# Foundations of Databases  - Final Project, Part 3

Case Scenario - Public Library (Circulation)

Umma Islam

25FL-KG573-101 - Foundations of Database

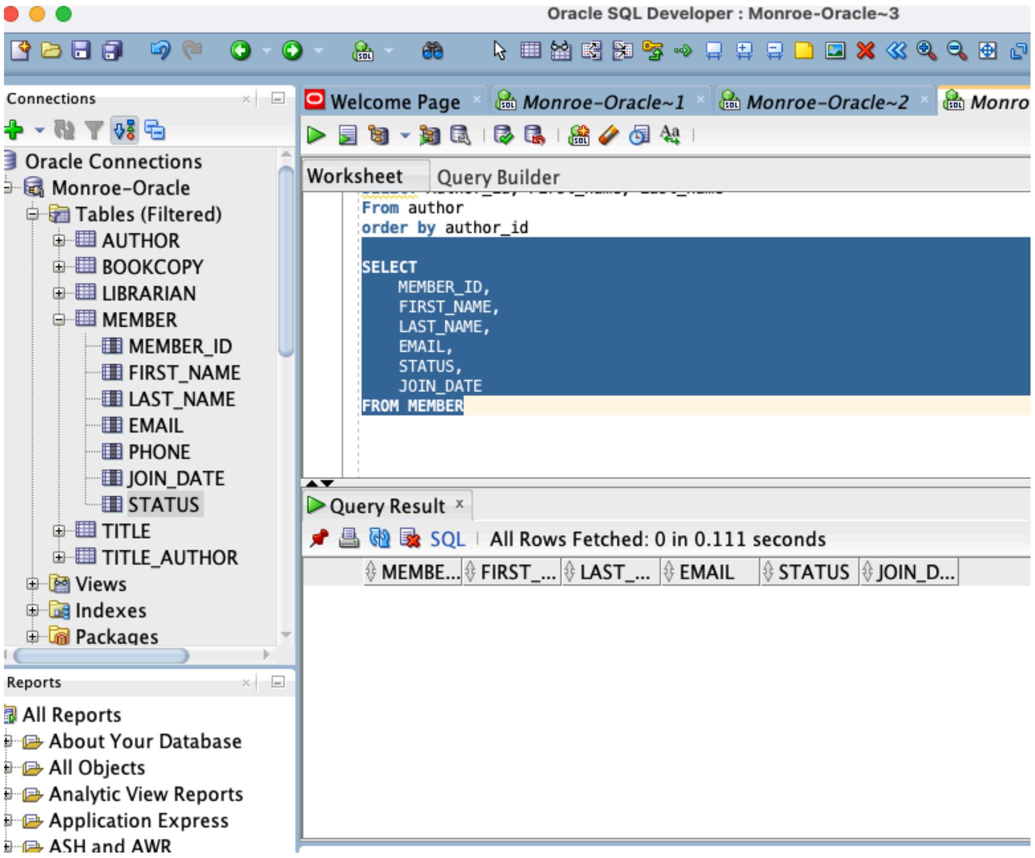
Date:11/18/2025

## 1.Basic Queries (SELECT)

Write at least 3 SELECT statements that use:

▪Specific columns (projection)

```
SELECT
   MEMBER_ID,
   FIRST_NAME,
   LAST_NAME,
   EMAIL,
   STATUS,
   JOIN_DATE
FROM MEMBER
```



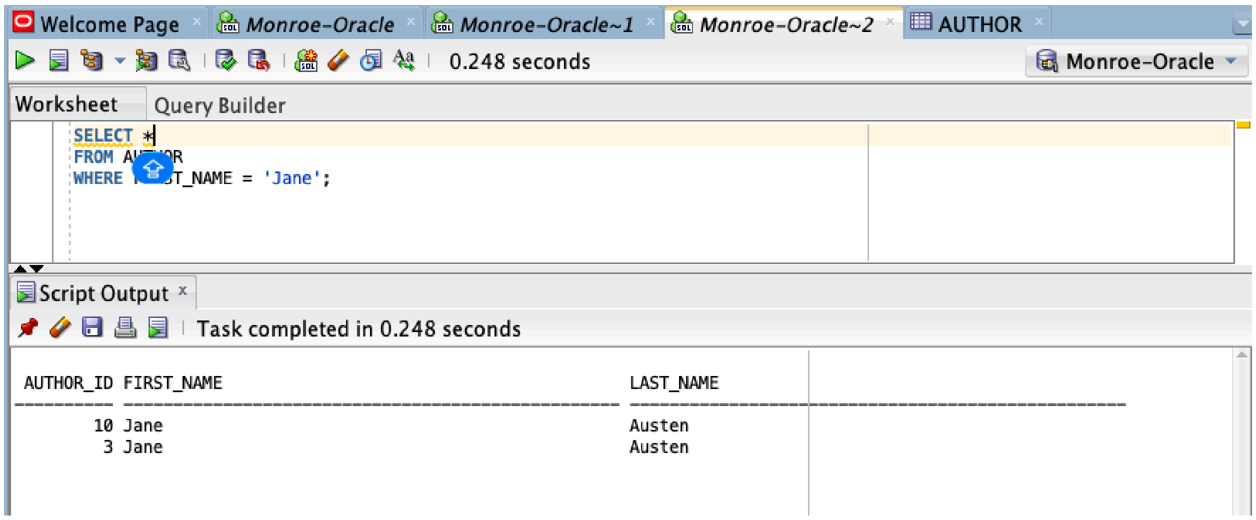
```
From author
order by author_id

SELECT
    MEMBER_ID,
    FIRST_NAME,
    LAST_NAME,
    EMAIL,
    STATUS,
    JOIN_DATE
FROM MEMBER
```

