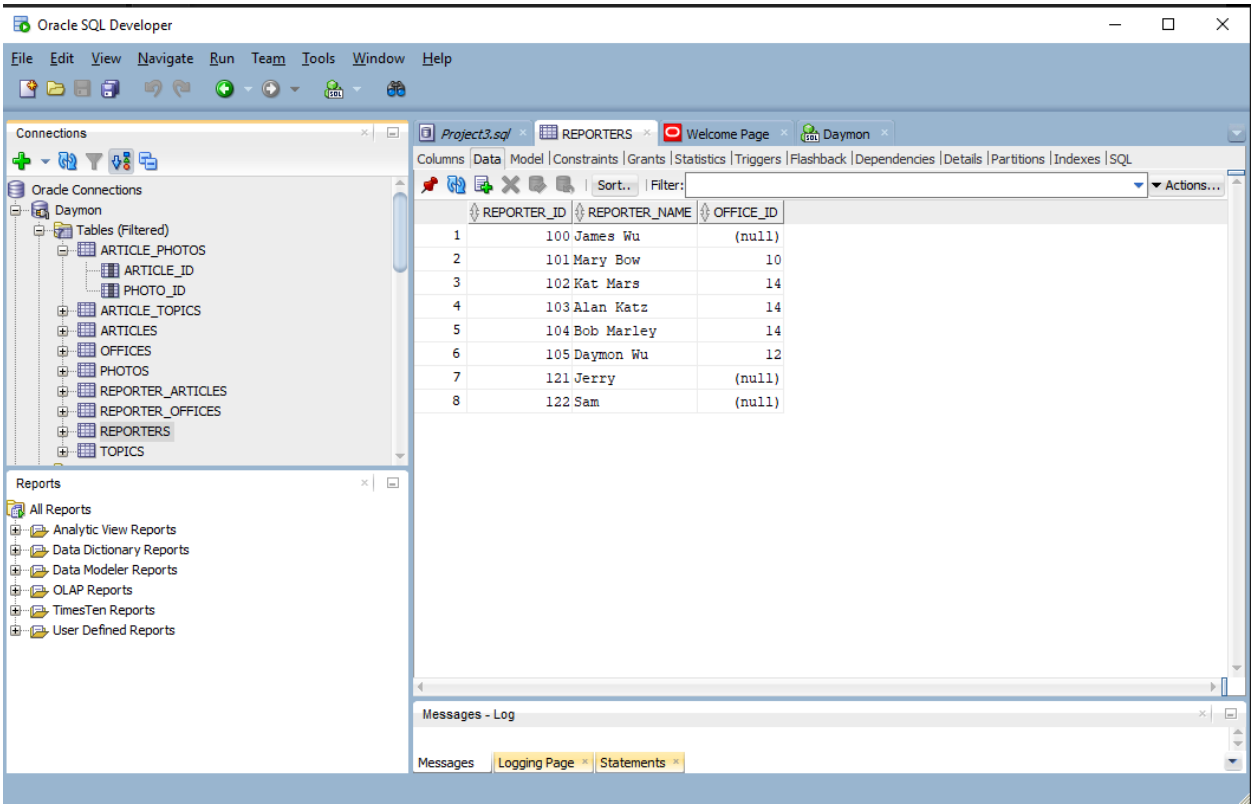
Name: Daymon Wu

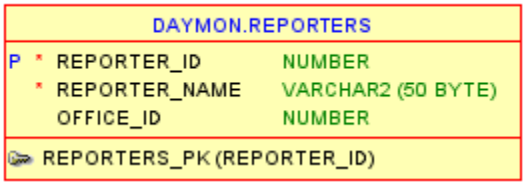
Session: CSCI 331 32[33506] MW 8PM

Professor Daniel Levitt

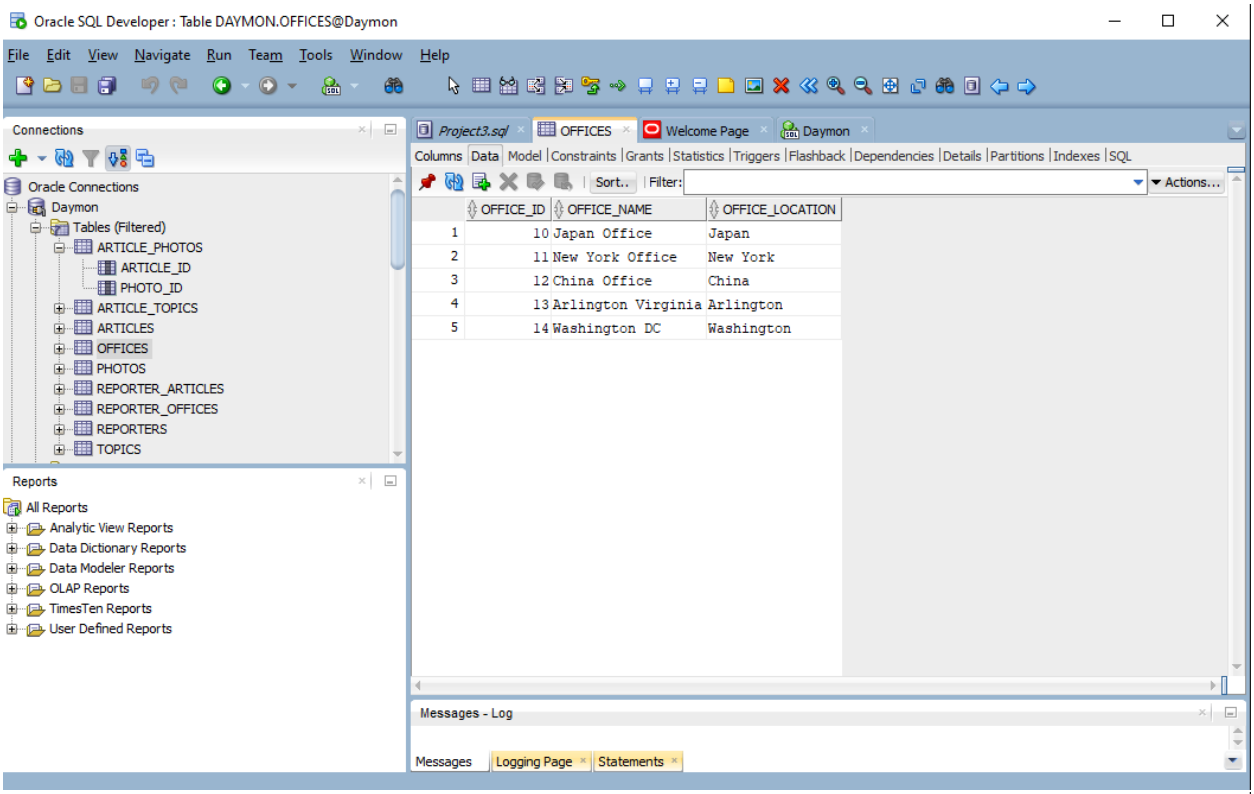
Tables:

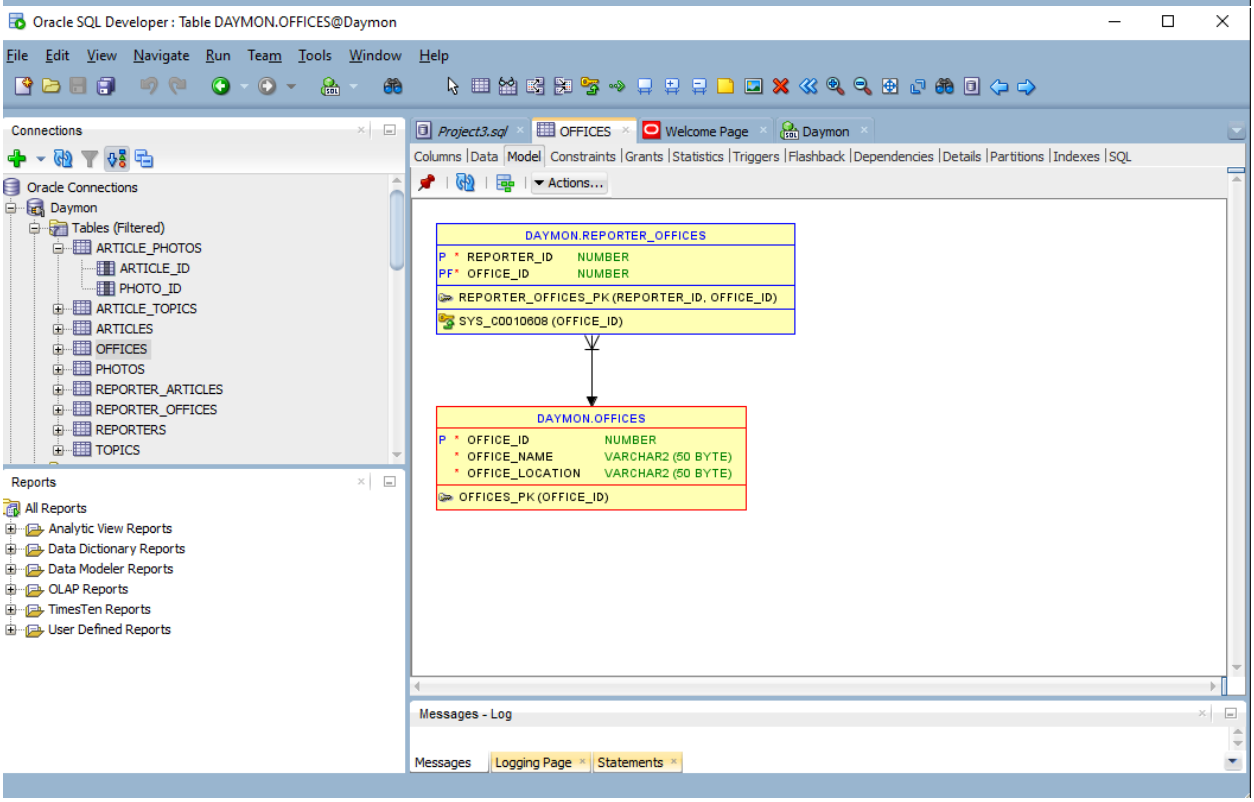
```
CREATE TABLE reporters (
    reporter_id NUMBER PRIMARY KEY,
    reporter_name VARCHAR2(50) NOT NULL
)
```



