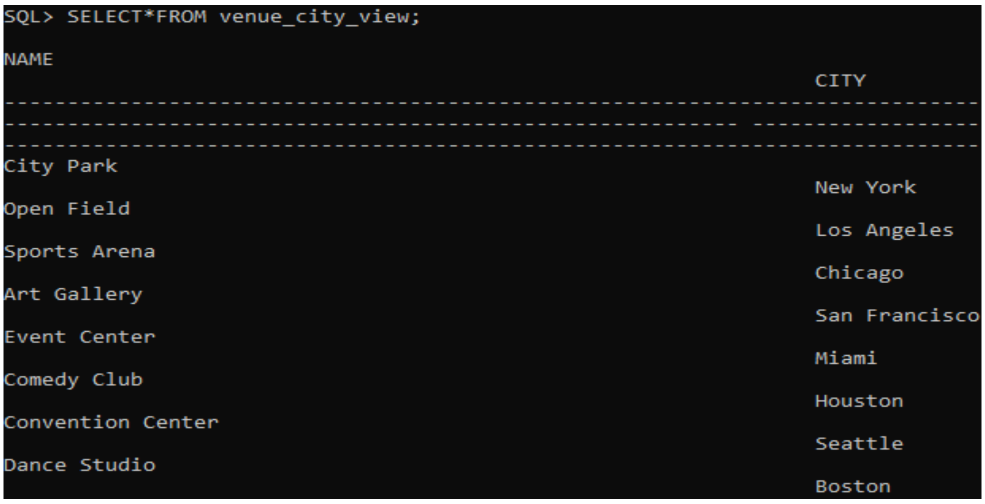
```
DROP SYNONYM sy_name;
```

Questions

1. Design a view that present a list of venues along with the cities.

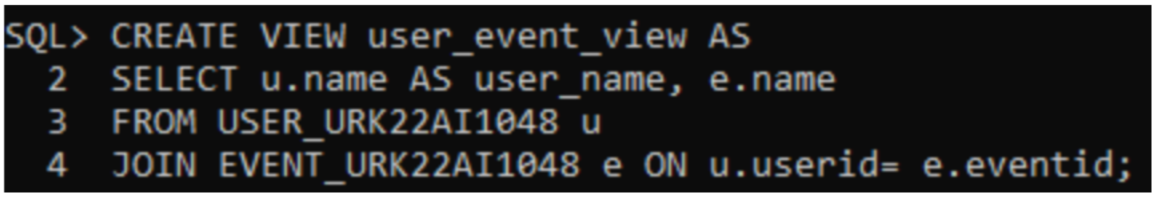


```
SQL> CREATE VIEW venue_city_view AS
     SELECT name, city
     FROM VENUE_URK22AI1048;
```

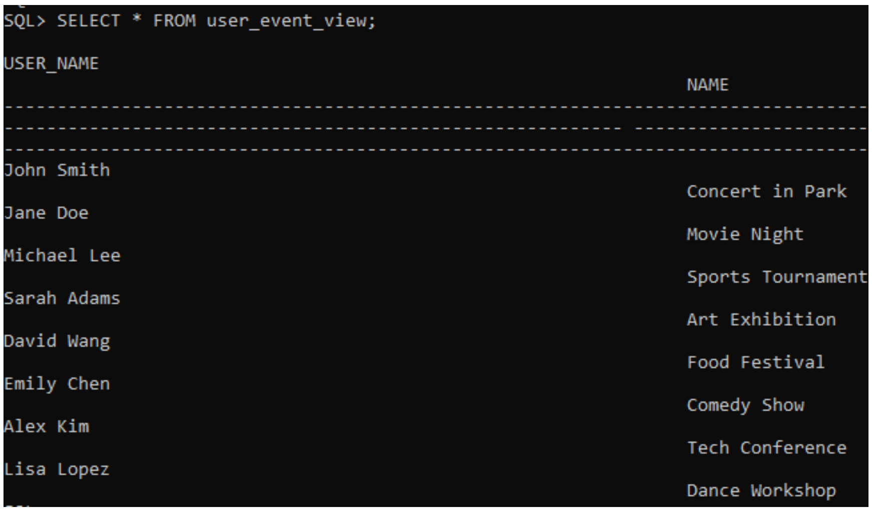


```
SQL> SELECT*FROM venue_city_view;

NAME
                                                                 CITY
-----------------------------------------------------------------------------
----------------------------------------------------------- -----------------
-----------------------------------------------------------------------------
City Park
                                                                 New York
Open Field
                                                                 Los Angeles
Sports Arena
                                                                 Chicago
Art Gallery
                                                                 San Francisco
Event Center
                                                                 Miami
Comedy Club
                                                                 Houston
Convention Center
                                                                 Seattle
Dance Studio
                                                                 Boston
```