All Rows Fetched: 0 in 0.111 seconds

| MEMBE... | FIRST_... | LAST_... | EMAIL | STATUS | JOIN_D... |
|---|---|---|---|---|---|

▪Conditions using WHERE

```
SELECT *
FROM AUTHOR
WHERE FIRST_NAME = 'Jane';
```



```
SELECT *
FROM AUTHOR
WHERE F    T_NAME = 'Jane';
```

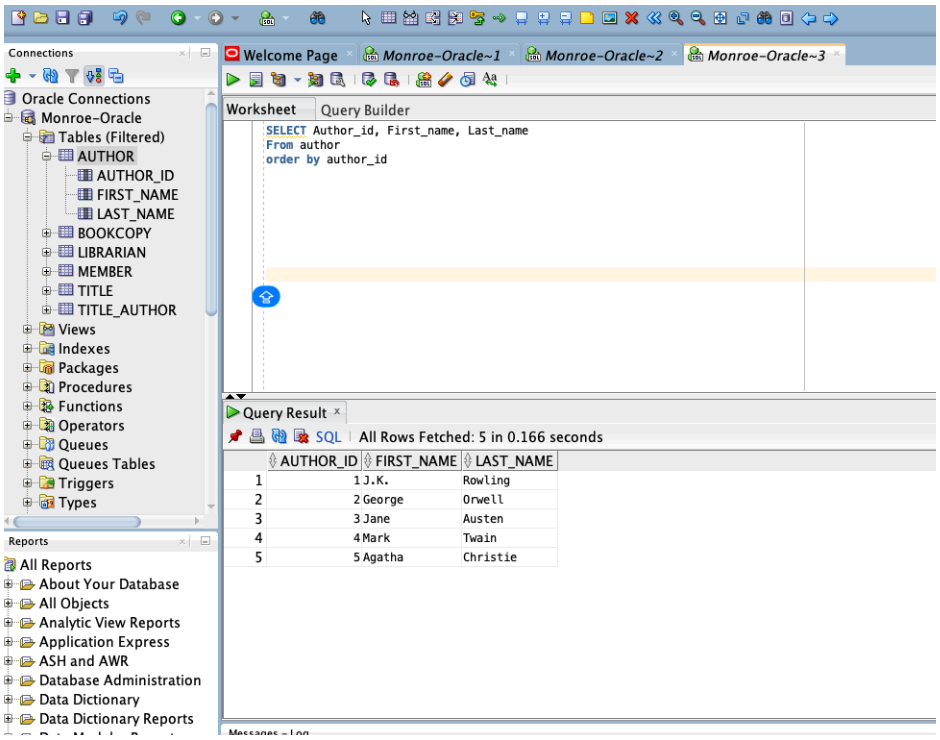
```
AUTHOR_ID FIRST_NAME                                          LAST_NAME
--------- --------------------------------------------------- --------------------------------------------------
       10 Jane                                                Austen
        3 Jane                                                Austen
```

▪Sorting using ORDER BY

SELECT Author_id, First_name, Last_name

From author

order by author_id



```
SELECT Author_id, First_name, Last_name
From author
order by author_id
```

| | AUTHOR_ID | FIRST_NAME | LAST_NAME |
|---|---|---|---|
| 1 | 1 | J.K. | Rowling |
| 2 | 2 | George | Orwell |
| 3 | 3 | Jane | Austen |
| 4 | 4 | Mark | Twain |
| 5 | 5 | Agatha | Christie |

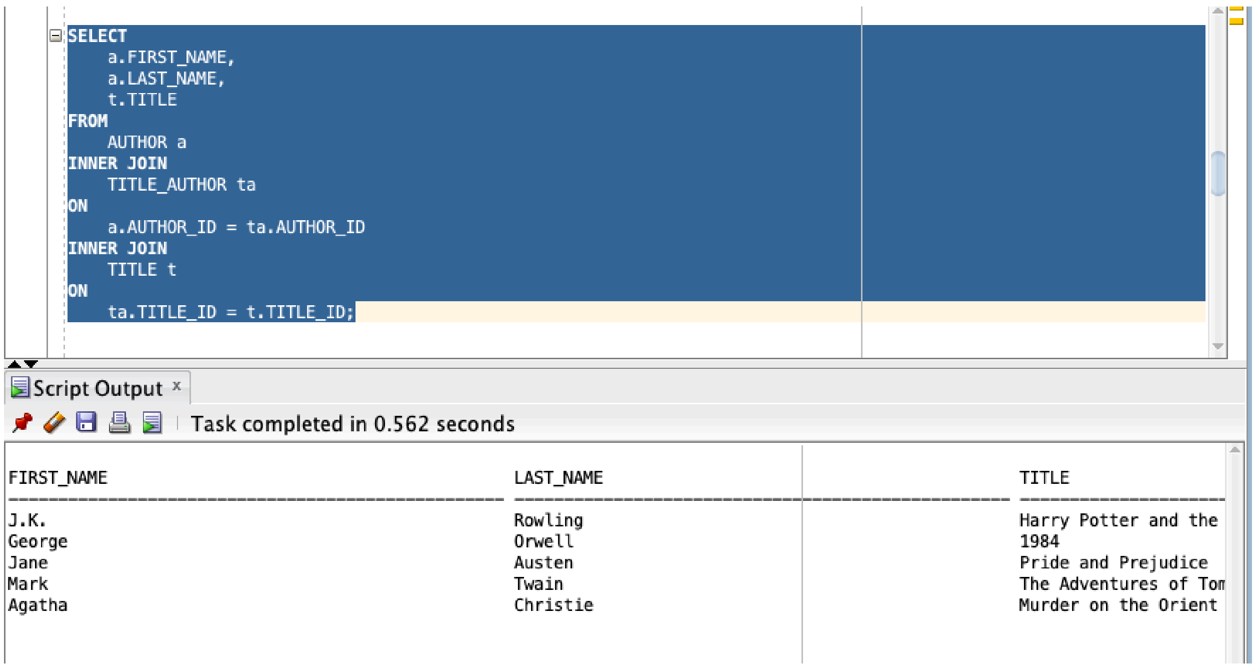
## 2. Join Queries

Write at least 3 queries that combine data from two or more tables using JOINs.