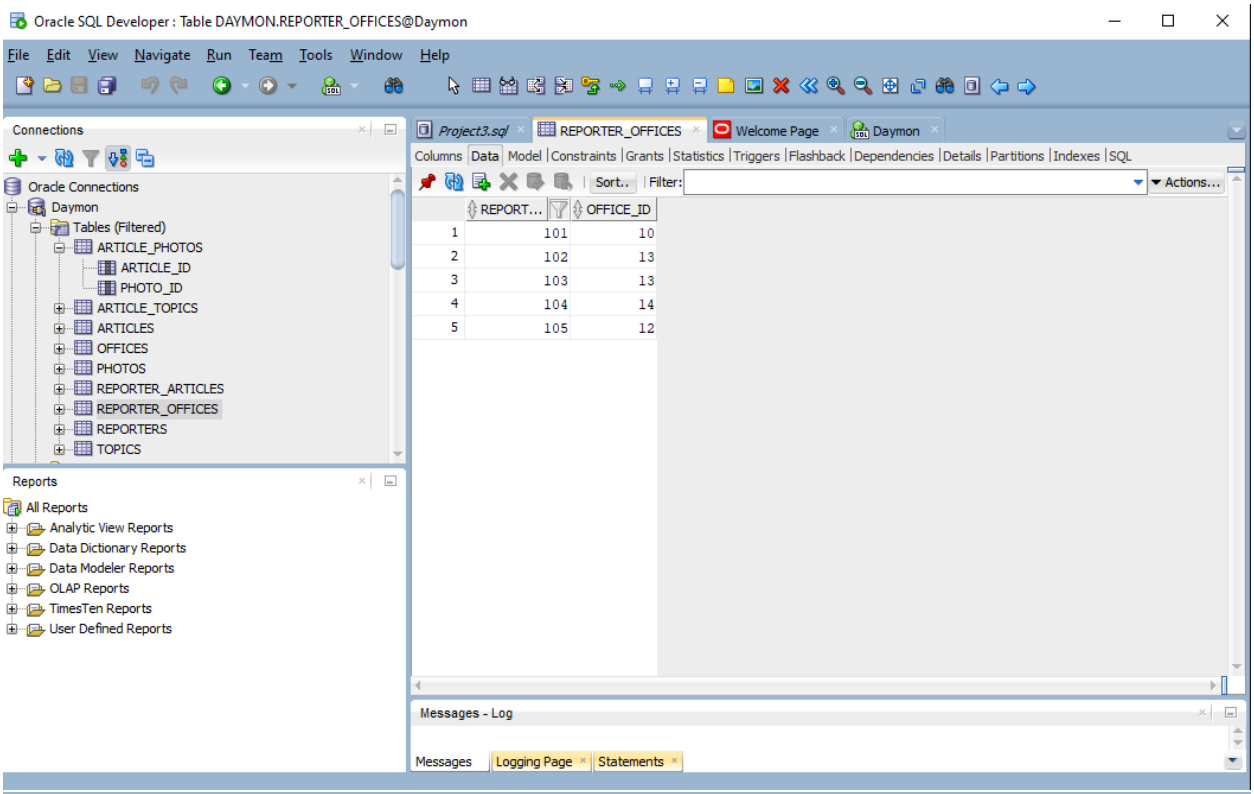


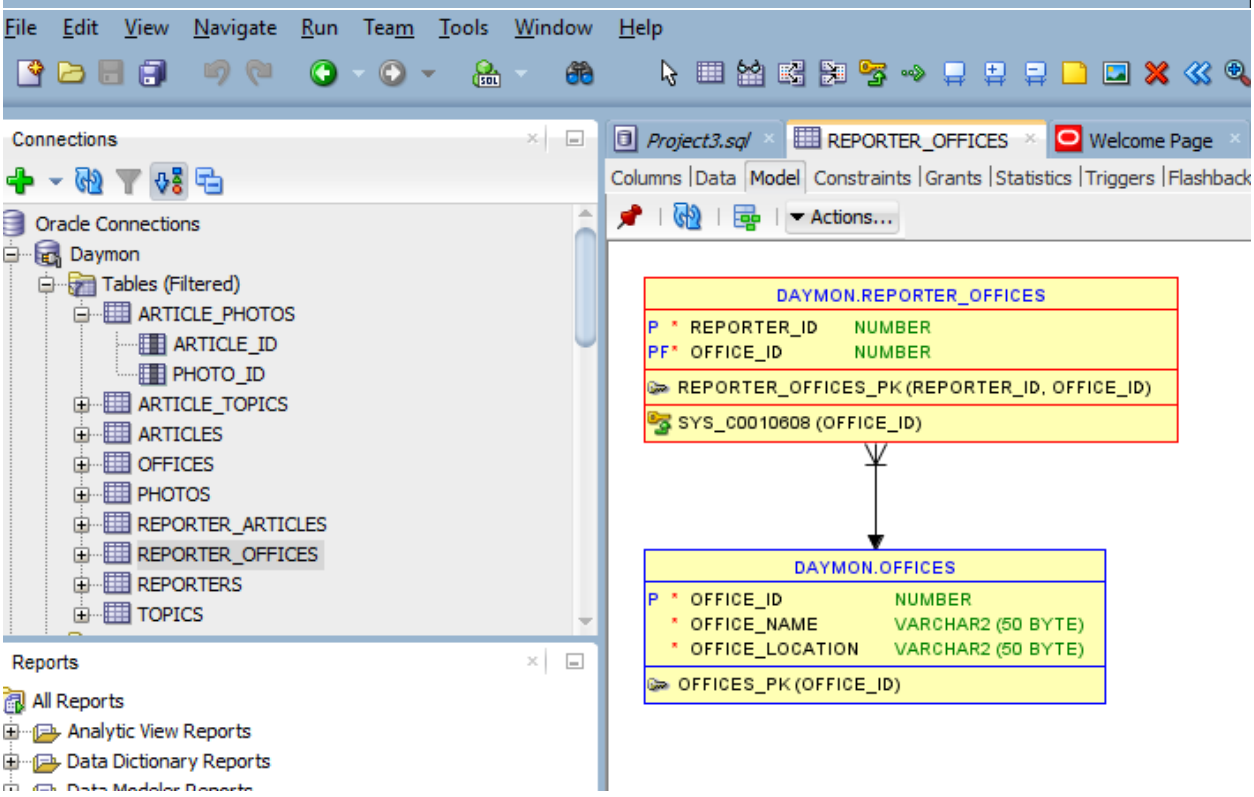

```
CREATE TABLE offices (
    office_id NUMBER PRIMARY KEY,
    office_name VARCHAR2(50) NOT NULL,
    office_location VARCHAR2(50) NOT NULL
)
```


Oracle SQL Developer : Table DAYMON.OFFICES@Daymon
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
Daymon
Tables (Filtered)
ARTICLE_PHOTOS
ARTICLE_ID
PHOTO_ID
ARTICLE_TOPICS
ARTICLES
OFFICES
PHOTOS
REPORTER_ARTICLES
REPORTER_OFFICES
REPORTERS
TOPICS
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Project3.sql
OFFICES
Welcome Page
Daymon
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Sort..
Filter:
Actions...
OFFICE_ID OFFICE_NAME OFFICE_LOCATION
1 10 Japan Office Japan
2 11 New York Office New York
3 12 China Office China
4 13 Arlington Virginia Arlington
5 14 Washington DC Washington
Messages - Log
Messages
Logging Page
Statements

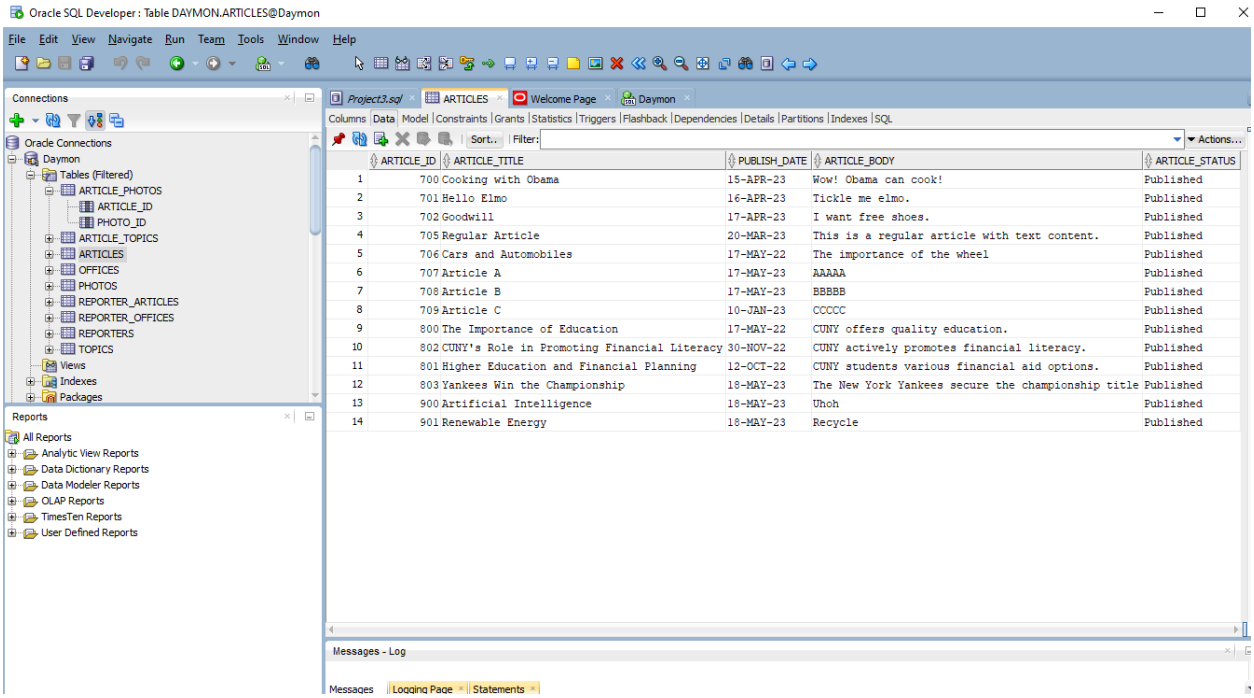

Oracle SQL Developer : Table DAYMON.OFFICES@Daymon
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
Daymon
Tables (Filtered)
ARTICLE_PHOTOS
ARTICLE_ID
PHOTO_ID
ARTICLE_TOPICS
ARTICLES
OFFICES
PHOTOS
REPORTER_ARTICLES
REPORTER_OFFICES
REPORTERS
TOPICS
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Project3.sql
OFFICES
Welcome Page
Daymon
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Actions...
DAYMON.REPORTER_OFFICES
P * REPORTER_ID NUMBER
PF* OFFICE_ID NUMBER
REPORTER_OFFICES_PK (REPORTER_ID, OFFICE_ID)
SYS_C0010608 (OFFICE_ID)
DAYMON.OFFICES
P * OFFICE_ID NUMBER
* OFFICE_NAME VARCHAR2 (50 BYTE)
* OFFICE_LOCATION VARCHAR2 (50 BYTE)
OFFICES_PK (OFFICE_ID)
Messages - Log
Messages
Logging Page
Statements

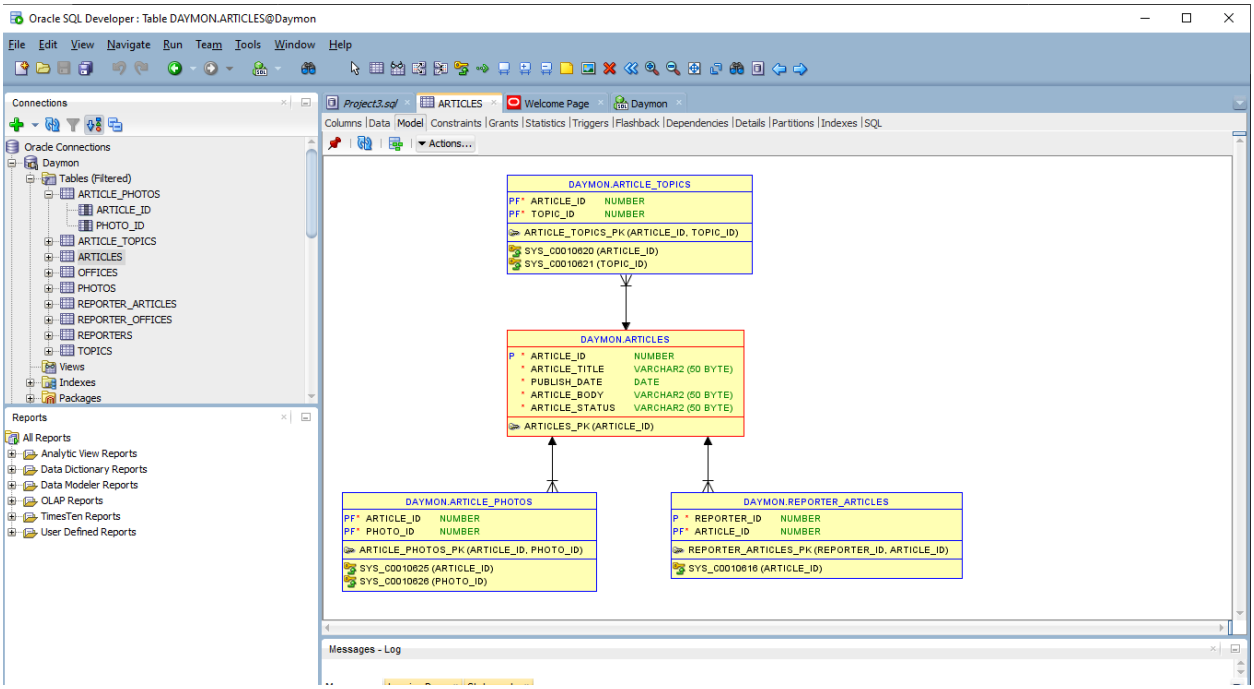
```
CREATE TABLE reporter_offices (
    reporter_id NUMBER,
    office_id NUMBER,
    PRIMARY KEY (reporter_id, office_id),
    FOREIGN KEY (reporter_id) REFERENCES reporters (reporter_id),
    FOREIGN KEY (office_id) REFERENCES offices (office_id)
)
```


Oracle SQL Developer : Table DAYMON.REPORTER_OFFICES@Daymon
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
Daymon
Tables (Filtered)
ARTICLE_PHOTOS
ARTICLE_ID
PHOTO_ID
ARTICLE_TOPICS
ARTICLES
OFFICES
PHOTOS
REPORTER_ARTICLES
REPORTER_OFFICES
REPORTERS
TOPICS
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Project3.sql
REPORTER_OFFICES
Welcome Page
Daymon
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Sort..
Filter:
Actions...
REPORT...
OFFICE_ID
1 101 10
2 102 13
3 103 13
4 104 14
5 105 12
Messages - Log
Messages
Logging Page
Statements

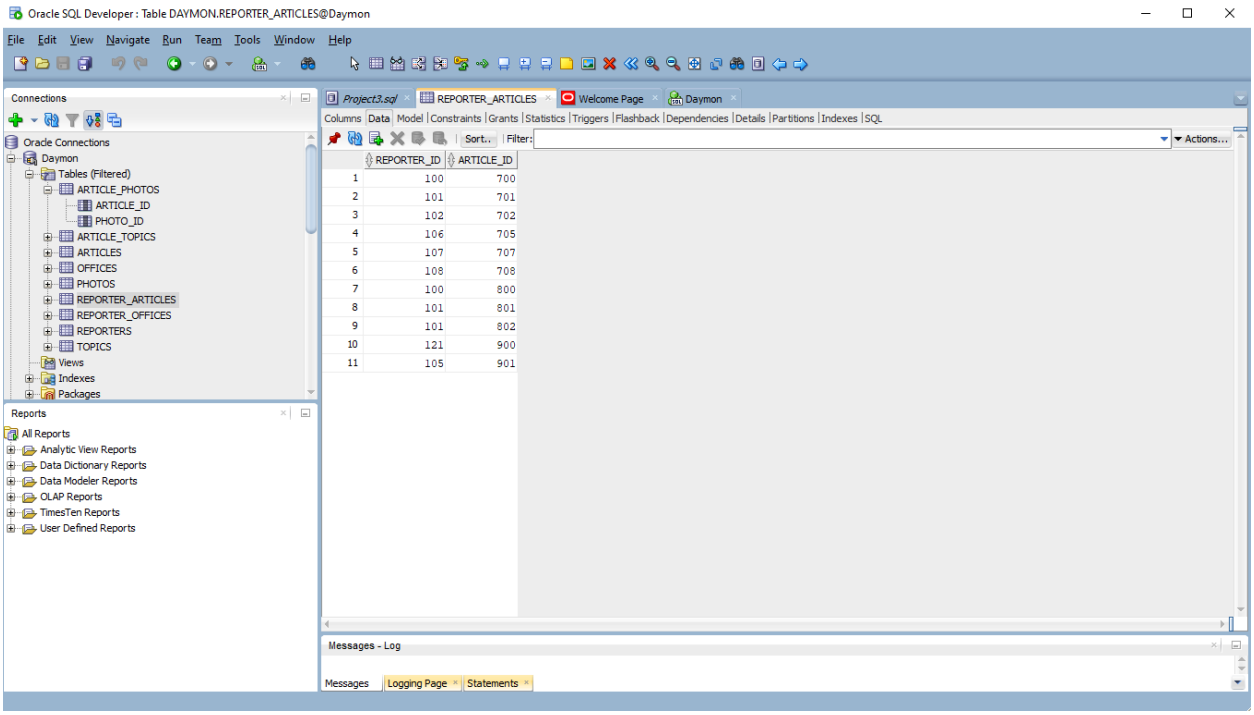

File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
Daymon
Tables (Filtered)
ARTICLE_PHOTOS
ARTICLE_ID
PHOTO_ID
ARTICLE_TOPICS
ARTICLES
OFFICES
PHOTOS
REPORTER_ARTICLES
REPORTER_OFFICES
REPORTERS
TOPICS
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Project3.sql
REPORTER_OFFICES
Welcome Page
Columns Data Model Constraints Grants Statistics Triggers Flashback
Actions...
DAYMON.REPORTER_OFFICES
P * REPORTER_ID NUMBER
PF* OFFICE_ID NUMBER
REPORTER_OFFICES_PK (REPORTER_ID, OFFICE_ID)
SYS_C0010608 (OFFICE_ID)
DAYMON.OFFICES
P * OFFICE_ID NUMBER
* OFFICE_NAME VARCHAR2 (50 BYTE)
* OFFICE_LOCATION VARCHAR2 (50 BYTE)
OFFICES_PK (OFFICE_ID)

