

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

)

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows a connection to 'Daymon'. Below it, the 'Reports' pane lists various report types. The main window displays the 'REPORTERS' table in the 'Data' tab. The table has three columns: 'REPORTER\_ID', 'REPORTER\_NAME', and 'OFFICE\_ID'. The data is as follows:

REPORTER_ID	REPORTER_NAME	OFFICE_ID
1	100 James Wu	(null)
2	101 Mary Bow	10
3	102 Kat Mars	14
4	103 Alan Katz	14
5	104 Bob Marley	14
6	105 Daymon Wu	12
7	121 Jerry	(null)
8	122 Sam	(null)

At the bottom, the 'Messages - Log' pane shows tabs for 'Messages', 'Logging Page', and 'Statements'.

DAYMON.REPORTERS		
P *	REPORTER_ID	NUMBER
*	REPORTER_NAME	VARCHAR2 (50 BYTE)
	OFFICE_ID	NUMBER
REPORTERS_PK (REPORTER_ID)		

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

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

DAYMON.REPORTER\_OFFICES

P	REPORTER_ID	NUMBER
PF	OFFICE_ID	NUMBER

REPORTER\_OFFICES\_PK (REPORTER\_ID, OFFICE\_ID)

SYS\_C0010808 (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

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
- Data Modeler 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

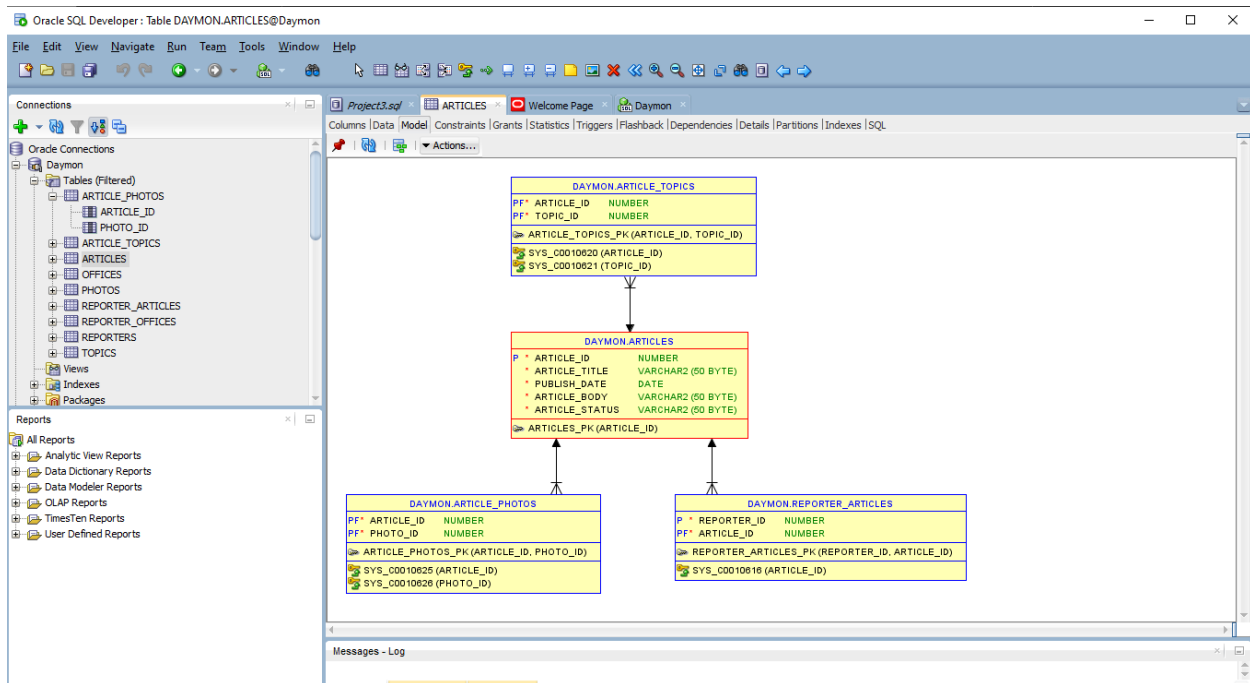
)

The screenshot shows the Oracle SQL Developer interface. The main window displays the 'ARTICLES' table data. The table has five columns: ARTICLE\_ID, ARTICLE\_TITLE, PUBLISH\_DATE, ARTICLE\_BODY, and ARTICLE\_STATUS. The data is as follows:

ARTICLE_ID	ARTICLE_TITLE	PUBLISH_DATE	ARTICLE_BODY	ARTICLE_STATUS
1	700 Cooking with Obama	15-APR-23	Wow! Obama can cook!	Published
2	701 Hello Elmo	16-APR-23	Tickle me elmo.	Published
3	702 Goodwill	17-APR-23	I want free shoes.	Published
4	705 Regular Article	20-MAR-23	This is a regular article with text content.	Published
5	706 Cars and Automobiles	17-MAY-22	The importance of the wheel	Published
6	707 Article A	17-MAY-23	AAAAA	Published
7	708 Article B	17-MAY-23	BBBBB	Published
8	709 Article C	10-JAN-23	CCCCC	Published
9	800 The Importance of Education	17-MAY-22	CUNY offers quality education.	Published
10	802 CUNY's Role in Promoting Financial Literacy	30-NOV-22	CUNY actively promotes financial literacy.	Published
11	801 Higher Education and Financial Planning	12-OCT-22	CUNY students various financial aid options.	Published
12	803 Yankees Win the Championship	18-MAY-23	The New York Yankees secure the championship title	Published
13	900 Artificial Intelligence	18-MAY-23	Uhoh	Published
14	901 Renewable Energy	18-MAY-23	Recycle	Published