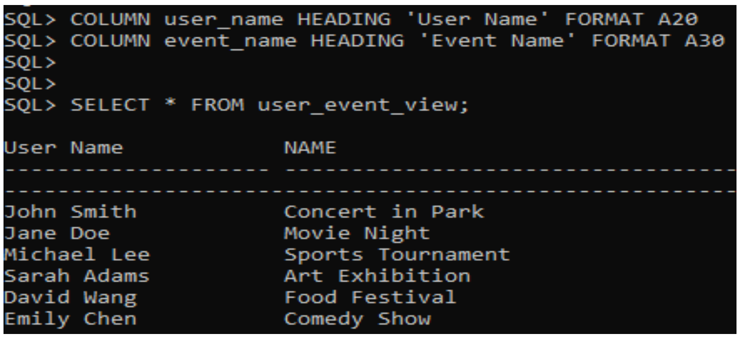
2. Create a view that combines data from the user and Events tables to display the name of each user along with the event names.



```
SQL> CREATE VIEW user_event_view AS
     SELECT u.name AS user_name, e.name
     FROM USER_URK22AI1048 u
     JOIN EVENT_URK22AI1048 e ON u.userid= e.eventid;
```

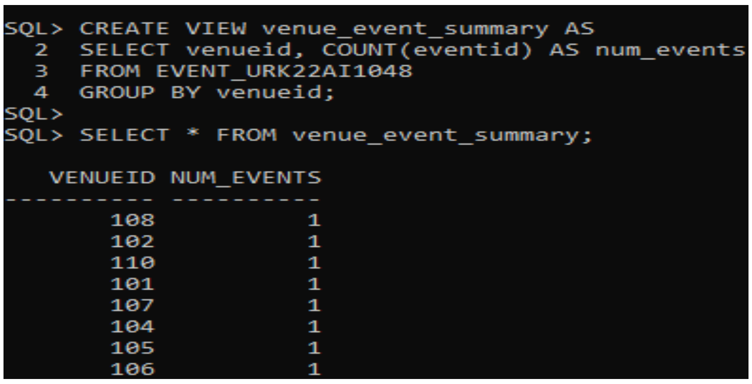


```
SQL> SELECT * FROM user_event_view;

USER_NAME
                                                                 NAME
-----------------------------------------------------------------------------
-------------------------------------------------------- --------------------
-----------------------------------------------------------------------------
John Smith
                                                                 Concert in Park
Jane Doe
                                                                 Movie Night
Michael Lee
                                                                 Sports Tournament
Sarah Adams
                                                                 Art Exhibition
David Wang
                                                                 Food Festival
Emily Chen
                                                                 Comedy Show
Alex Kim
                                                                 Tech Conference
Lisa Lopez
                                                                 Dance Workshop
```