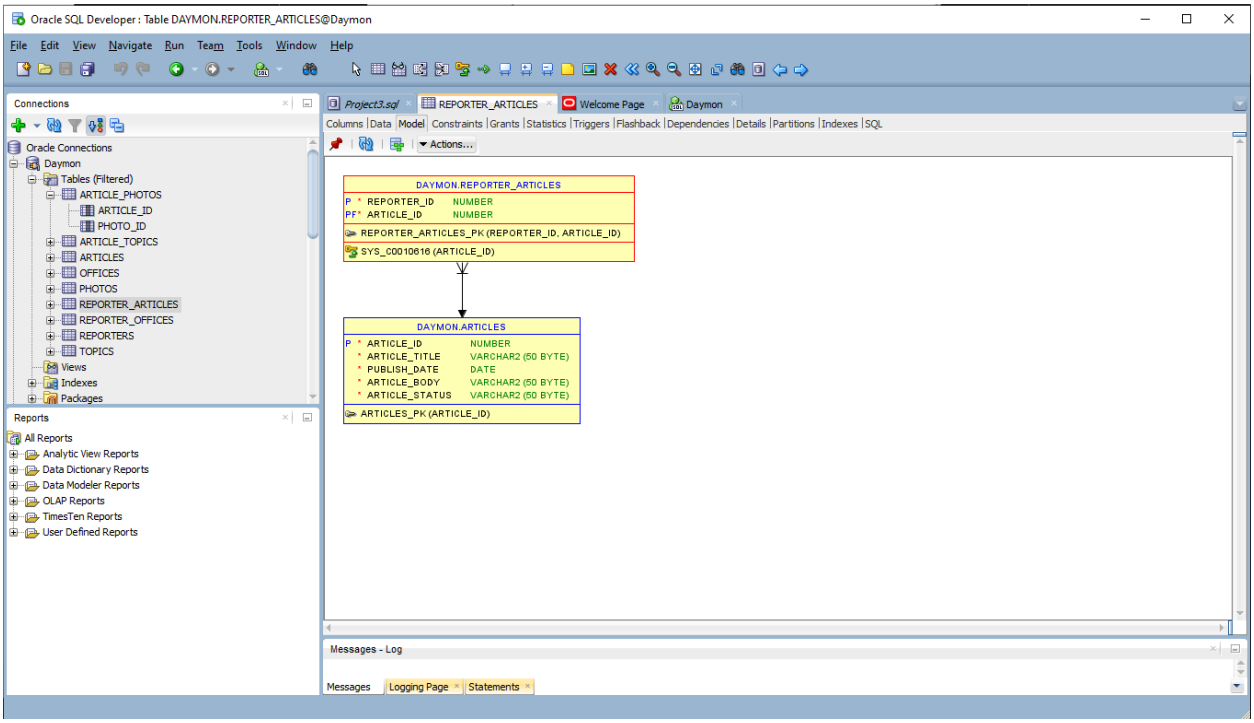
```
CREATE TABLE articles (
    article_id NUMBER PRIMARY KEY,
    article_title VARCHAR2(50) NOT NULL,
    publish_date DATE NOT NULL,
    article_body VARCHAR2(50) NOT NULL,
    article_status VARCHAR2(50) NOT NULL
)
```



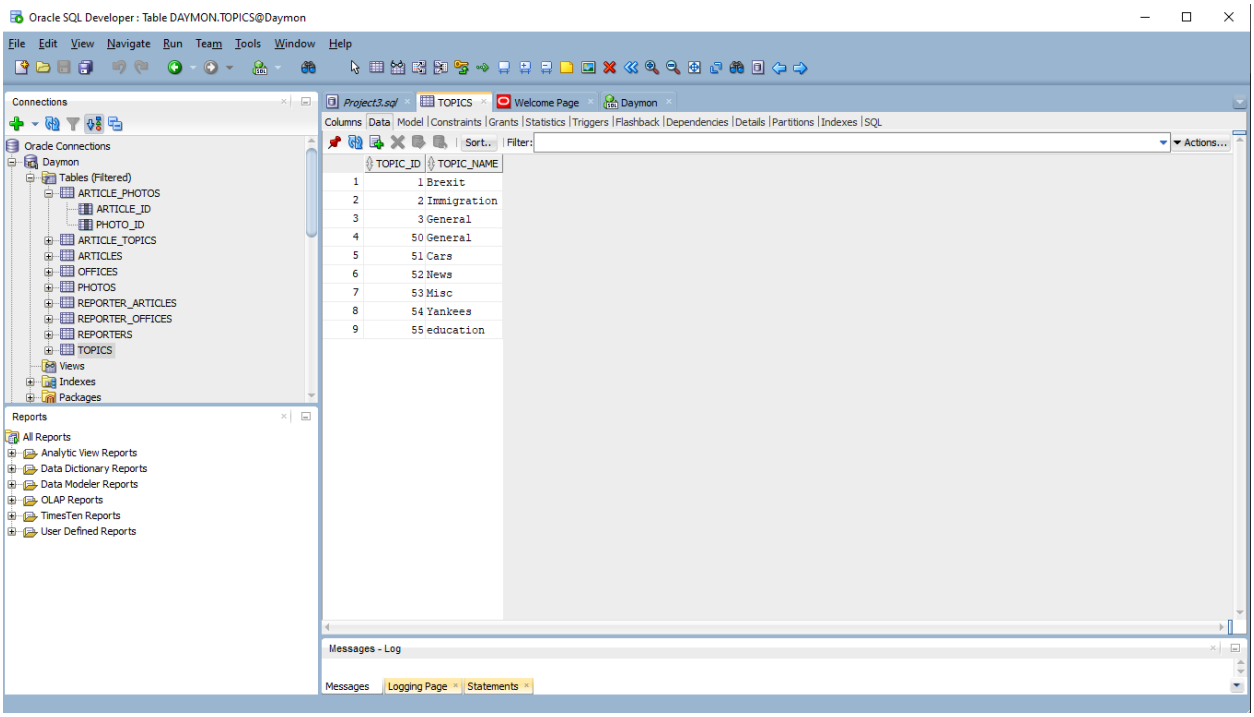


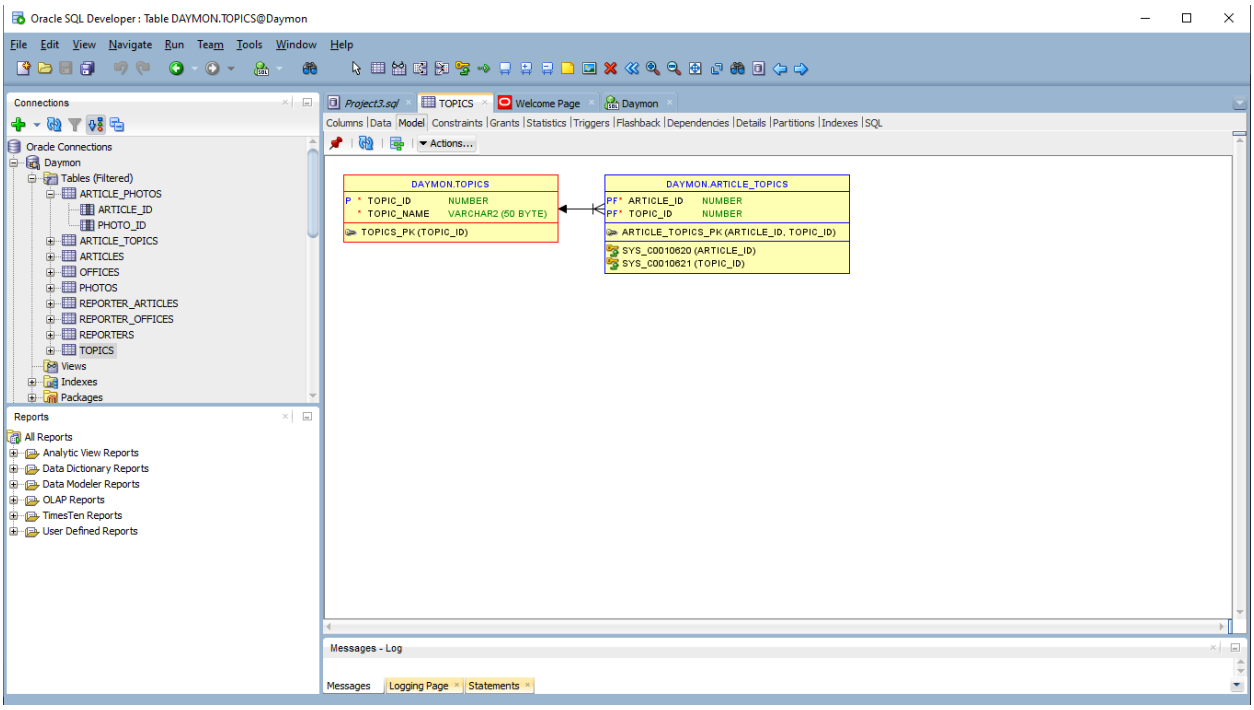

```
CREATE TABLE reporter_articles (
    reporter_id NUMBER,
    article_id NUMBER,
    PRIMARY KEY (reporter_id, article_id),
    FOREIGN KEY (reporter_id) REFERENCES reporters (reporter_id),
    FOREIGN KEY (article_id) REFERENCES articles (article_id)
)
```


Oracle SQL Developer : Table DAYMON.REPORTER_ARTICLES@Daymon
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
Daymon
Tables (Filtered)
ARTICLE_PHOTOS
ARTICLE_ID
PHOTO_ID
ARTICLE_TOPICS
ARTICLES
OFFICES
PHOTOS
REPORTER_ARTICLES
REPORTER_OFFICES
REPORTERS
TOPICS
Views
Indexes
Packages
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Project3.sql
REPORTER_ARTICLES
Welcome Page
Daymon
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Sort..
Filter:
Actions...
REPORTER_ID ARTICLE_ID
1 100 700
2 101 701
3 102 702
4 106 705
5 107 707
6 108 708
7 100 800
8 101 801
9 101 802
10 121 900
11 105 901
Messages - Log
Messages
Logging Page
Statements

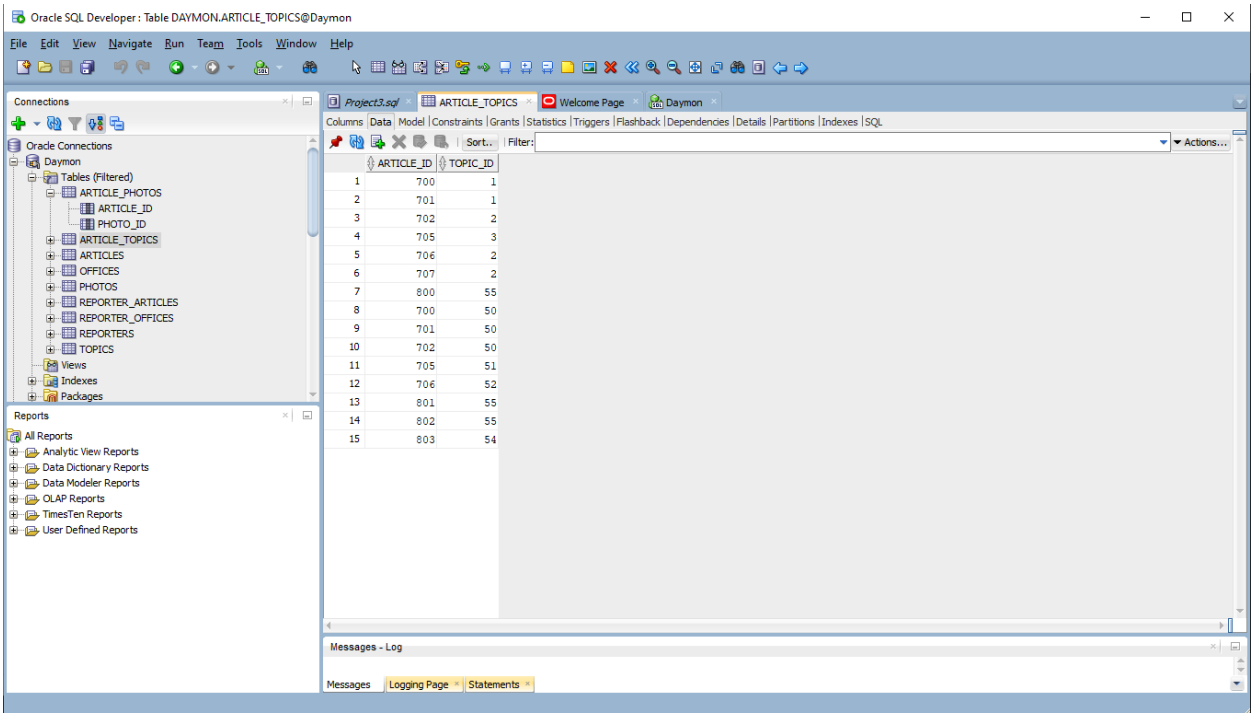

Oracle SQL Developer : Table DAYMON.REPORTER_ARTICLES@Daymon
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
Daymon
Tables (Filtered)
ARTICLE_PHOTOS
ARTICLE_ID
PHOTO_ID
ARTICLE_TOPICS
ARTICLES
OFFICES
PHOTOS
REPORTER_ARTICLES
REPORTER_OFFICES
REPORTERS
TOPICS
Views
Indexes
Packages
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Project3.sql
REPORTER_ARTICLES
Welcome Page
Daymon
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Actions...
DAYMON.REPORTER_ARTICLES
P * REPORTER_ID NUMBER
PF* ARTICLE_ID NUMBER
REPORTER_ARTICLES_PK (REPORTER_ID, ARTICLE_ID)
SYS_C0010616 (ARTICLE_ID)
DAYMON.ARTICLES
P * ARTICLE_ID NUMBER
* ARTICLE_TITLE VARCHAR2 (50 BYTE)
* PUBLISH_DATE DATE
* ARTICLE_BODY VARCHAR2 (50 BYTE)
* ARTICLE_STATUS VARCHAR2 (50 BYTE)
ARTICLES_PK (ARTICLE_ID)
Messages - Log
Messages
Logging Page
Statements