Examples: INNER JOIN, LEFT JOIN

1. Using Inner join command:

```
SELECT
   a.FIRST_NAME,
   a.LAST_NAME,
   t.TITLE
FROM
   AUTHOR a
INNER JOIN
   TITLE_AUTHOR ta
ON
   a.AUTHOR_ID = ta.AUTHOR_ID
INNER JOIN
   TITLE t
ON
   ta.TITLE_ID = t.TITLE_ID;
```



```
SELECT
    a.FIRST_NAME,
    a.LAST_NAME,
    t.TITLE
FROM
    AUTHOR a
INNER JOIN
    TITLE_AUTHOR ta
ON
    a.AUTHOR_ID = ta.AUTHOR_ID
INNER JOIN
    TITLE t
ON
    ta.TITLE_ID = t.TITLE_ID;
```

Script Output

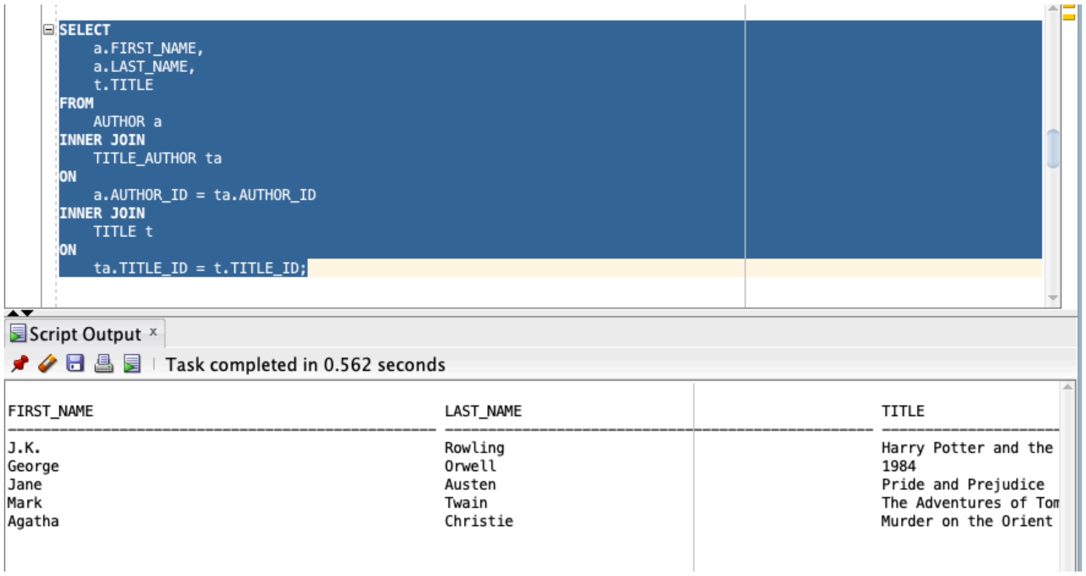
Task completed in 0.562 seconds

```
FIRST_NAME                                               LAST_NAME                                                TITLE
-------------------------------------------------------- -------------------------------------------------------- ----------------------
J.K.                                                     Rowling                                                  Harry Potter and the
George                                                   Orwell                                                   1984
Jane                                                     Austen                                                   Pride and Prejudice
Mark                                                     Twain                                                    The Adventures of Tom
Agatha                                                   Christie                                                 Murder on the Orient
```

2. **Using left join Command:**

```
SELECT
   t.TITLE,
   a.FIRST_NAME,
   a.LAST_NAME
FROM  TITLE t
LEFT JOIN TITLE_AUTHOR ta
ON t.TITLE_ID = ta.TITLE_ID
LEFT JOIN  AUTHOR a
```

ON ta.AUTHOR_ID = a.AUTHOR_ID;