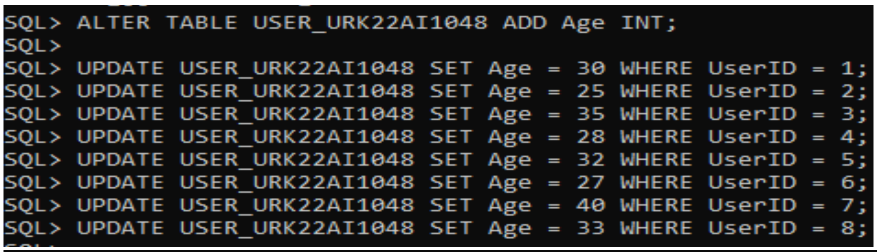
```
SQL> COLUMN user_name HEADING 'User Name' FORMAT A20
SQL> COLUMN event_name HEADING 'Event Name' FORMAT A30
SQL>
SQL>
SQL> SELECT * FROM user_event_view;

User Name            NAME
-------------------- -----------------------------------
--------------------------------------------------------
John Smith           Concert in Park
Jane Doe             Movie Night
Michael Lee          Sports Tournament
Sarah Adams          Art Exhibition
David Wang           Food Festival
Emily Chen           Comedy Show
```

3. Build a view that shows a summary of the number of events in each venue.

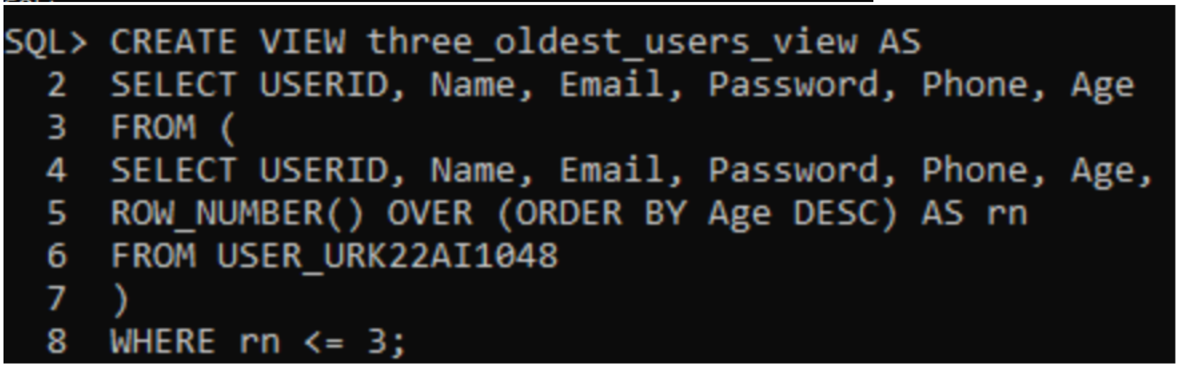


```
SQL> CREATE VIEW venue_event_summary AS
     SELECT venueid, COUNT(eventid) AS num_events
     FROM EVENT_URK22AI1048
     GROUP BY venueid;
SQL>
SQL> SELECT * FROM venue_event_summary;

   VENUEID NUM_EVENTS
---------- ----------
       108          1
       102          1
       110          1
       101          1
       107          1
       104          1
       105          1
       106          1
```

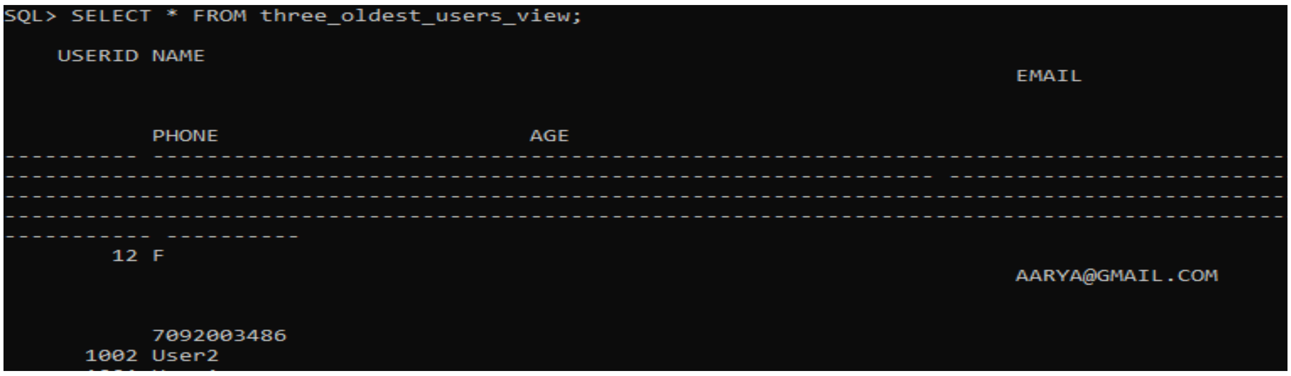
4. Display the 3 oldest users from the users table.



```
SQL> ALTER TABLE USER_URK22AI1048 ADD Age INT;
SQL>
SQL> UPDATE USER_URK22AI1048 SET Age = 30 WHERE UserID = 1;
SQL> UPDATE USER_URK22AI1048 SET Age = 25 WHERE UserID = 2;
SQL> UPDATE USER_URK22AI1048 SET Age = 35 WHERE UserID = 3;
SQL> UPDATE USER_URK22AI1048 SET Age = 28 WHERE UserID = 4;
SQL> UPDATE USER_URK22AI1048 SET Age = 32 WHERE UserID = 5;
SQL> UPDATE USER_URK22AI1048 SET Age = 27 WHERE UserID = 6;
SQL> UPDATE USER_URK22AI1048 SET Age = 40 WHERE UserID = 7;
SQL> UPDATE USER_URK22AI1048 SET Age = 33 WHERE UserID = 8;
```