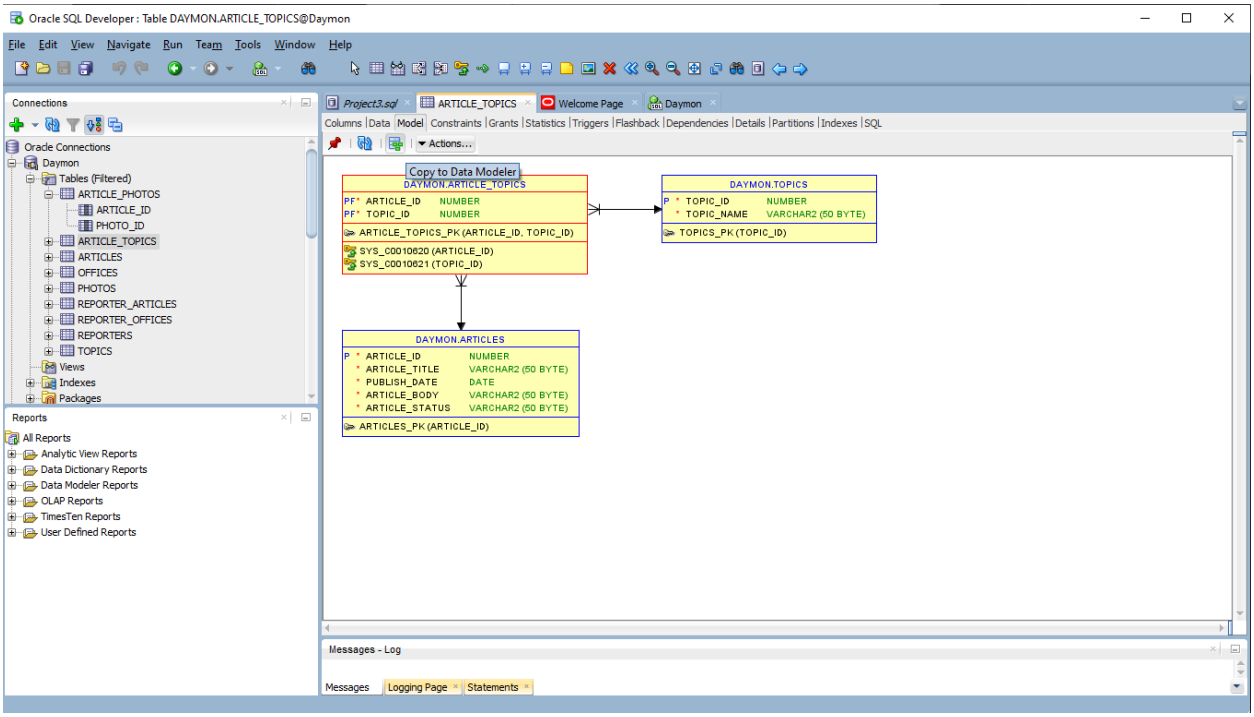
```
CREATE TABLE topics (
    topic_id NUMBER PRIMARY KEY,
    topic_name VARCHAR2(50) NOT NULL
)
```



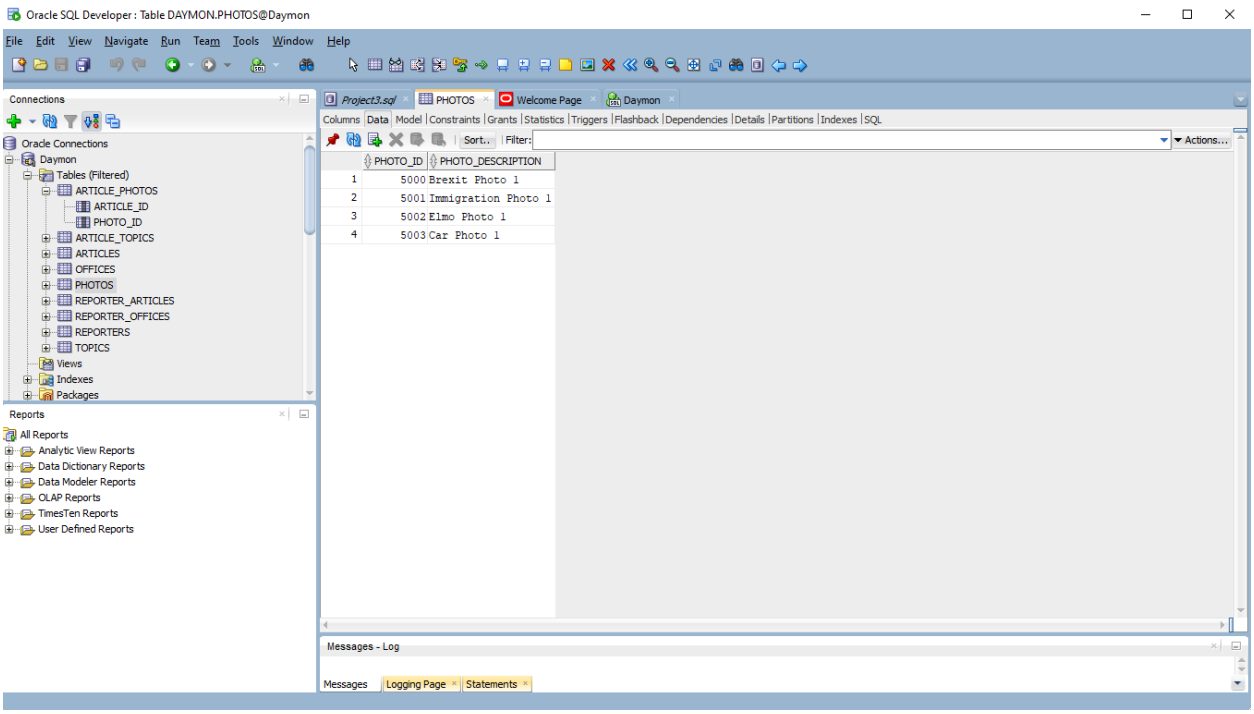


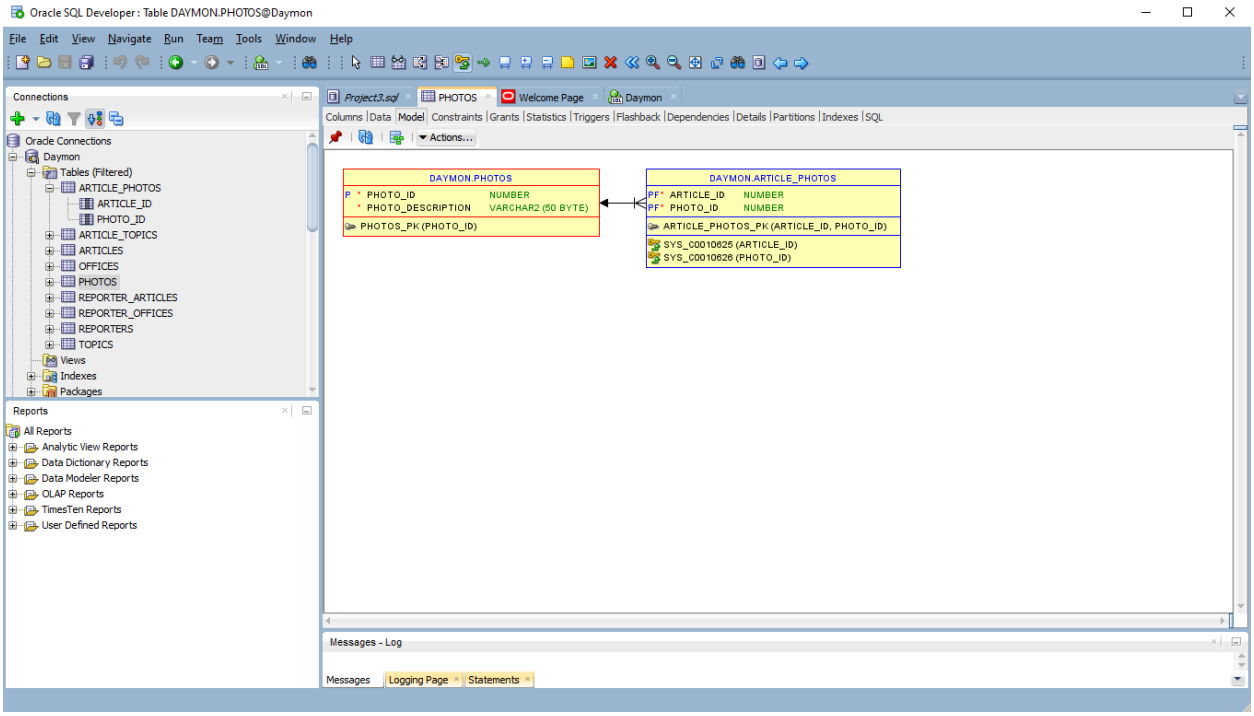
```
CREATE TABLE article_topics (
    article_id NUMBER,
    topic_id NUMBER,
    PRIMARY KEY (article_id, topic_id),
    FOREIGN KEY (article_id) REFERENCES articles (article_id),
    FOREIGN KEY (topic_id) REFERENCES topics (topic_id)
)
```



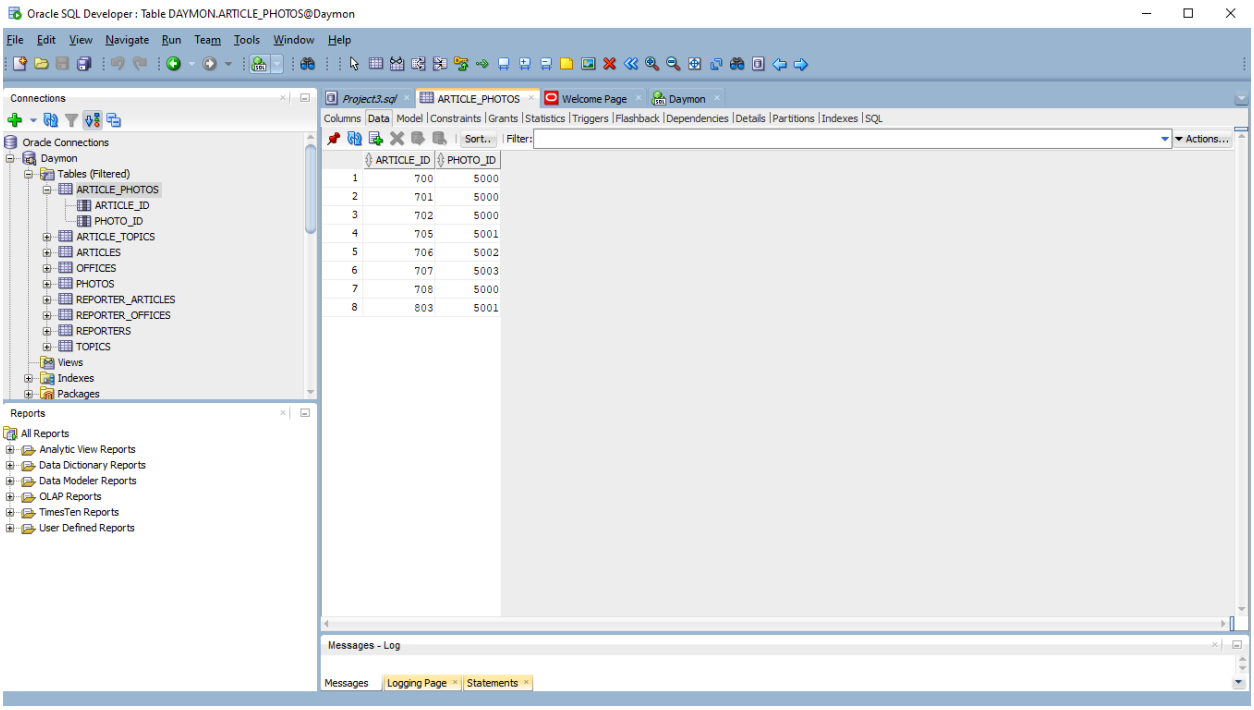


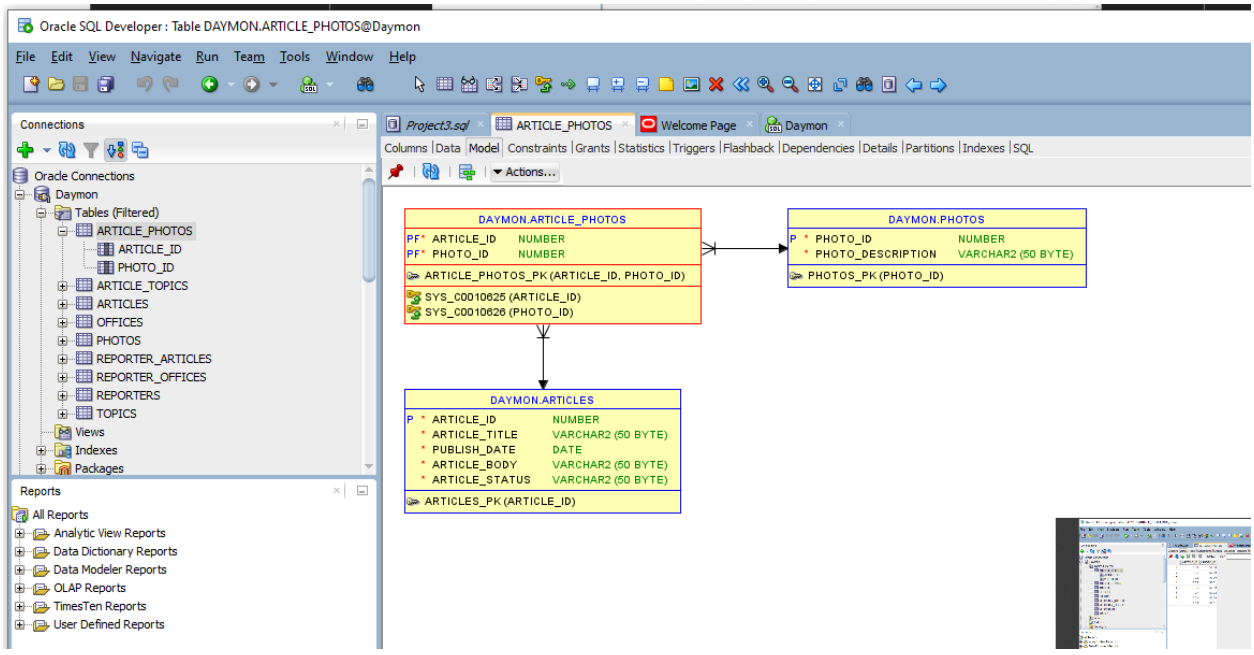

```
CREATE TABLE photos (
    photo_id NUMBER PRIMARY KEY,
    photo_description VARCHAR2(50) NOT NULL
)
```






```
CREATE TABLE article_photos (
    article_id NUMBER,
    photo_id NUMBER,
    PRIMARY KEY (article_id, photo_id),
    FOREIGN KEY (article_id) REFERENCES articles (article_id),
    FOREIGN KEY (photo_id) REFERENCES photos (photo_id)
)
```