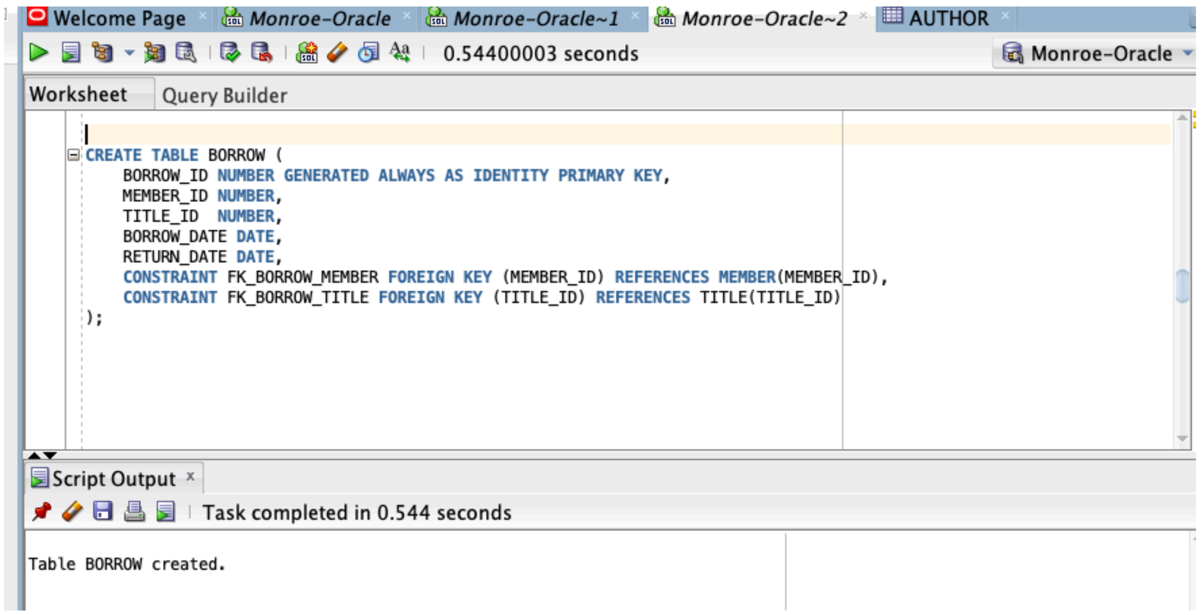


```
SELECT
    a.FIRST_NAME,
    a.LAST_NAME,
    t.TITLE
FROM
    AUTHOR a
INNER JOIN
    TITLE_AUTHOR ta
ON
    a.AUTHOR_ID = ta.AUTHOR_ID
INNER JOIN
    TITLE t
ON
    ta.TITLE_ID = t.TITLE_ID;
```

```
Task completed in 0.562 seconds

FIRST_NAME                                                  LAST_NAME                                                   TITLE
----------------------------------------------------------  ----------------------------------------------------------  -------------------------
J.K.                                                        Rowling                                                     Harry Potter and the
George                                                      Orwell                                                      1984
Jane                                                        Austen                                                      Pride and Prejudice
Mark                                                        Twain                                                       The Adventures of Tom
Agatha                                                      Christie                                                    Murder on the Orient
```

**3.Using Borrow command.**

```
CREATE TABLE BORROW (
    BORROW_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    MEMBER_ID NUMBER,
    TITLE_ID  NUMBER,
    BORROW_DATE DATE,
    RETURN_DATE DATE,
    CONSTRAINT FK_BORROW_MEMBER FOREIGN KEY (MEMBER_ID) REFERENCES
MEMBER(MEMBER_ID),
    CONSTRAINT FK_BORROW_TITLE FOREIGN KEY (TITLE_ID) REFERENCES
TITLE(TITLE_ID)
);
```



```
CREATE TABLE BORROW (
    BORROW_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    MEMBER_ID NUMBER,
    TITLE_ID  NUMBER,
    BORROW_DATE DATE,
    RETURN_DATE DATE,
    CONSTRAINT FK_BORROW_MEMBER FOREIGN KEY (MEMBER_ID) REFERENCES MEMBER(MEMBER_ID),
    CONSTRAINT FK_BORROW_TITLE FOREIGN KEY (TITLE_ID) REFERENCES TITLE(TITLE_ID)
);
```

Task completed in 0.544 seconds

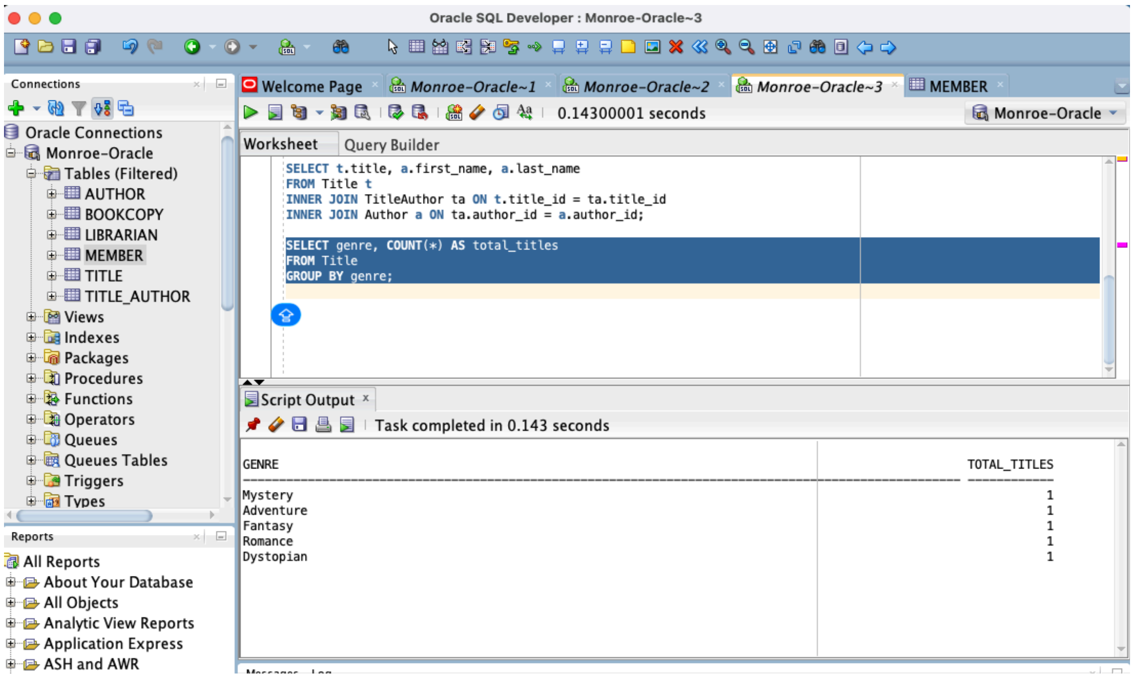
```
Table BORROW created.
```

**3.Aggregate Queries**

oWrite at least 3 queries using aggregate functions such as COUNT, SUM, AVG, MIN, or MAX.
oUse GROUP BY and HAVING where appropriate.