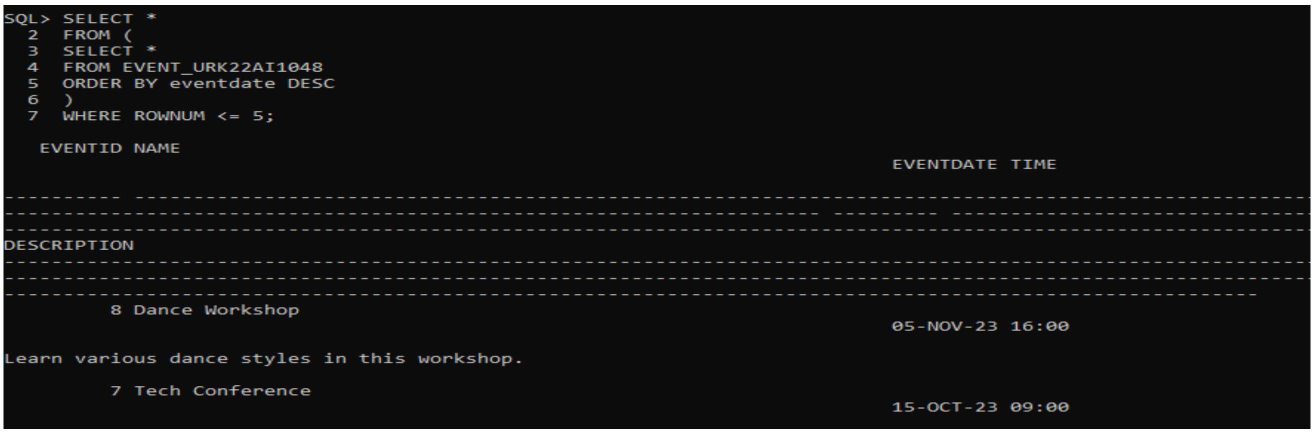


```
SQL> CREATE VIEW three_oldest_users_view AS
     SELECT USERID, Name, Email, Password, Phone, Age
     FROM (
     SELECT USERID, Name, Email, Password, Phone, Age,
     ROW_NUMBER() OVER (ORDER BY Age DESC) AS rn
     FROM USER_URK22AI1048
     )
     WHERE rn <= 3;
```



```
SQL> SELECT * FROM three_oldest_users_view;

    USERID NAME
                                                                          EMAIL


           PHONE                        AGE
---------- ---------------------------------------------------------------------------------
-------------------------------------------------------------------- ------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
---------- ----------
        12 F
                                                                          AARYA@GMAIL.COM


           7092003486
      1002 User2
```