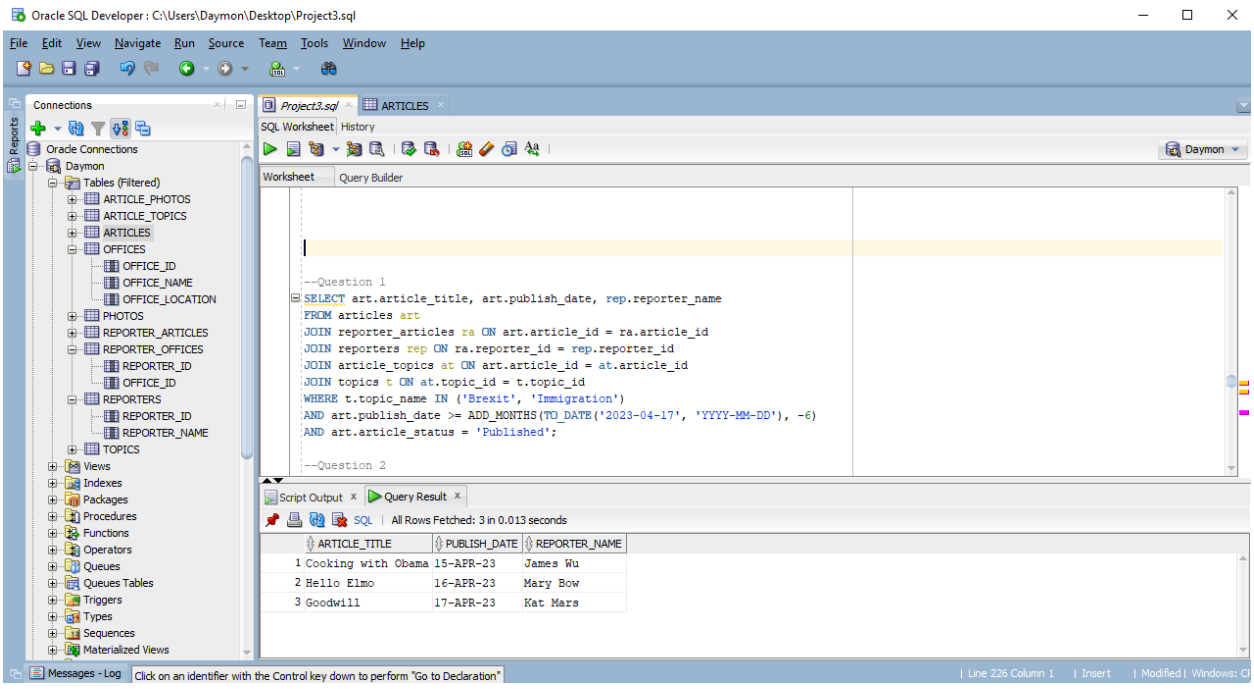

Oracle SQL Developer : Table DAYMON.ARTICLE_PHOTOS@Daymon
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
Daymon
Tables (Filtered)
ARTICLE_PHOTOS
ARTICLE_ID
PHOTO_ID
ARTICLE_TOPICS
ARTICLES
OFFICES
PHOTOS
REPORTER_ARTICLES
REPORTER_OFFICES
REPORTERS
TOPICS
Views
Indexes
Packages
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Project3.sql
ARTICLE_PHOTOS
Welcome Page
Daymon
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Sort..
Filter:
Actions...
ARTICLE_ID PHOTO_ID
1 700 5000
2 701 5000
3 702 5000
4 705 5001
5 706 5002
6 707 5003
7 708 5000
8 803 5001
Messages - Log
Messages
Logging Page
Statements


Oracle SQL Developer : Table DAYMON.ARTICLE_PHOTOS@Daymon
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
Daymon
Tables (Filtered)
ARTICLE_PHOTOS
ARTICLE_ID
PHOTO_ID
ARTICLE_TOPICS
ARTICLES
OFFICES
PHOTOS
REPORTER_ARTICLES
REPORTER_OFFICES
REPORTERS
TOPICS
Views
Indexes
Packages
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Project3.sql
ARTICLE_PHOTOS
Welcome Page
Daymon
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Actions...
DAYMON.ARTICLE_PHOTOS
PF* ARTICLE_ID NUMBER
PF* PHOTO_ID NUMBER
ARTICLE_PHOTOS_PK (ARTICLE_ID, PHOTO_ID)
SYS_C0010625 (ARTICLE_ID)
SYS_C0010626 (PHOTO_ID)
DAYMON.PHOTOS
P * PHOTO_ID NUMBER
* PHOTO_DESCRIPTION VARCHAR2 (50 BYTE)
PHOTOS_PK (PHOTO_ID)
DAYMON.ARTICLES
P * ARTICLE_ID NUMBER
* ARTICLE_TITLE VARCHAR2 (50 BYTE)
* PUBLISH_DATE DATE
* ARTICLE_BODY VARCHAR2 (50 BYTE)
* ARTICLE_STATUS VARCHAR2 (50 BYTE)
ARTICLES_PK (ARTICLE_ID)

1. Identify published articles in the last six months about Brexit and immigration.

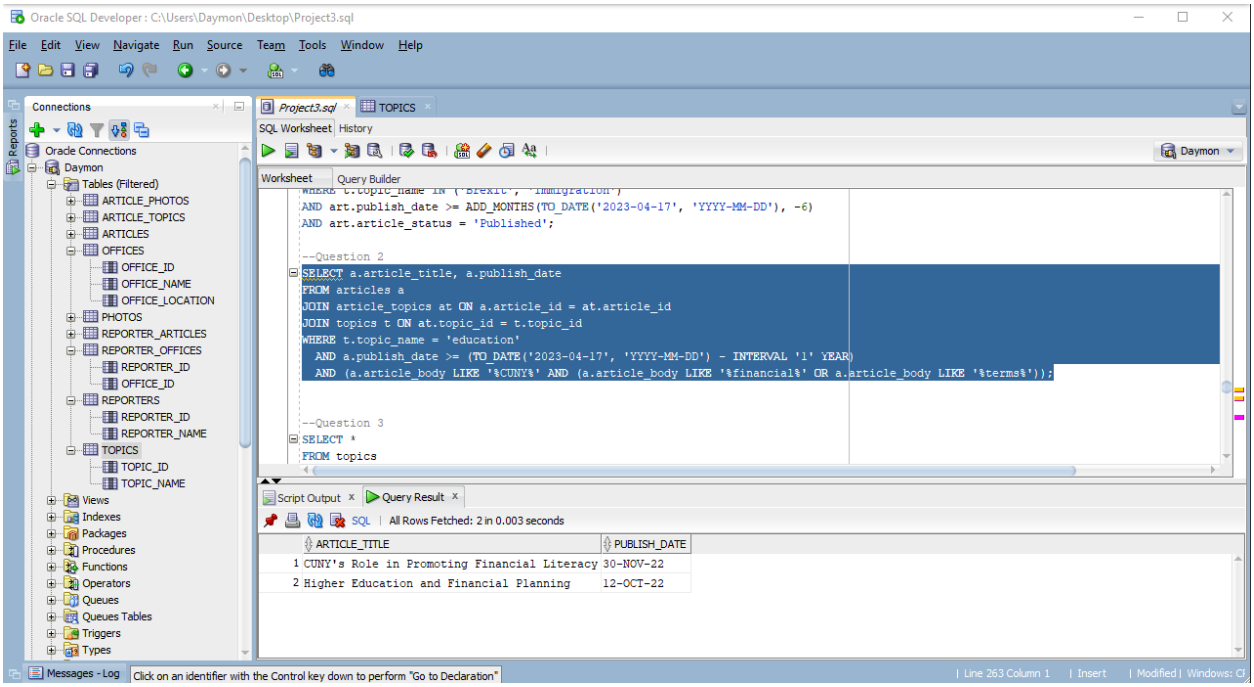
Display the article title, publish date and reporters.

```
SELECT art.article_title, art.publish_date, rep.reporter_name
FROM articles art
JOIN reporter_articles ra ON art.article_id = ra.article_id
JOIN reporters rep ON ra.reporter_id = rep.reporter_id
JOIN article_topics at ON art.article_id = at.article_id
JOIN topics t ON at.topic_id = t.topic_id
WHERE t.topic_name IN ('Brexit', 'Immigration')
AND art.publish_date >= ADD_MONTHS(TO_DATE('2023-04-17', 'YYYY-MM-DD'), -6)
AND art.article_status = 'Published';
```



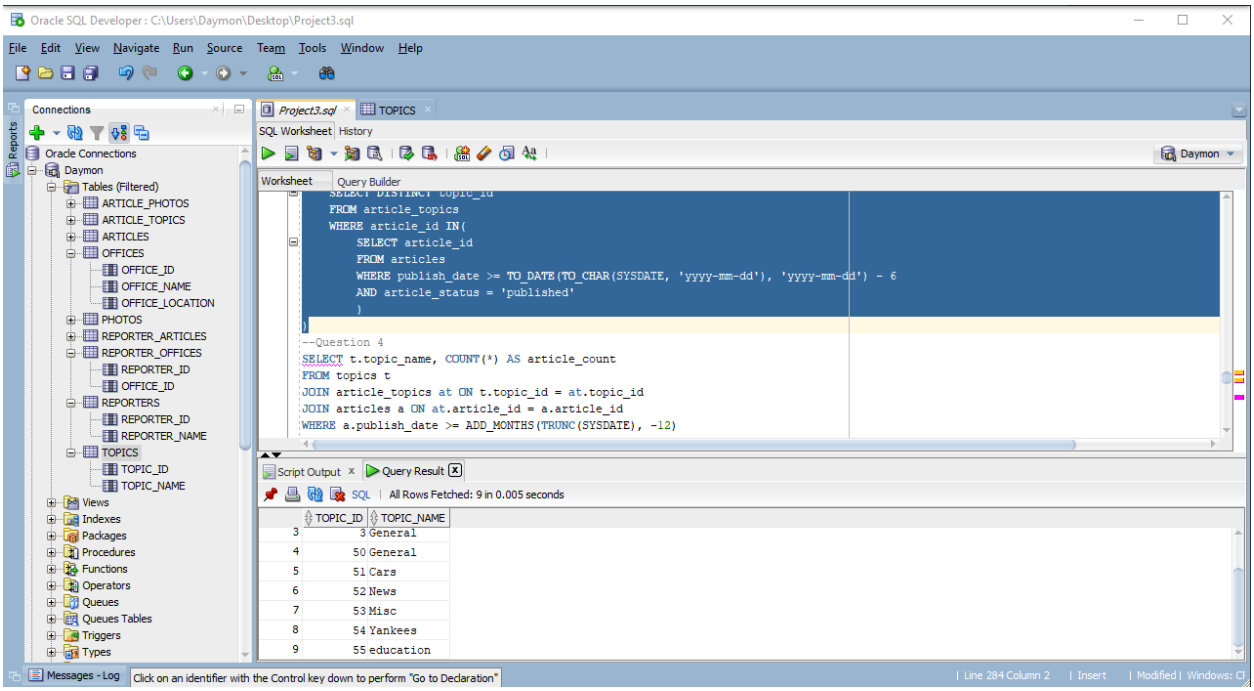
2. Identify articles in the last year about the topic education and the article body includes references to CUNY and financial terms. Display the article title and publish date. Use a nested select to answer this question. Order the articles. chronologically.

```
SELECT a.article_title, a.publish_date
FROM articles a
JOIN article_topics at ON a.article_id = at.article_id
JOIN topics t ON at.topic_id = t.topic_id
WHERE t.topic_name = 'education'
AND a.publish_date >= TO_DATE('2023-04-17', 'YYYY-MM-DD') - INTERVAL '1' YEAR
AND a.article_body LIKE '%CUNY%'
AND (a.article_body LIKE '%financial%' OR  a.article_body LIKE '%terms%');
```