**Group by:**

SELECT genre, COUNT(*) AS total_titles
FROM Title
GROUP BY genre;



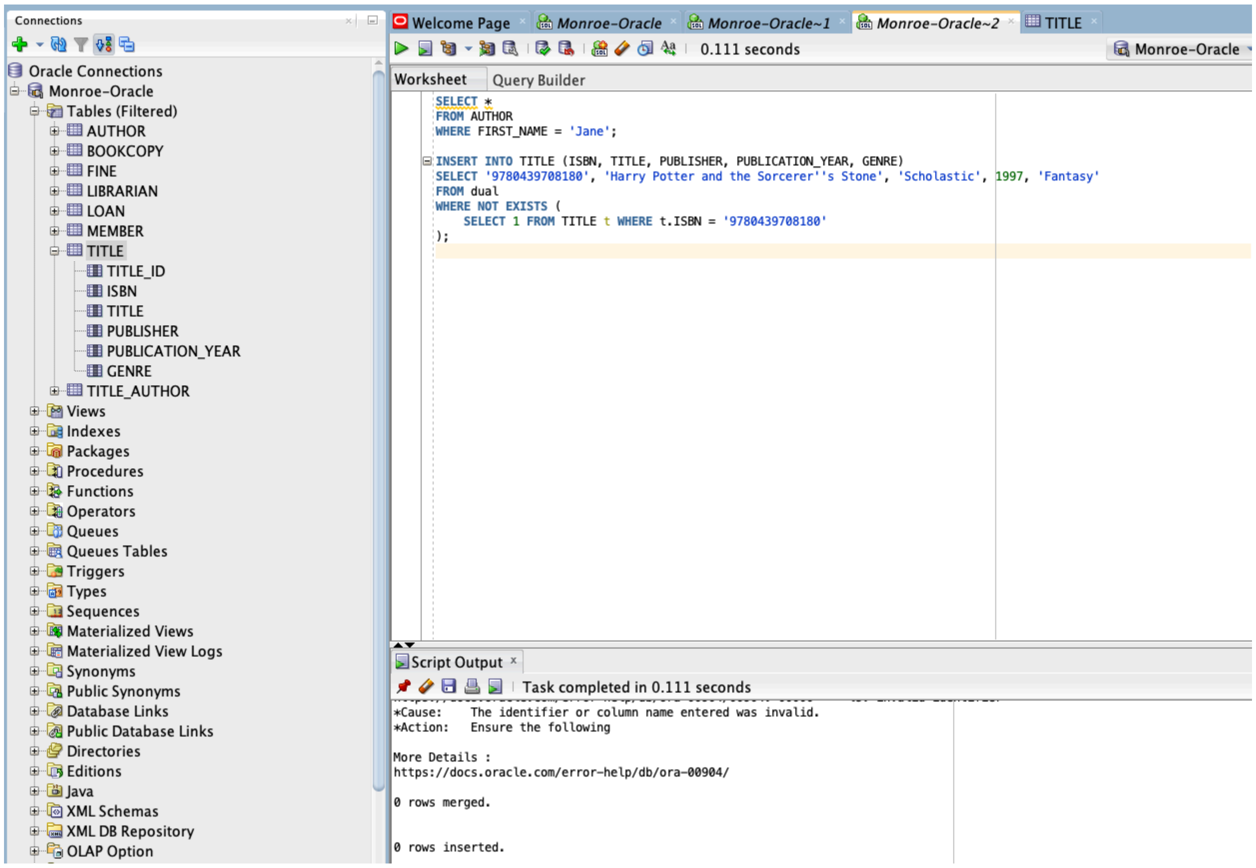
```
SELECT t.title, a.first_name, a.last_name
FROM Title t
INNER JOIN TitleAuthor ta ON t.title_id = ta.title_id
INNER JOIN Author a ON ta.author_id = a.author_id;

SELECT genre, COUNT(*) AS total_titles
FROM Title
GROUP BY genre;
```

Task completed in 0.143 seconds

```
GENRE                                                                                   TOTAL_TITLES
--------------------------------------------------------------------------------------- ------------
Mystery                                                                                            1
Adventure                                                                                          1
Fantasy                                                                                            1
Romance                                                                                            1
Dystopian                                                                                          1
```

**Count:**

```
SELECT
   t.TITLE,
   COUNT(ta.AUTHOR_ID) AS AUTHOR_COUNT
FROM
   TITLE t
JOIN
   TITLE_AUTHOR ta
ON
   t.TITLE_ID = ta.TITLE_ID
GROUP BY
   t.TITLE
HAVING
   COUNT(ta.AUTHOR_ID) > 1;
```