The left pane shows the 'Connections' tree with 'Daymon' selected. The right pane shows the 'Messages' log with the following messages:

```
Messages
Loading Page
Statements
```



```

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

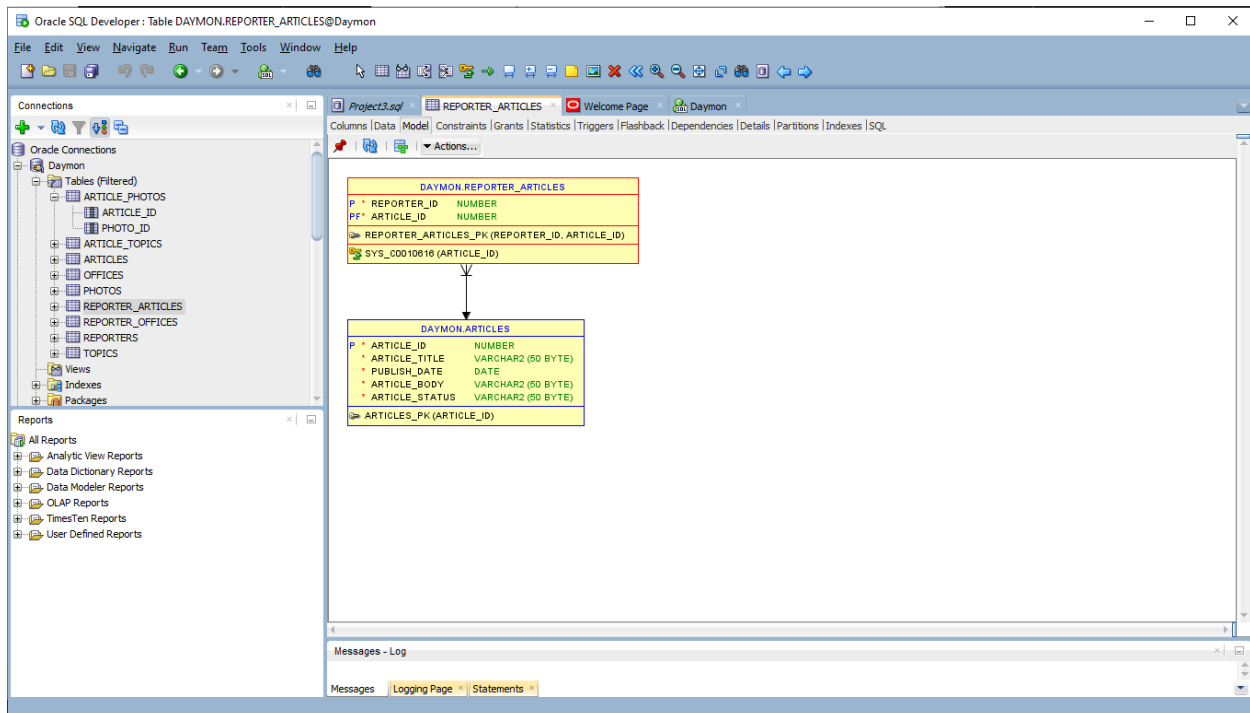