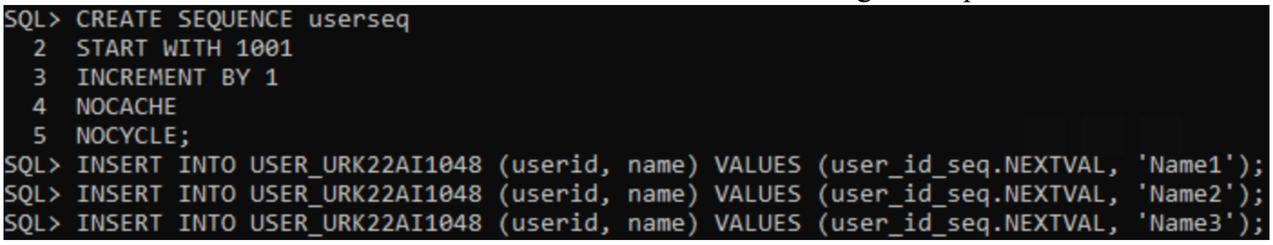
5. Display the last 5 events from the events table.



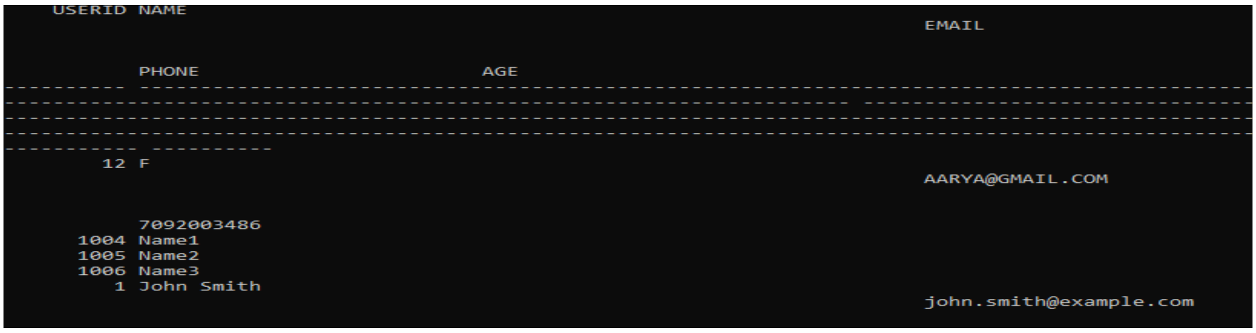
```
SQL> SELECT *
  2  FROM (
  3  SELECT *
  4  FROM EVENT_URK22AI1048
  5  ORDER BY eventdate DESC
  6  )
  7  WHERE ROWNUM <= 5;

   EVENTID NAME
                                                                           EVENTDATE TIME

---------- ----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------- --------- ------------------------------
-----------------------------------------------------------------------------------------------------------------
DESCRIPTION
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
         8 Dance Workshop
                                                                           05-NOV-23 16:00

Learn various dance styles in this workshop.

         7 Tech Conference
                                                                           15-OCT-23 09:00
```

6. Create a sequence that generates unique user IDs starting from 1001 and incrementing by 1

for each new user added to the users table. Add 3 new records using the sequence.

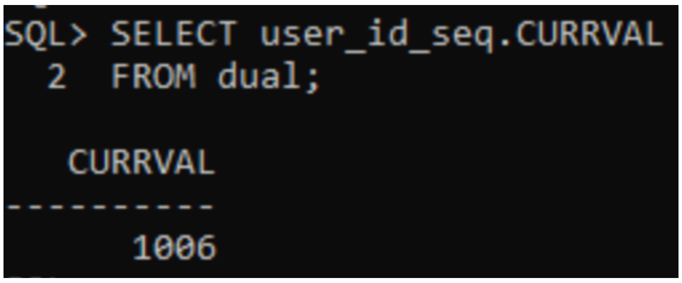


```
SQL> CREATE SEQUENCE userseq
  2  START WITH 1001
  3  INCREMENT BY 1
  4  NOCACHE
  5  NOCYCLE;
SQL> INSERT INTO USER_URK22AI1048 (userid, name) VALUES (user_id_seq.NEXTVAL, 'Name1');
SQL> INSERT INTO USER_URK22AI1048 (userid, name) VALUES (user_id_seq.NEXTVAL, 'Name2');
SQL> INSERT INTO USER_URK22AI1048 (userid, name) VALUES (user_id_seq.NEXTVAL, 'Name3');
```