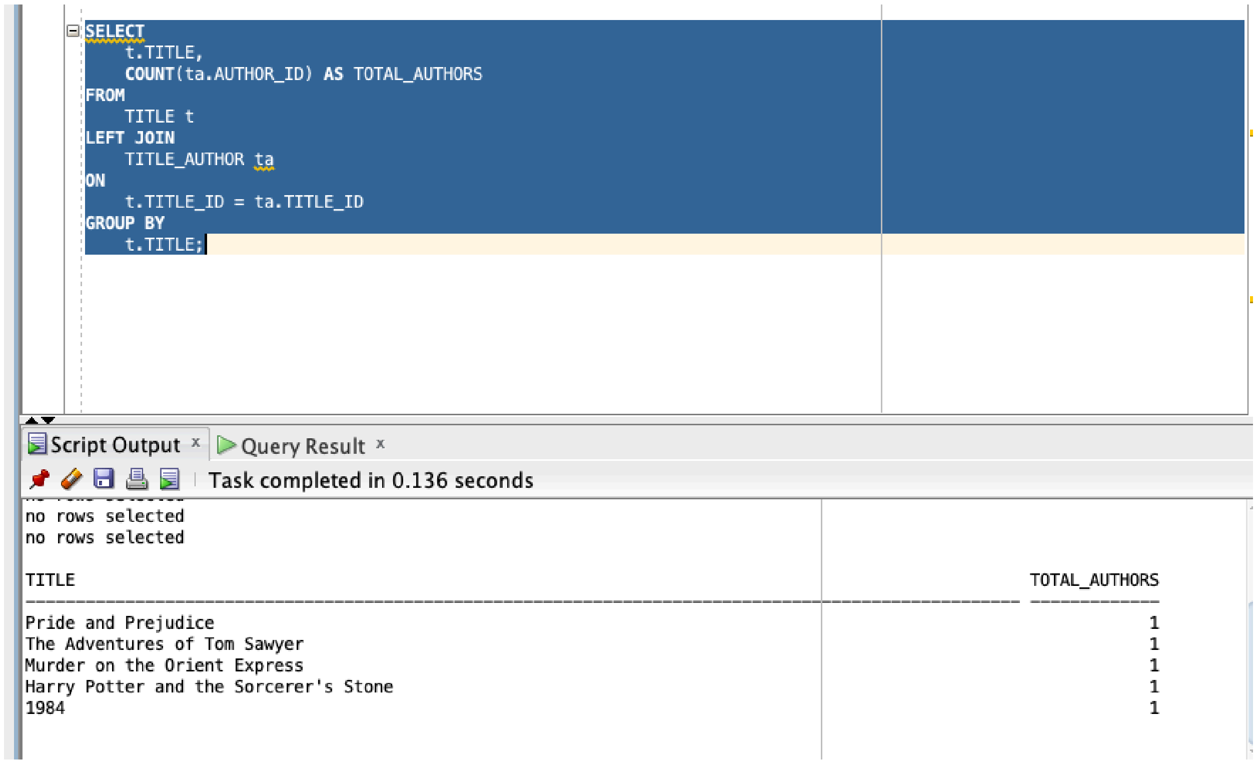
```
SELECT *
FROM AUTHOR
WHERE FIRST_NAME = 'Jane';

INSERT INTO TITLE (ISBN, TITLE, PUBLISHER, PUBLICATION_YEAR, GENRE)
SELECT '9780439708180', 'Harry Potter and the Sorcerer''s Stone', 'Scholastic', 1997, 'Fantasy'
FROM dual
WHERE NOT EXISTS (
    SELECT 1 FROM TITLE t WHERE t.ISBN = '9780439708180'
);
```

```
*Cause:    The identifier or column name entered was invalid.
*Action:   Ensure the following

More Details :
https://docs.oracle.com/error-help/db/ora-00904/

0 rows merged.


0 rows inserted.
```

**Sum:**

```
SELECT
   t.TITLE,
   COUNT(ta.AUTHOR_ID) AS TOTAL_AUTHORS
FROM
```

    TITLE t
LEFT JOIN
    TITLE_AUTHOR ta
ON
    t.TITLE_ID = ta.TITLE_ID
GROUP BY
    t.TITLE;



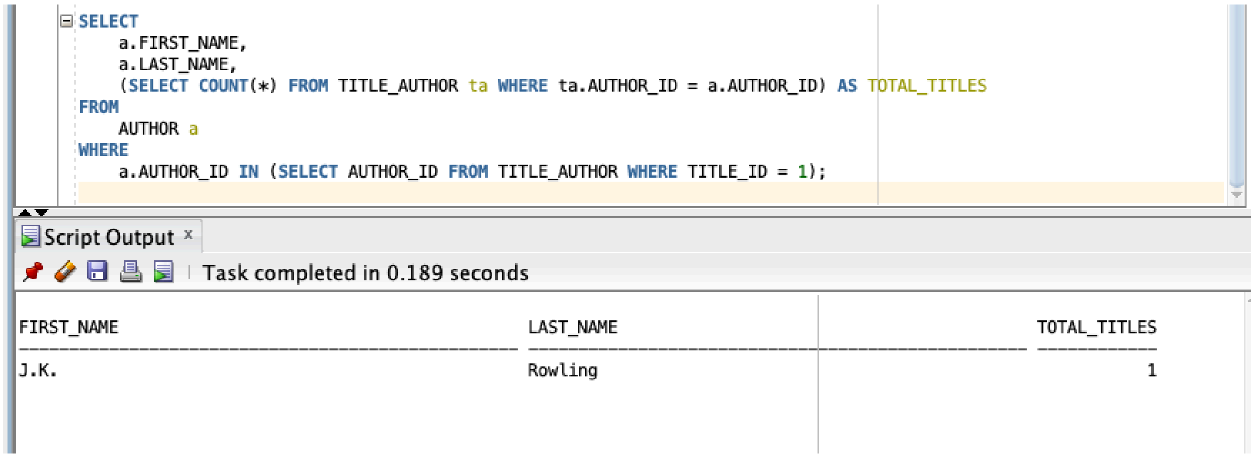
```
SELECT
    t.TITLE,
    COUNT(ta.AUTHOR_ID) AS TOTAL_AUTHORS
FROM
    TITLE t
LEFT JOIN
    TITLE_AUTHOR ta
ON
    t.TITLE_ID = ta.TITLE_ID
GROUP BY
    t.TITLE;
```