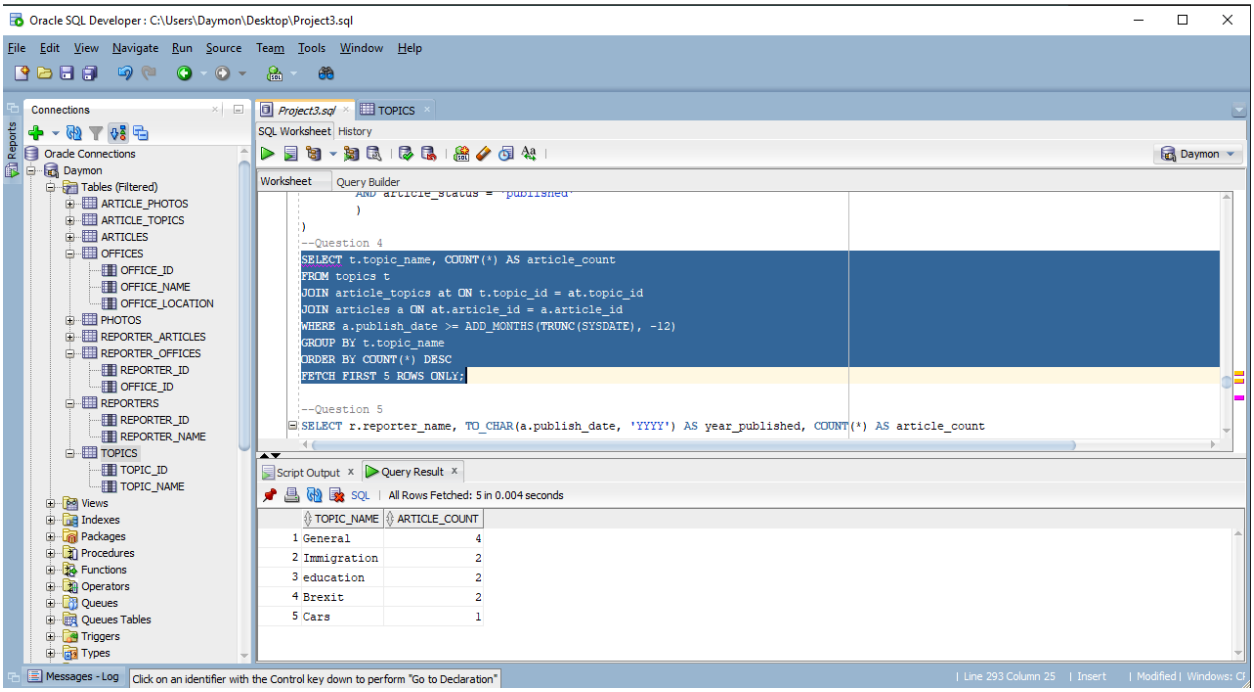
3. Identify topics not assigned to published articles in the last six months. Display the topics. Use a nested select to answer this question.

```
SELECT *
FROM topics
WHERE topic_id NOT IN(
   SELECT DISTINCT topic_id
   FROM article_topics
   WHERE article_id IN(
      SELECT article_id
      FROM articles
      WHERE publish_date >= TO_DATE(TO_CHAR(SYSDATE, 'yyyy-mm-dd'), 'yyyy-mm-dd') - 6
      AND article_status = 'published'
      )
)
```



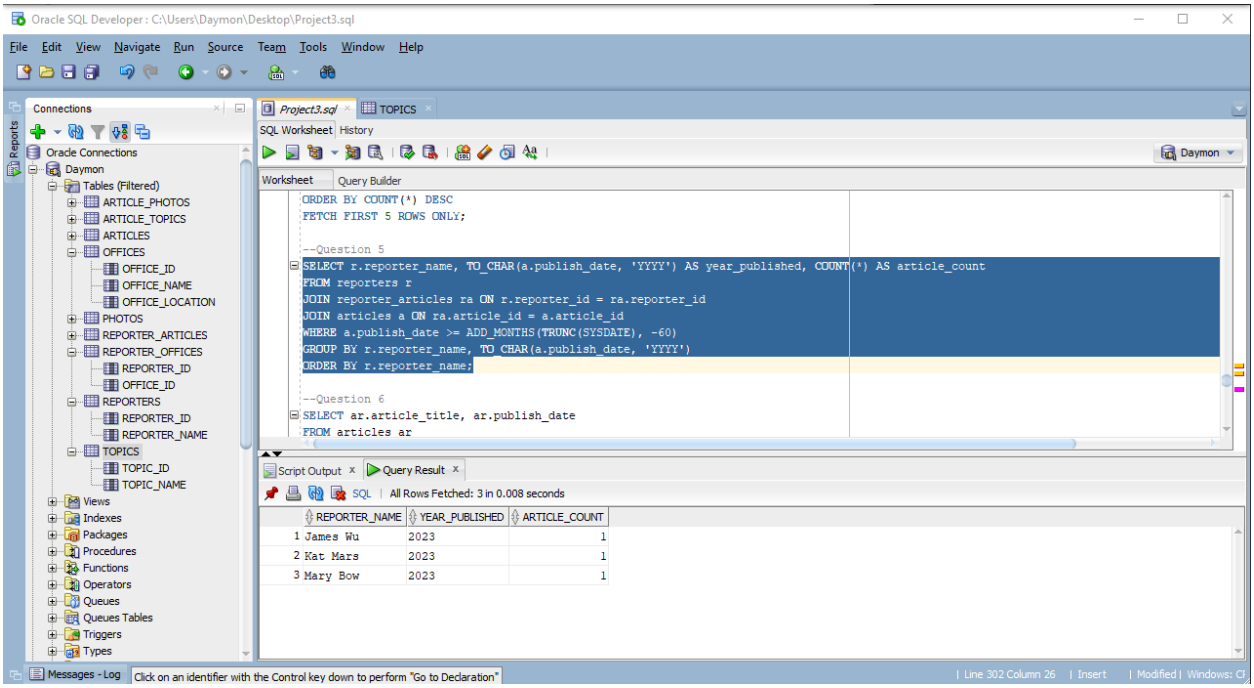
4. Identify the top five topics in the last year. Display two columns: topic and number of articles. Display the topic with the most articles first. Display one row. for each distinct topic. Use a function to answer this question.

```
SELECT t.topic_name, COUNT(*) AS article_count
FROM topics t, article_topics at, articles a
WHERE t.topic_id = at.topic_id
AND at.article_id = a.article_id
AND a.publish_date >= ADD_MONTHS(TO_DATE('2023-05-18', 'YYYY-MM-DD'), -12)
GROUP BY t.topic_name
ORDER BY COUNT(*) DESC
FETCH FIRST 5 ROWS ONLY;
```



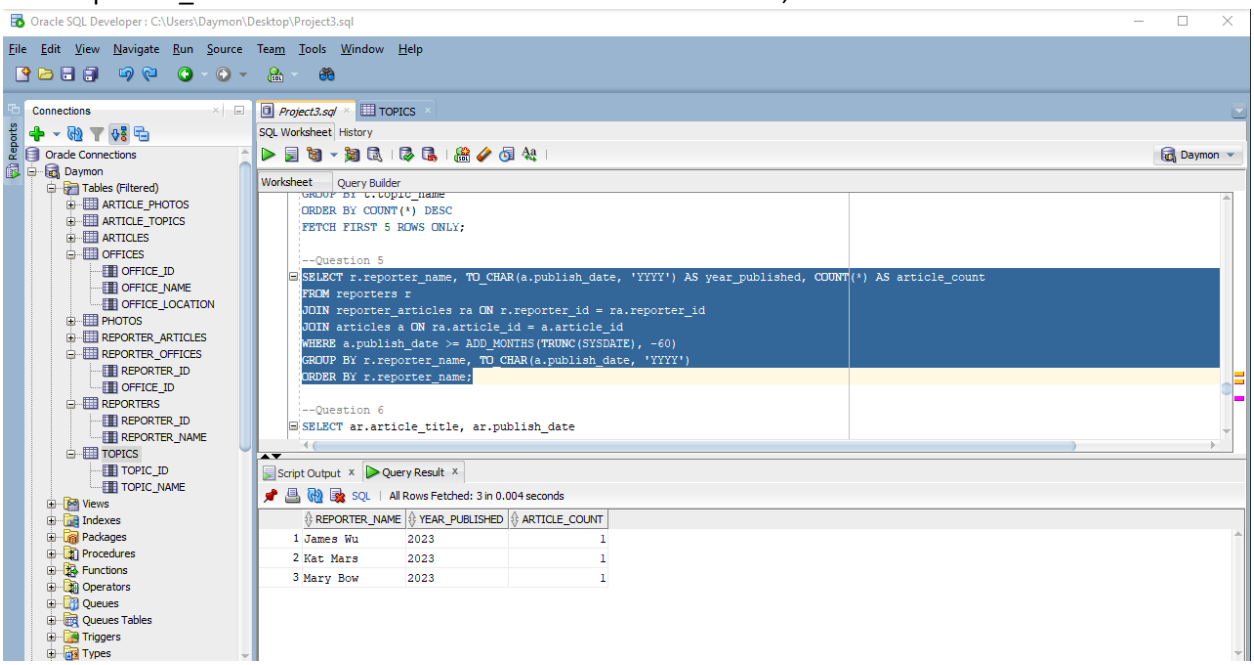
5. Identify the number of articles by reporter in the last five years. Display three columns: Reporter, year published and number of articles. Display one row for each distinct reporter and year. Order the output by reporter name. Use a function to answer this question.

```
SELECT r.reporter_name, TO_CHAR(a.publish_date, 'YYYY') AS year_published, COUNT(*) AS article_count
FROM reporters r, reporter_articles ra, articles a
WHERE r.reporter_id = ra.reporter_id
AND ra.article_id = a.article_id
AND a.publish_date >= ADD_MONTHS(TRUNC(SYSDATE), -60)
GROUP BY r.reporter_name, TO_CHAR(a.publish_date, 'YYYY')
ORDER BY r.reporter_name;
```



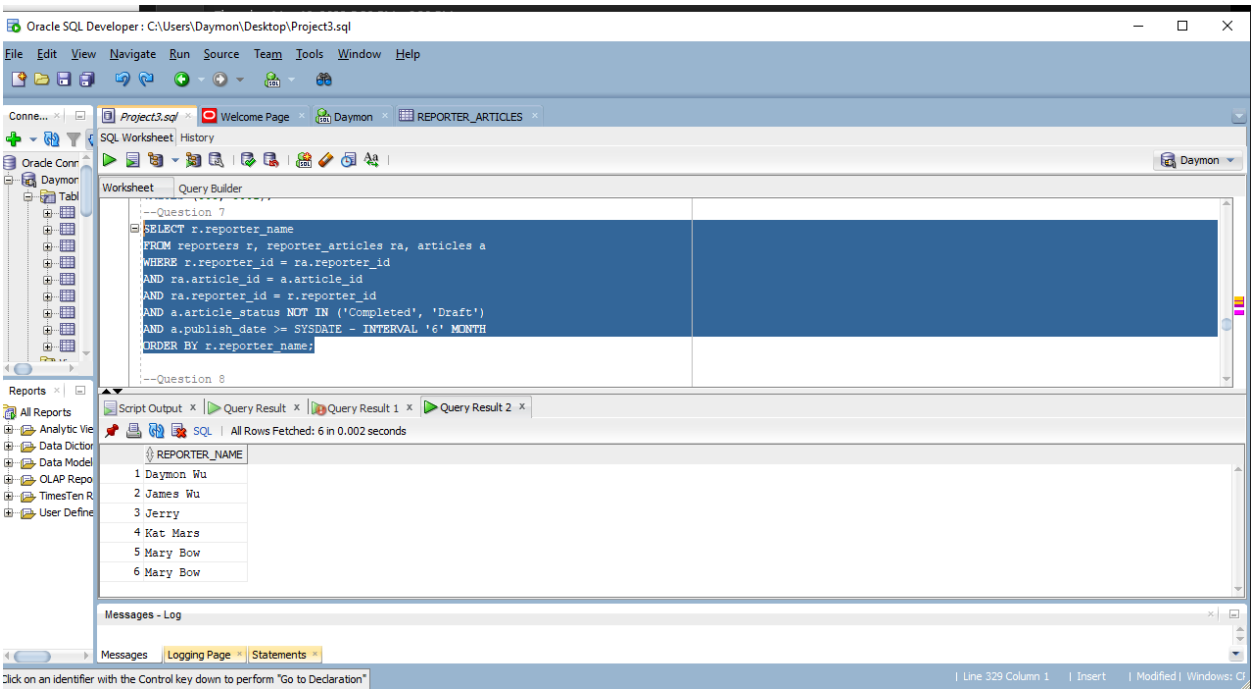
6. Identify photo only articles about the Yankees in the last six months. Display the article title and publish date. Use a nested select to answer this question. Note, a photo only article has no text in the body and a photo.

```
SELECT ar.article_title, ar.publish_date
FROM articles ar, article_photos ap, photos ph, article_topics at, topics tp
WHERE ar.article_id = ap.article_id
AND ap.photo_id = ph.photo_id
AND ap.article_id = at.article_id
AND at.topic_id = tp.topic_id
AND tp.topic_name = 'Yankees'
AND ar.publish_date >= DATE '2023-05-17' - INTERVAL '6' MONTH;
```



7. Identify New York reporters with no completed or draft articles in the last six months. Display the reporter. Order the output by reporter name. Use a nested select to answer this question.