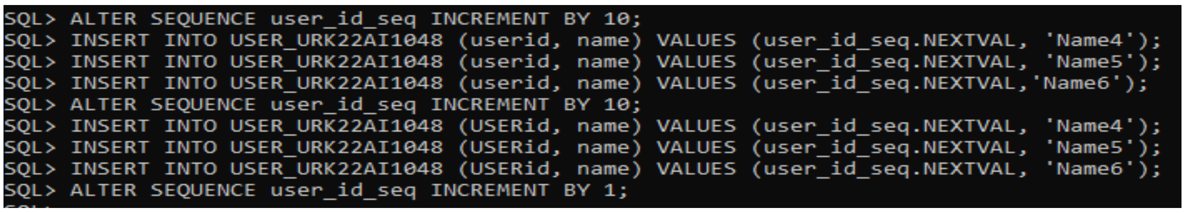
```
    USERID NAME
                                                                            EMAIL

           PHONE                         AGE
---------- ----------------------------------------------------------------------------------------------------
---------------------------------------------------------------------- ------------------------------------
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
----------- ----------
        12 F
                                                                            AARYA@GMAIL.COM


           7092003486
      1004 Name1
      1005 Name2
      1006 Name3
         1 John Smith
                                                                            john.smith@example.com
```

7. Display the current value of the sequence.

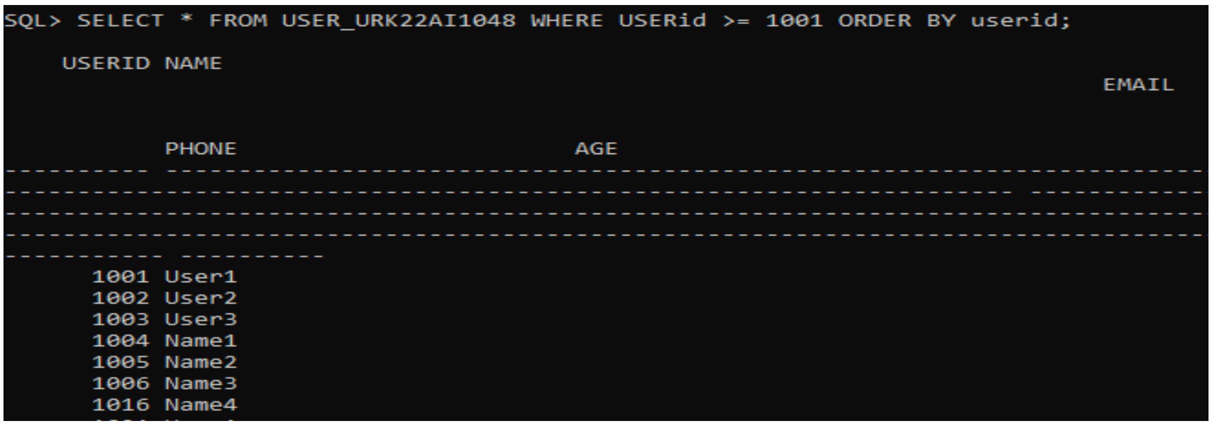


```
SQL> SELECT user_id_seq.CURRVAL
  2  FROM dual;

   CURRVAL
----------
      1006
```