Task completed in 0.136 seconds

```
no rows selected
no rows selected

TITLE                                                                                                  TOTAL_AUTHORS
---------------------------------------------------------------------------------------------------- -------------
Pride and Prejudice                                                                                                1
The Adventures of Tom Sawyer                                                                                       1
Murder on the Orient Express                                                                                       1
Harry Potter and the Sorcerer's Stone                                                                              1
1984                                                                                                               1
```

**4.Subqueries**
oWrite at least 2 queries that use subqueries (nested SELECT statements).

**1.Using nested SELECT statements**
SELECT
    a.FIRST_NAME,
    a.LAST_NAME,
    (SELECT COUNT(*) FROM TITLE_AUTHOR ta WHERE ta.AUTHOR_ID = a.AUTHOR_ID) AS TOTAL_TITLES
FROM
    AUTHOR a
WHERE
    a.AUTHOR_ID IN (SELECT AUTHOR_ID FROM TITLE_AUTHOR WHERE TITLE_ID = 1);

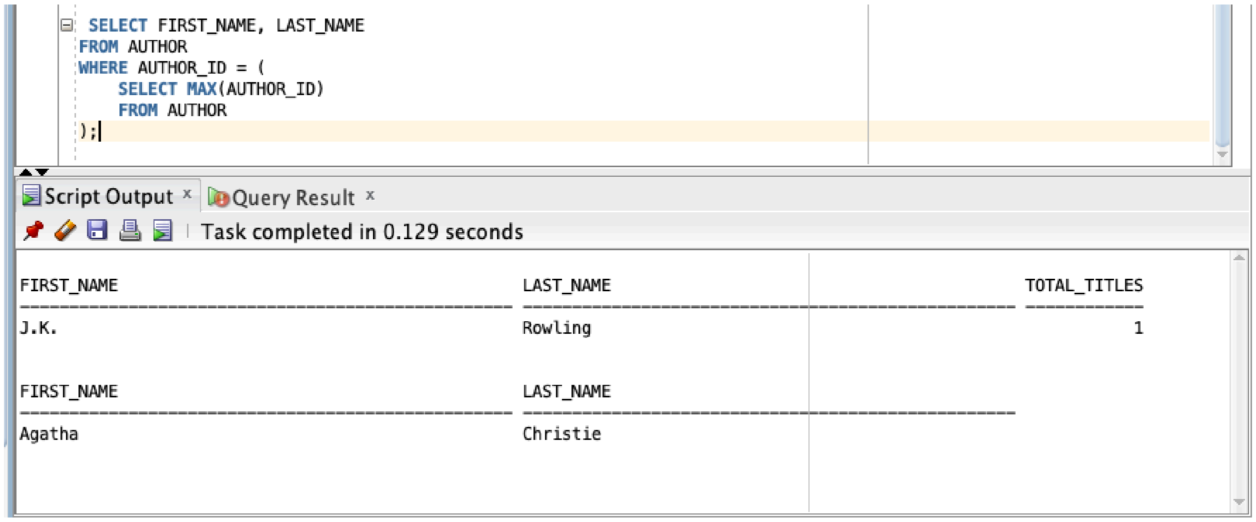


```
SELECT
    a.FIRST_NAME,
    a.LAST_NAME,
    (SELECT COUNT(*) FROM TITLE_AUTHOR ta WHERE ta.AUTHOR_ID = a.AUTHOR_ID) AS TOTAL_TITLES
FROM
    AUTHOR a
WHERE
    a.AUTHOR_ID IN (SELECT AUTHOR_ID FROM TITLE_AUTHOR WHERE TITLE_ID = 1);
```

```
FIRST_NAME                                         LAST_NAME                                          TOTAL_TITLES
-------------------------------------------------- -------------------------------------------------- ------------
J.K.                                               Rowling                                                       1
```

**2.Using nested SELECT statements**

SELECT FIRST_NAME, LAST_NAME
FROM AUTHOR
WHERE AUTHOR_ID = (
SELECT MAX(AUTHOR_ID)
FROM AUTHOR
);



```
SELECT FIRST_NAME, LAST_NAME
FROM AUTHOR
WHERE AUTHOR_ID = (
    SELECT MAX(AUTHOR_ID)
    FROM AUTHOR
);
```

```
FIRST_NAME                                         LAST_NAME                                          TOTAL_TITLES
-------------------------------------------------- -------------------------------------------------- ------------
J.K.                                               Rowling                                                       1

FIRST_NAME                                         LAST_NAME
-------------------------------------------------- --------------------------------------------------
Agatha                                             Christie
```