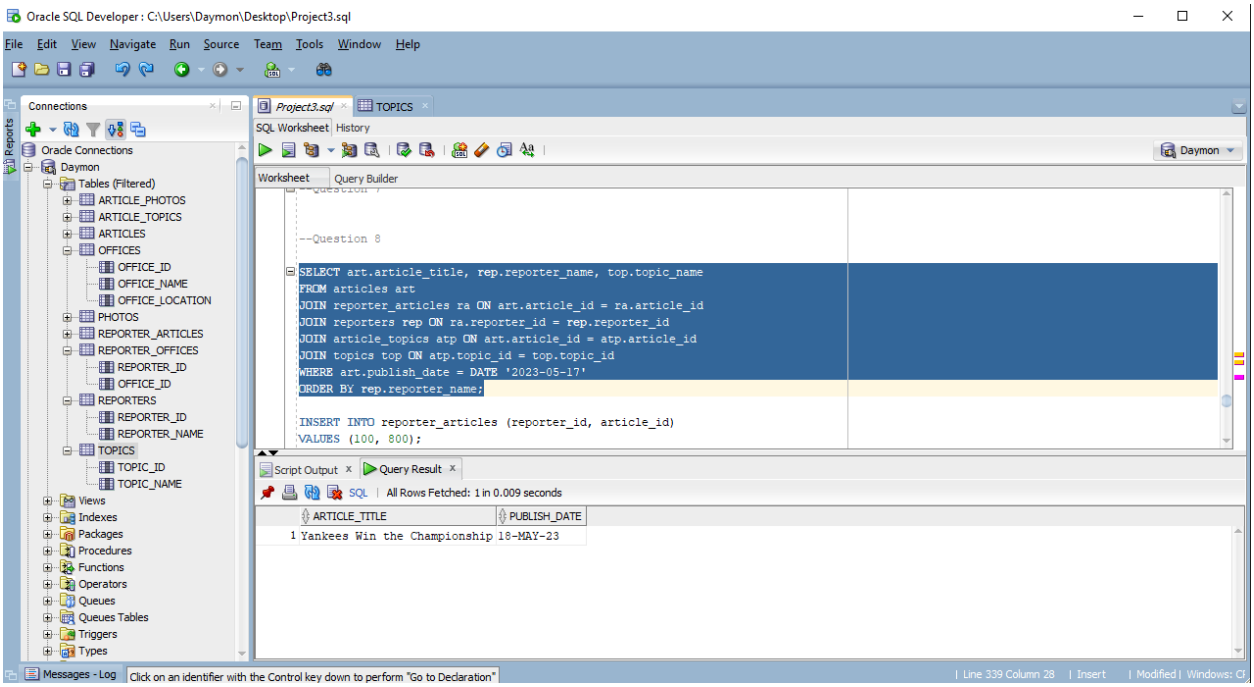
```
SELECT r.reporter_name
FROM reporters r, reporter_articles ra, articles a
WHERE r.reporter_id = ra.reporter_id
AND ra.article_id = a.article_id
AND ra.reporter_id = r.reporter_id
AND a.article_status NOT IN ('Completed', 'Draft')
AND a.publish_date >= SYSDATE - INTERVAL '6' MONTH
ORDER BY r.reporter_name;
```



8. Identify all articles published today. Display the article title, authors and topics.

Order the output by author.

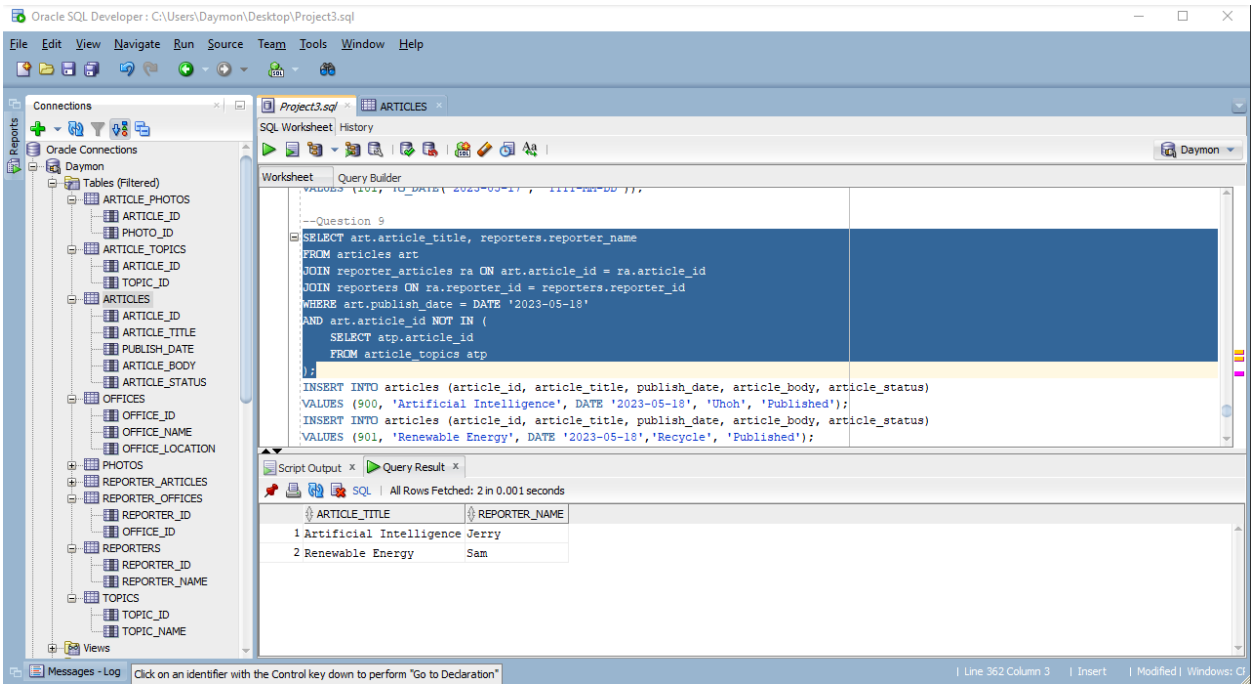
```
SELECT art.article_title, rep.reporter_name, top.topic_name
FROM articles art
JOIN reporter_articles ra ON art.article_id = ra.article_id
JOIN reporters rep ON ra.reporter_id = rep.reporter_id
JOIN article_topics atp ON art.article_id = atp.article_id
JOIN topics top ON atp.topic_id = top.topic_id
WHERE art.publish_date = DATE '2023-05-17'
ORDER BY rep.reporter_name;
```



9. Identify articles published today without a topic. Display the article and authors.

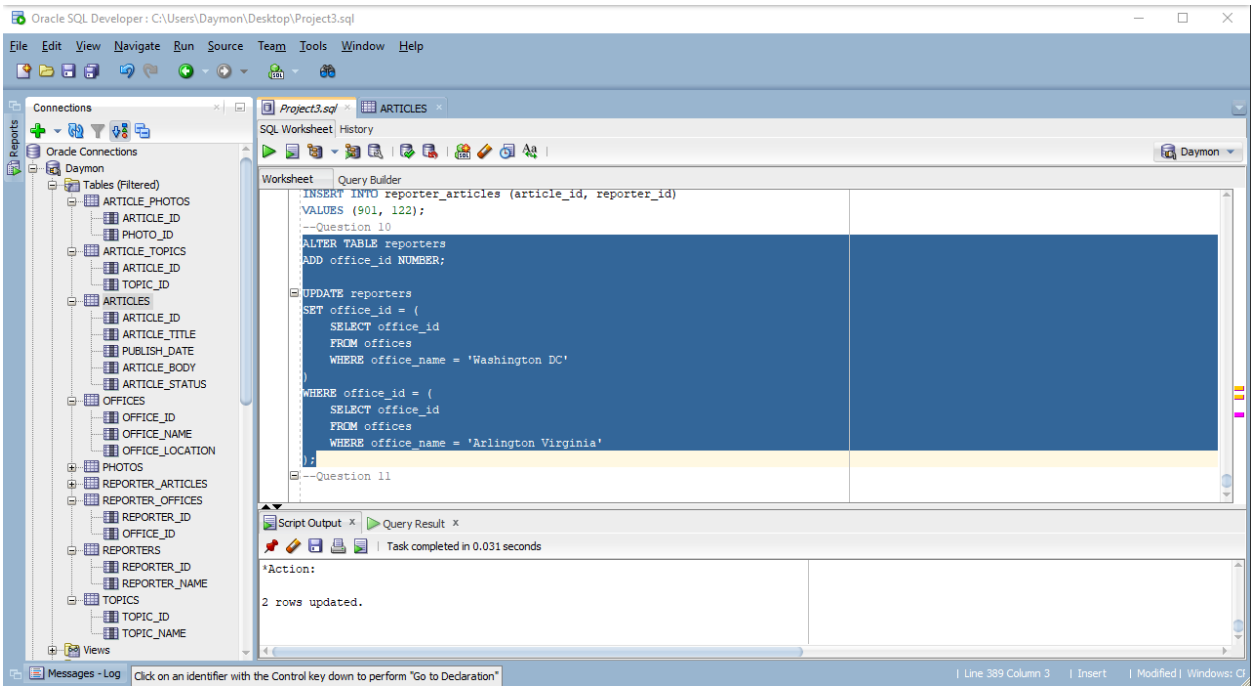
Use a nested select to answer this question.

```
SELECT art.article_title, reporters.reporter_name
FROM articles art
JOIN reporter_articles ra ON art.article_id = ra.article_id
JOIN reporters ON ra.reporter_id = reporters.reporter_id
WHERE art.publish_date = DATE '2023-05-18'
AND art.article_id NOT IN (
   SELECT atp.article_id
   FROM article_topics atp
);
```



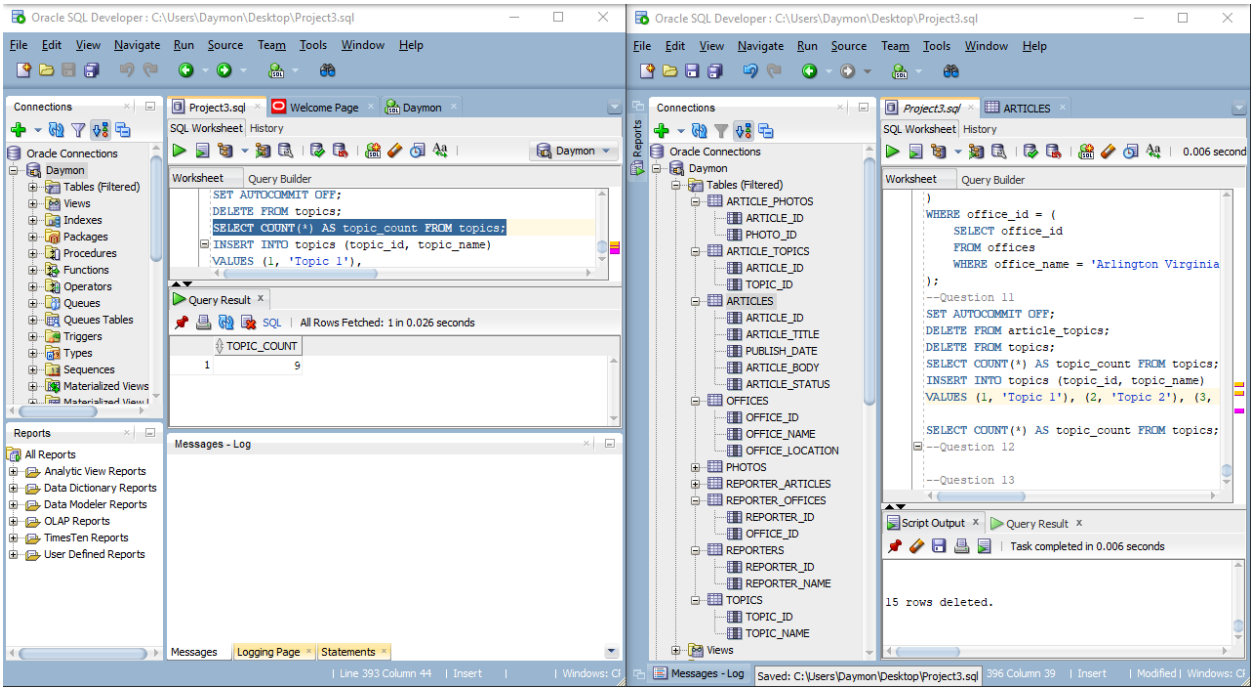
10. Close the Arlington Virginia office and re-assign all reporters to the Washington DC office. Identify the SQL code to implement.

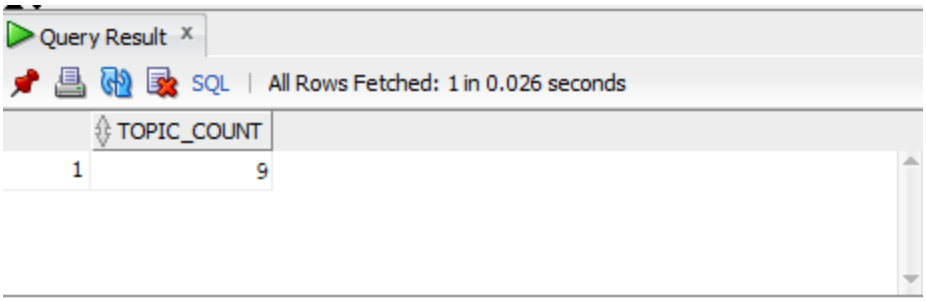
```
ALTER TABLE reporters
ADD office_id NUMBER;


UPDATE reporters
SET office_id = (
   SELECT office_id
   FROM offices
   WHERE office_name = 'Washington DC'
)
WHERE office_id = (
   SELECT office_id
   FROM offices
   WHERE office_name = 'Arlington Virginia'
);
```