8. Alter the sequence to increment by 10. Add 3 records to the users table using the sequence.

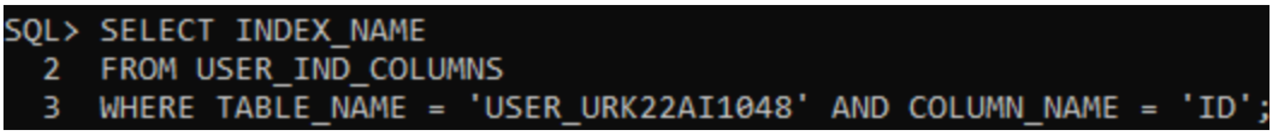


```
SQL> ALTER SEQUENCE user_id_seq INCREMENT BY 10;
SQL> INSERT INTO USER_URK22AI1048 (userid, name) VALUES (user_id_seq.NEXTVAL, 'Name4');
SQL> INSERT INTO USER_URK22AI1048 (userid, name) VALUES (user_id_seq.NEXTVAL, 'Name5');
SQL> INSERT INTO USER_URK22AI1048 (userid, name) VALUES (user_id_seq.NEXTVAL,'Name6');
SQL> ALTER SEQUENCE user_id_seq INCREMENT BY 10;
SQL> INSERT INTO USER_URK22AI1048 (USERid, name) VALUES (user_id_seq.NEXTVAL, 'Name4');
SQL> INSERT INTO USER_URK22AI1048 (USERid, name) VALUES (user_id_seq.NEXTVAL, 'Name5');
SQL> INSERT INTO USER_URK22AI1048 (USERid, name) VALUES (user_id_seq.NEXTVAL, 'Name6');
SQL> ALTER SEQUENCE user_id_seq INCREMENT BY 1;
```

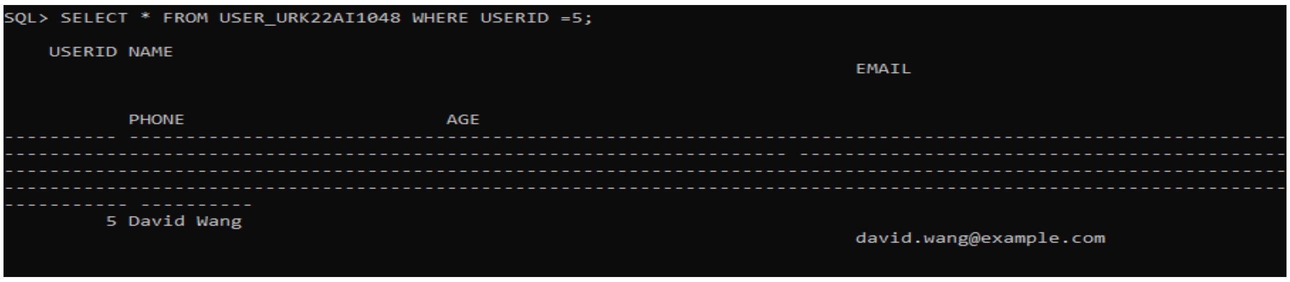


```
SQL> SELECT * FROM USER_URK22AI1048 WHERE USERid >= 1001 ORDER BY userid;

    USERID NAME
                                                                                 EMAIL


           PHONE                          AGE
---------- ------------------------------------------------------------------------------
------------------------------------------------------------------ ------------
---------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------
----------- ----------
      1001 User1
      1002 User2
      1003 User3
      1004 Name1
      1005 Name2
      1006 Name3
      1016 Name4
```

9. Create an index on the userid column of the users table and check the access time with userid in the where clause.

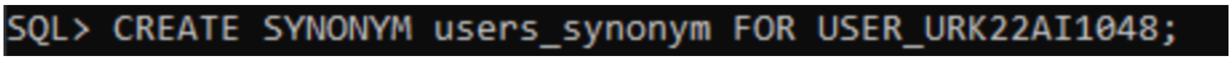


```
SQL> SELECT INDEX_NAME
  2  FROM USER_IND_COLUMNS
  3  WHERE TABLE_NAME = 'USER_URK22AI1048' AND COLUMN_NAME = 'ID';
```



```
SQL> SELECT * FROM USER_URK22AI1048 WHERE USERID =5;

    USERID NAME
                                                                      EMAIL


           PHONE                          AGE
---------- ------------------------------------------------------------------------------------------------
------------------------------------------------------------------- -----------------------------------
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
----------- ----------
         5 David Wang
                                                                      david.wang@example.com
```

10. Create a synonym for the users table.



```
SQL> CREATE SYNONYM users_synonym FOR USER_URK22AI1048;
```

Result

The given queries executed by the Creating of Other Database Objects successfully.