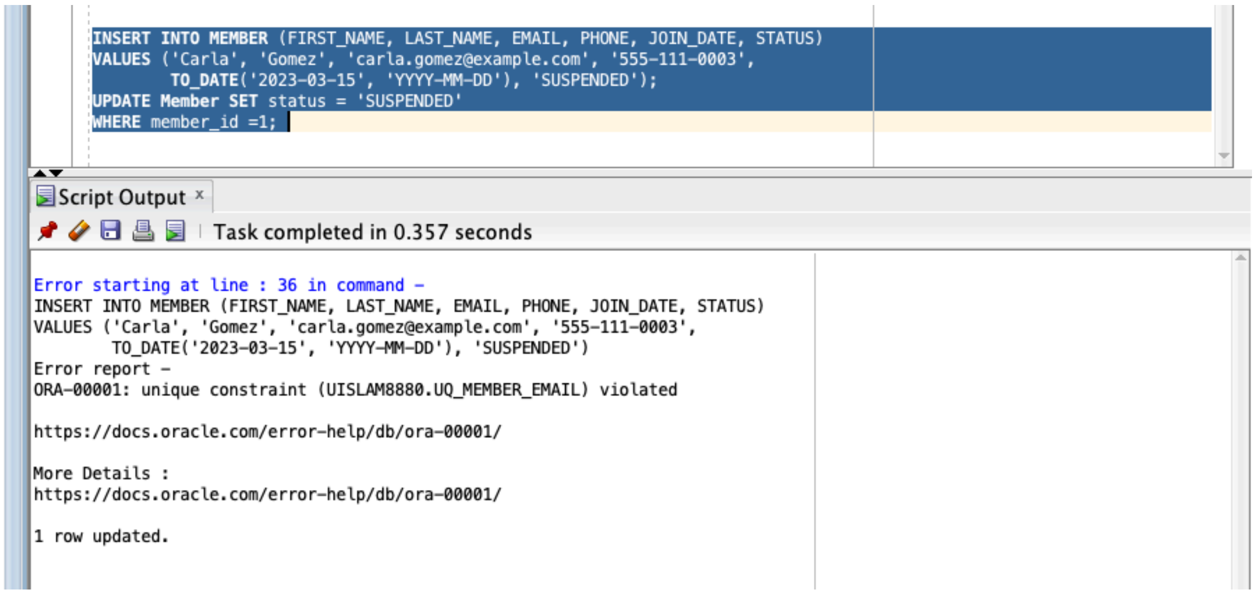
**5.Data Manipulation (DML)**

- ☐ Include examples of the following commands:
  - ▪INSERT – Add new records to a table
  - ▪UPDATE –Modify existing records
  - ▪DELETE – Remove records from a table

oProvide at least 2 examples of each command type

1. **Using Update command:**

INSERT INTO MEMBER (FIRST_NAME, LAST_NAME, EMAIL, PHONE, JOIN_DATE, STATUS)
VALUES ('Carla', 'Gomez', 'carla.gomez@example.com', '555-111-0003',
 TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'SUSPENDED');
UPDATE Member SET status = 'SUSPENDED'
WHERE member_id =1;



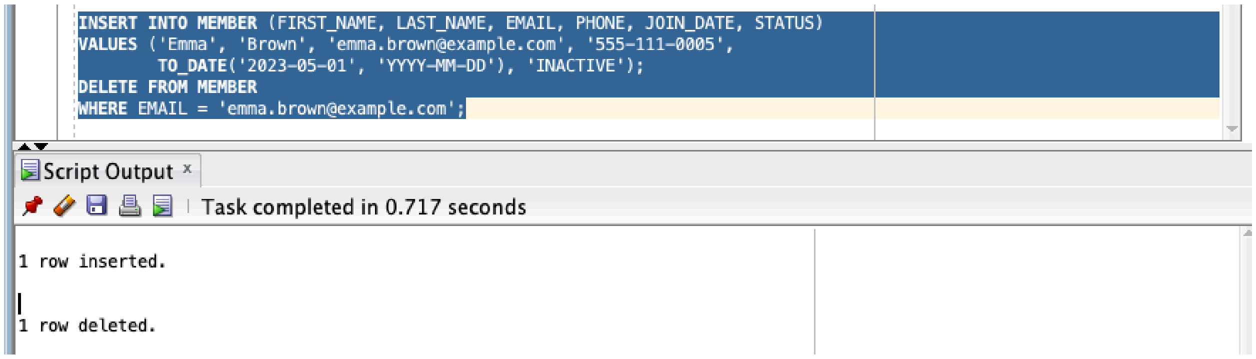
```
INSERT INTO MEMBER (FIRST_NAME, LAST_NAME, EMAIL, PHONE, JOIN_DATE, STATUS)
VALUES ('Carla', 'Gomez', 'carla.gomez@example.com', '555-111-0003',
          TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'SUSPENDED');
UPDATE Member SET status = 'SUSPENDED'
WHERE member_id =1;
```

Script Output × | Task completed in 0.357 seconds

```
Error starting at line : 36 in command -
INSERT INTO MEMBER (FIRST_NAME, LAST_NAME, EMAIL, PHONE, JOIN_DATE, STATUS)
VALUES ('Carla', 'Gomez', 'carla.gomez@example.com', '555-111-0003',
          TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'SUSPENDED')
Error report -
ORA-00001: unique constraint (UISLAM8880.UQ_MEMBER_EMAIL) violated

https://docs.oracle.com/error-help/db/ora-00001/

More Details :
https://docs.oracle.com/error-help/db/ora-00001/

1 row updated.
```

2. **Using DELETE command:**

INSERT INTO MEMBER (FIRST_NAME, LAST_NAME, EMAIL, PHONE, JOIN_DATE, STATUS)
VALUES ('Emma', 'Brown', 'emma.brown@example.com', '555-111-0005',
 TO_DATE('2023-05-01', 'YYYY-MM-DD'), 'INACTIVE');
DELETE FROM MEMBER
WHERE EMAIL = 'emma.brown@example.com';



```
INSERT INTO MEMBER (FIRST_NAME, LAST_NAME, EMAIL, PHONE, JOIN_DATE, STATUS)
VALUES ('Emma', 'Brown', 'emma.brown@example.com', '555-111-0005',
        TO_DATE('2023-05-01', 'YYYY-MM-DD'), 'INACTIVE');
DELETE FROM MEMBER
WHERE EMAIL = 'emma.brown@example.com';
```

```
1 row inserted.


1 row deleted.
```