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort: | Filter:

	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



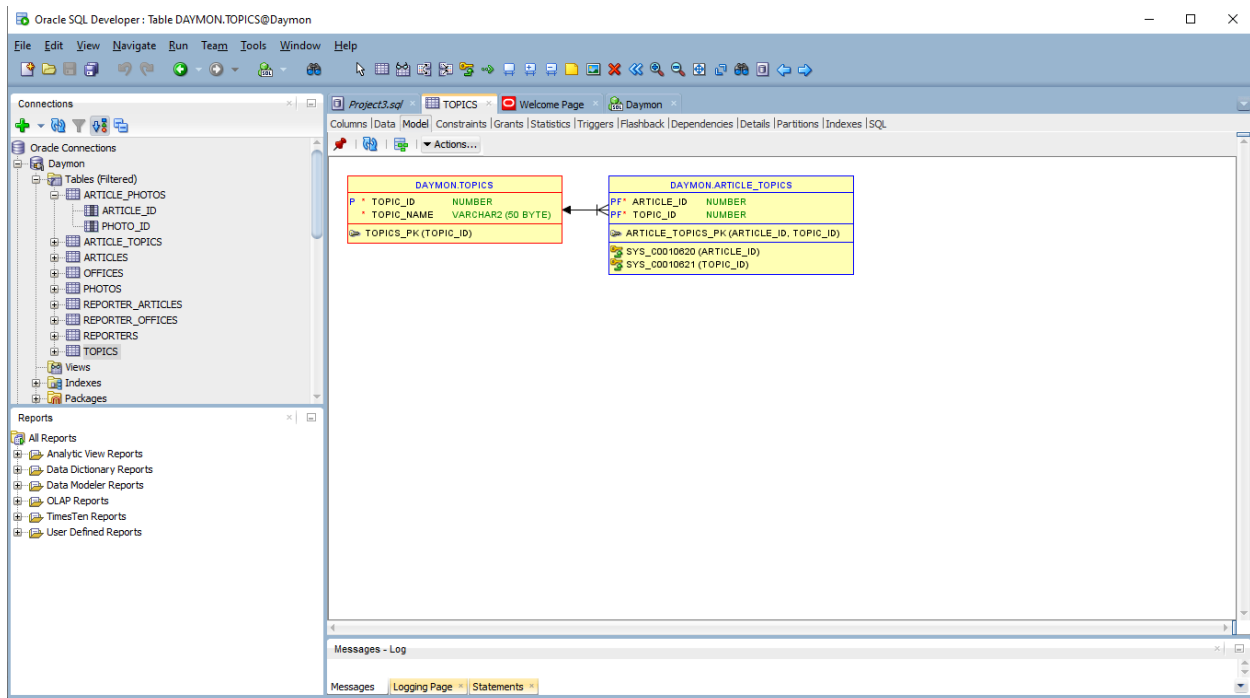
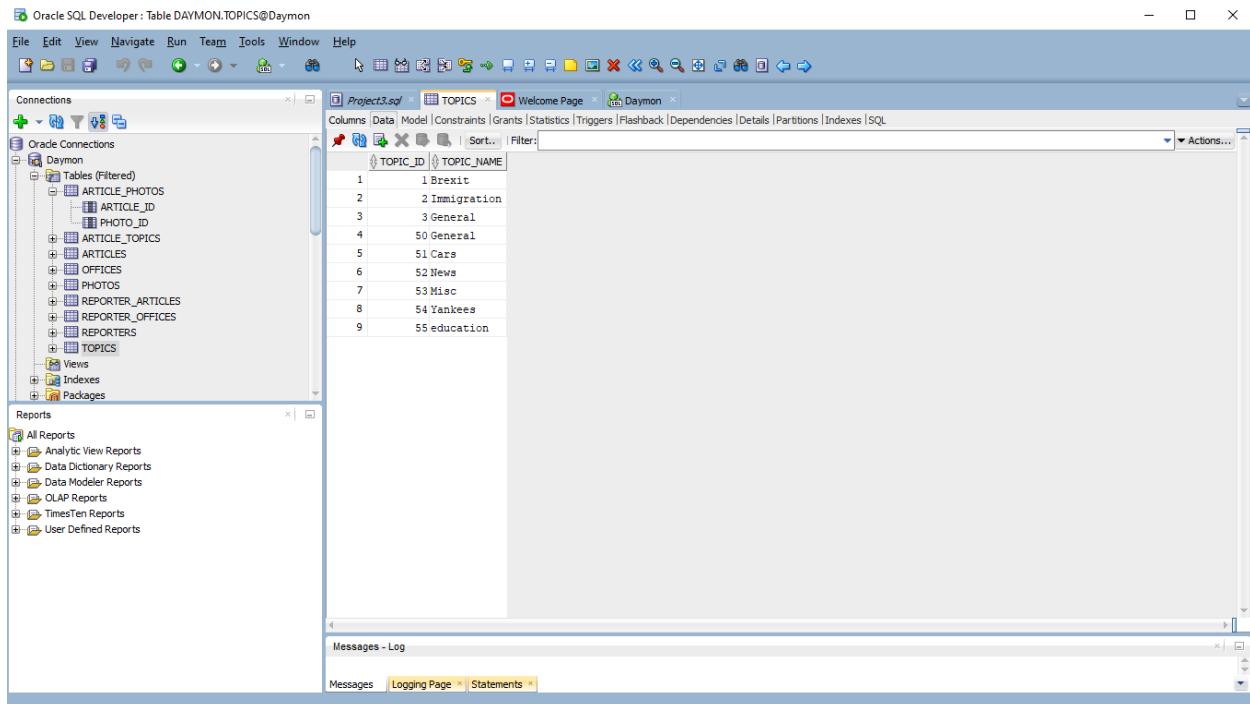


CREATE TABLE topics (

topic\_id NUMBER PRIMARY KEY,

topic\_name VARCHAR2(50) NOT NULL

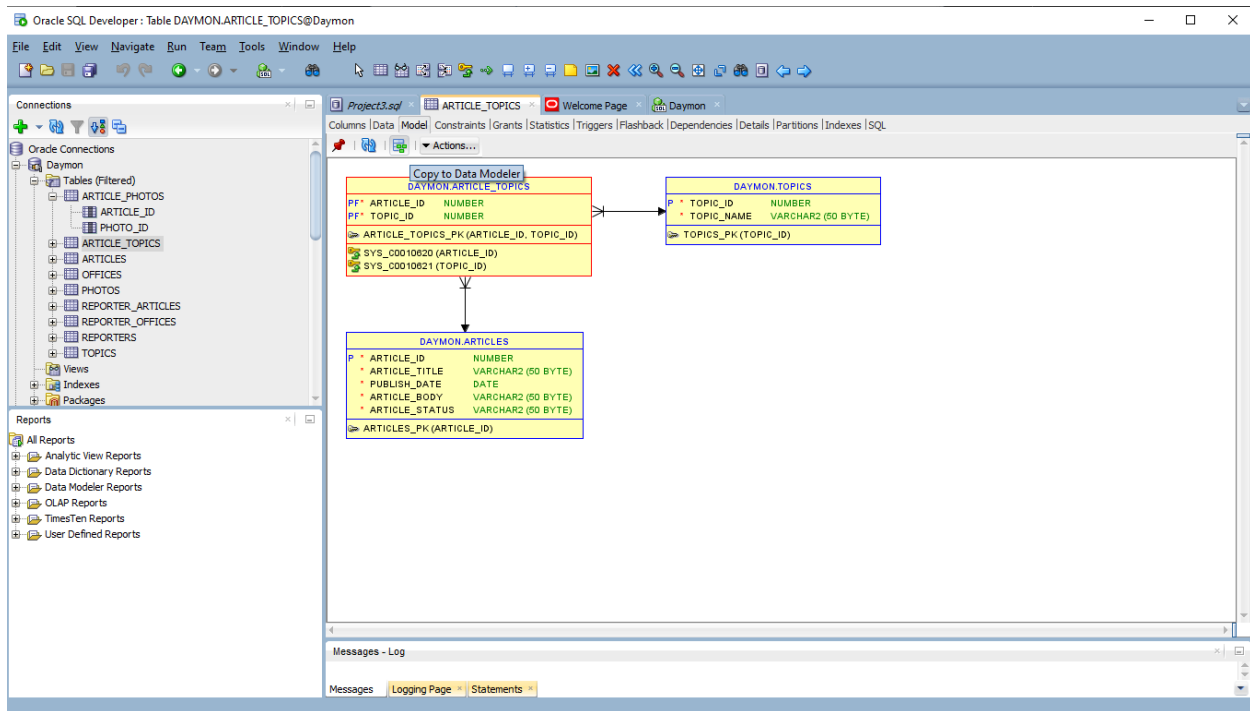
)