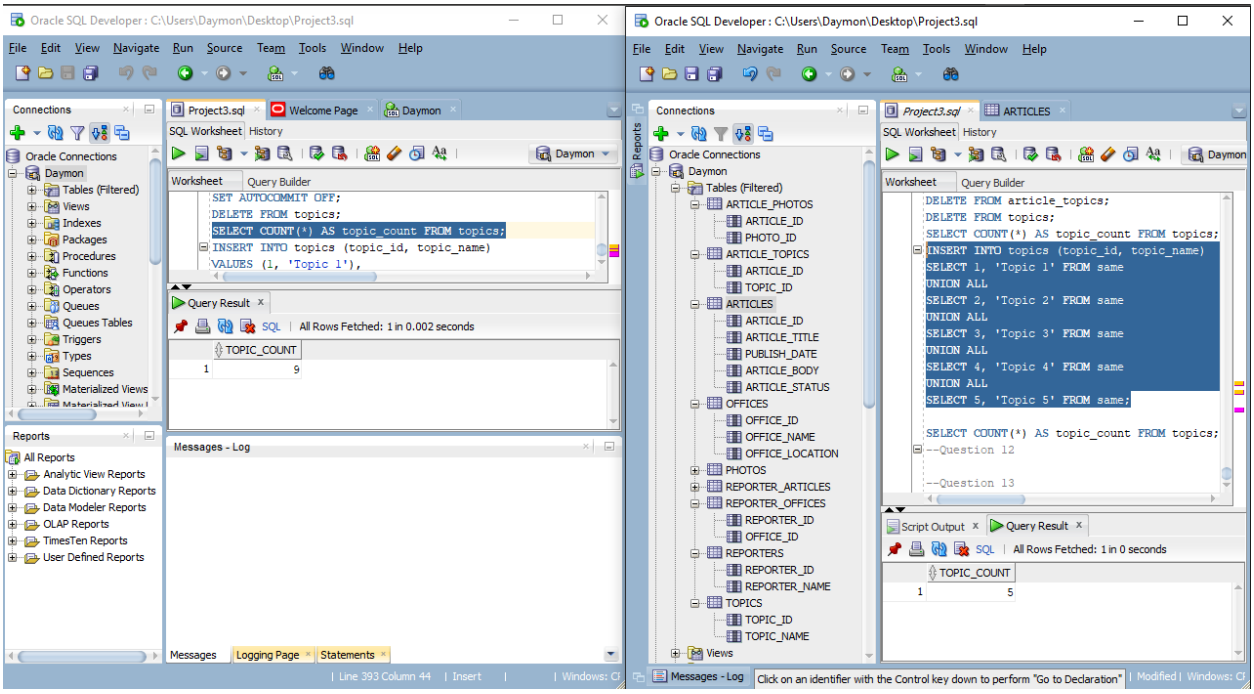


11. In one SQL window, delete all topics. Don't commit. In another SQL window, create five new topics. Don't commit. In each SQL window, identify the number of topics. Explain your results. Disable the auto commit flag at the top of the window before performing this operation. Show all SQL to perform these operations.

```
SET AUTOCOMMIT OFF;
DELETE FROM article_topics;
DELETE FROM topics;
SELECT COUNT(*) AS topic_count FROM topics;
INSERT INTO topics (topic_id, topic_name)
SELECT 1, 'Topic 1' FROM same UNION ALL
SELECT 2, 'Topic 2' FROM same UNION ALL
SELECT 3, 'Topic 3' FROM same UNION ALL
SELECT 4, 'Topic 4' FROM same UNION ALL
SELECT 5, 'Topic 5' FROM same;
SELECT COUNT(*) AS topic_count FROM topics;
```







The results are not the same because the results have not been committed.

12. In one SQL window, delete all articles written by reporter Daymon Wu. Don't

commit. In another SQL window, reassign all articles written by Daymon Wu to Sam. Don't commit. Explain your results. Resolve the problem. Create a backup ofyour table before implementing. To create a backup table, enter CREATE TABLE <NEWTABLE> AS SELECT * FROM <ORIGINALTABLE>; COMMIT; Then you can rename a table using the RENAME TABLE commit. Disable the auto commit flag at the top of the window before performing this operation. Show all SQL to perform these operations.

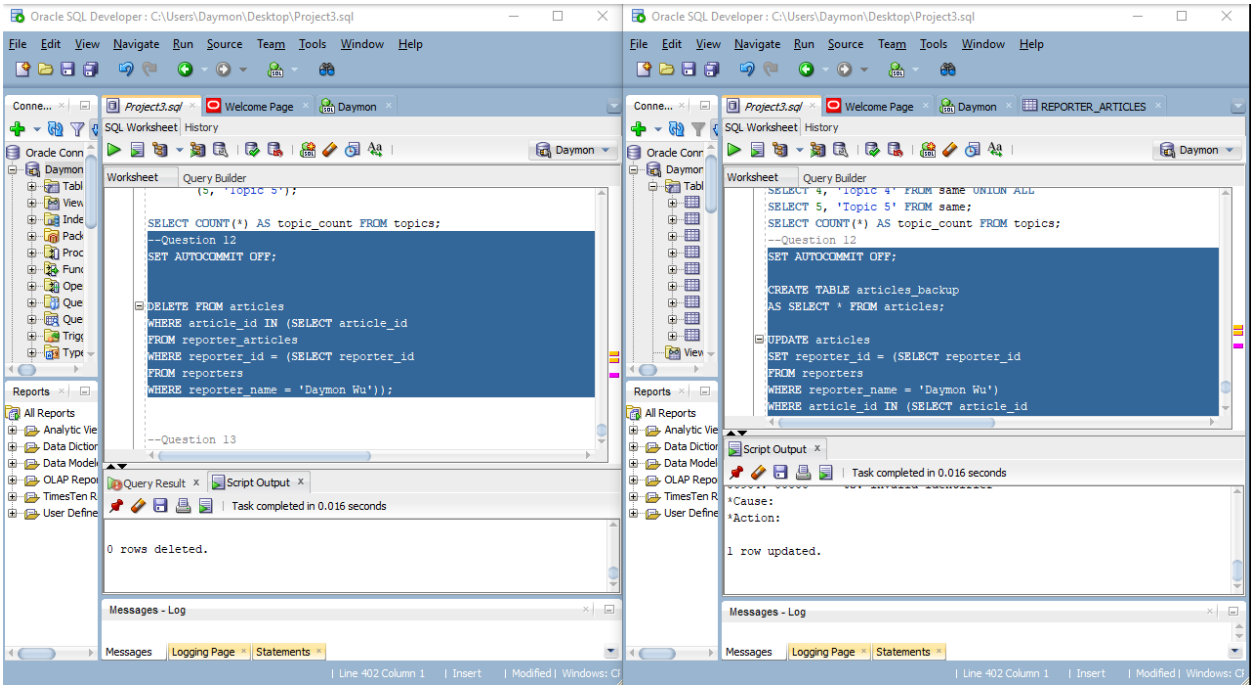
Window 1:

```
--Question 12

SET AUTOCOMMIT OFF;

DELETE FROM articles

WHERE article_id IN (SELECT article_id

FROM reporter_articles

WHERE reporter_id = (SELECT reporter_id

FROM reporters

WHERE reporter_name = 'Daymon Wu'));
```

Window 2:

```
SET AUTOCOMMIT OFF;

CREATE TABLE articles_backup

AS SELECT * FROM articles;

UPDATE articles

SET reporter_id = (SELECT reporter_id

FROM reporters

WHERE reporter_name = 'Daymon Wu')

WHERE article_id IN (SELECT article_id

FROM reporter_articles

WHERE reporter_id = (SELECT reporter_id

FROM reporters

WHERE reporter_name = 'Sam'));


UPDATE reporter_articles
```

```
SET reporter_id = (SELECT reporter_id
FROM reporters
WHERE reporter_name = 'Daymon Wu')
WHERE reporter_id = (SELECT reporter_id
FROM reporters
WHERE reporter_name = 'Sam');

DELETE FROM articles;
INSERT INTO articles SELECT * FROM articles_backup;
COMMIT;

DROP TABLE articles_backup;
COMMIT;
```

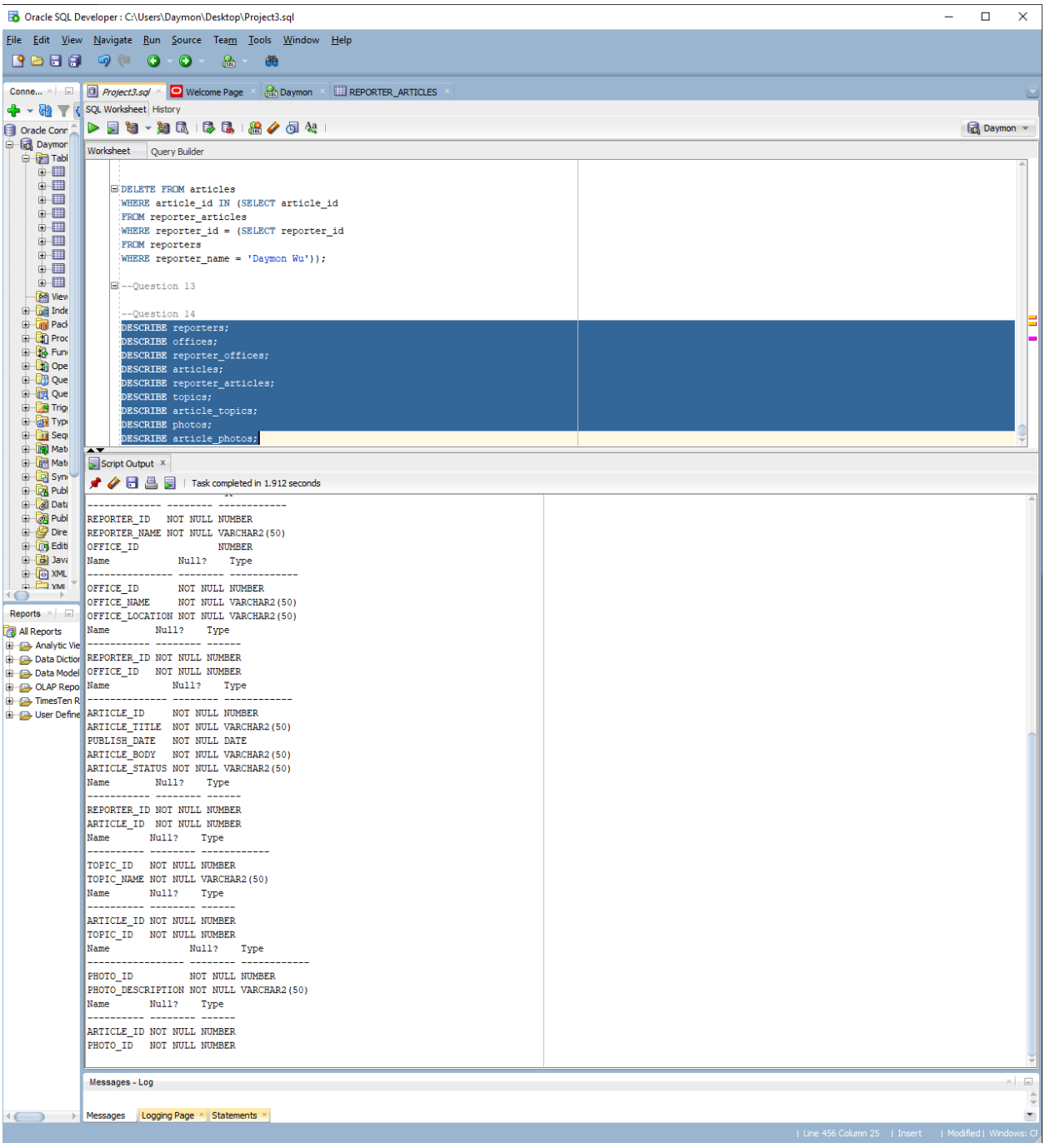


There are two instances that are running at the same time. They will not effect each other unless the changes have been committed and have been refreshed.

13. In one SQL window, change the password for the reporter Bo Li. Don't commit. In another SQL window, change the last name of reporter Bo Li. Don't commit. Quit both Oracle sessions. Login to Oracle again and display all columns for the customer Bo Li . Explain your results. Disable the auto commit flag at the top of the windows before performing this operation. Show all SQL to perform these operations.

14. Display the structure of ALL tables using SQL Describe.

```
DESCRIBE reporters;

DESCRIBE offices;

DESCRIBE reporter_offices;

DESCRIBE articles;

DESCRIBE reporter_articles;

DESCRIBE topics;

DESCRIBE article_topics;

DESCRIBE photos;

DESCRIBE article_photos;
```


Oracle SQL Developer : C:\Users\Daymon\Desktop\Project3.sql
File Edit View Navigate Run Source Team Tools Window Help
Project3.sql
Welcome Page
Daymon
REPORTER_ARTICLES
SQL Worksheet History
Worksheet Query Builder
DELETE FROM articles
WHERE article_id IN (SELECT article_id
FROM reporter_articles
WHERE reporter_id = (SELECT reporter_id
FROM reporters
WHERE reporter_name = 'Daymon Wu'));
--Question 13
--Question 14
DESCRIBE reporters;
DESCRIBE offices;
DESCRIBE reporter_offices;
DESCRIBE articles;
DESCRIBE reporter_articles;
DESCRIBE topics;
DESCRIBE article_topics;
DESCRIBE photos;
DESCRIBE article_photos;
Script Output
Task completed in 1.912 seconds
REPORTER_ID NOT NULL NUMBER
REPORTER_NAME NOT NULL VARCHAR2(50)
OFFICE_ID NUMBER
OFFICE_ID NOT NULL NUMBER
OFFICE_NAME NOT NULL VARCHAR2(50)
OFFICE_LOCATION NOT NULL VARCHAR2(50)
REPORTER_ID NOT NULL NUMBER
OFFICE_ID NOT NULL NUMBER
ARTICLE_ID NOT NULL NUMBER
ARTICLE_TITLE NOT NULL VARCHAR2(50)
PUBLISH_DATE NOT NULL DATE
ARTICLE_BODY NOT NULL VARCHAR2(50)
ARTICLE_STATUS NOT NULL VARCHAR2(50)
REPORTER_ID NOT NULL NUMBER
ARTICLE_ID NOT NULL NUMBER
TOPIC_ID NOT NULL NUMBER
TOPIC_NAME NOT NULL VARCHAR2(50)
ARTICLE_ID NOT NULL NUMBER
TOPIC_ID NOT NULL NUMBER
PHOTO_ID NOT NULL NUMBER
PHOTO_DESCRIPTION NOT NULL VARCHAR2(50)
ARTICLE_ID NOT NULL NUMBER
PHOTO_ID NOT NULL NUMBER
Messages - Log
Messages Logging Page Statements
Line 456 Column 25 Insert Modified Windows: CR

15. Display the version of Oracle. Enter:

SELECT *

FROM v$version;