```
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)  
)
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' tree with 'Daymon' selected, showing a hierarchy of tables including 'ARTICLE\_TOPICS'. The main pane shows the 'Data' tab for the 'ARTICLE\_TOPICS' table, displaying 15 rows of data. The bottom pane shows the 'Messages - Log' window.

ARTICLE_ID	TOPIC_ID
1	700
2	701
3	702
4	705
5	706
6	707
7	800
8	700
9	701
10	702
11	705
12	706
13	801
14	802
15	803



CREATE TABLE photos (

photo\_id NUMBER PRIMARY KEY,

photo\_description VARCHAR2(50) NOT NULL

)

Oracle SQL Developer: Table DAYMON.PHOTOS@Daymon

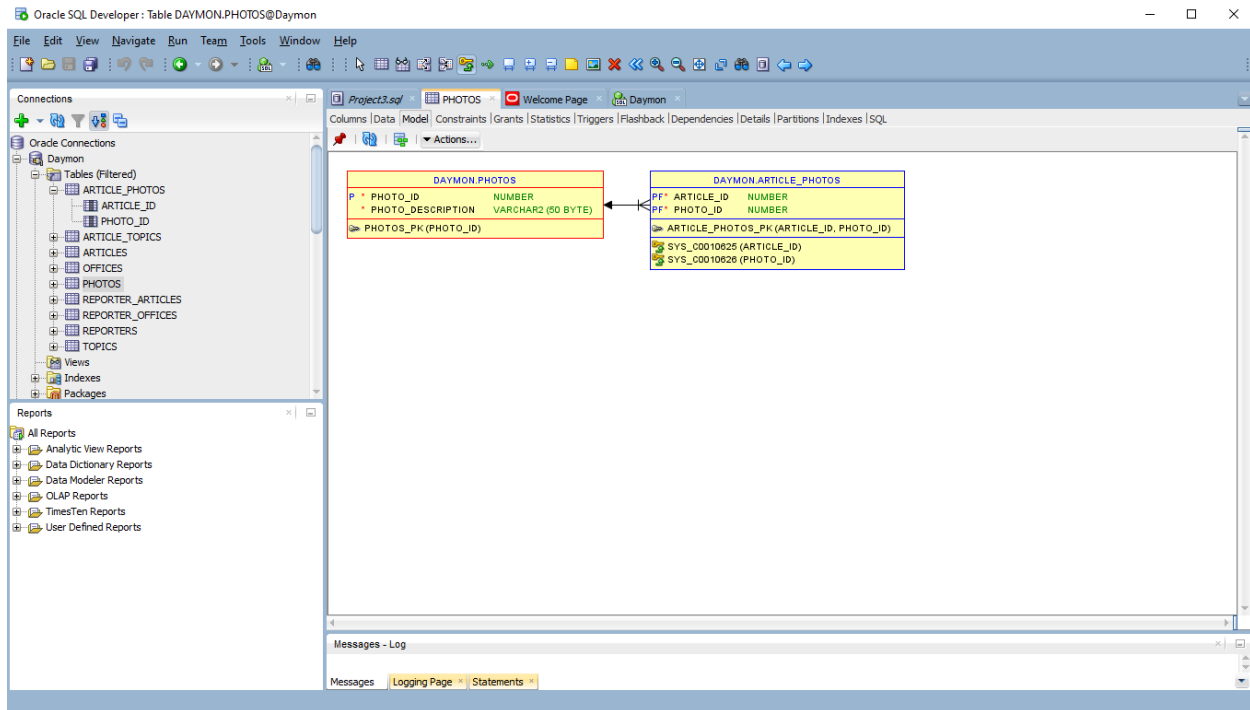
Project3.sql | PHOTOS | Welcome Page | Daymon

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

	PHOTO_ID	PHOTO_DESCRIPTION
1	5000	Brexit Photo 1
2	5001	Immigration Photo 1
3	5002	Elmo Photo 1
4	5003	Car Photo 1

Messages - Log

Messages | Logging Page | Statements



```
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

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Project3.sql

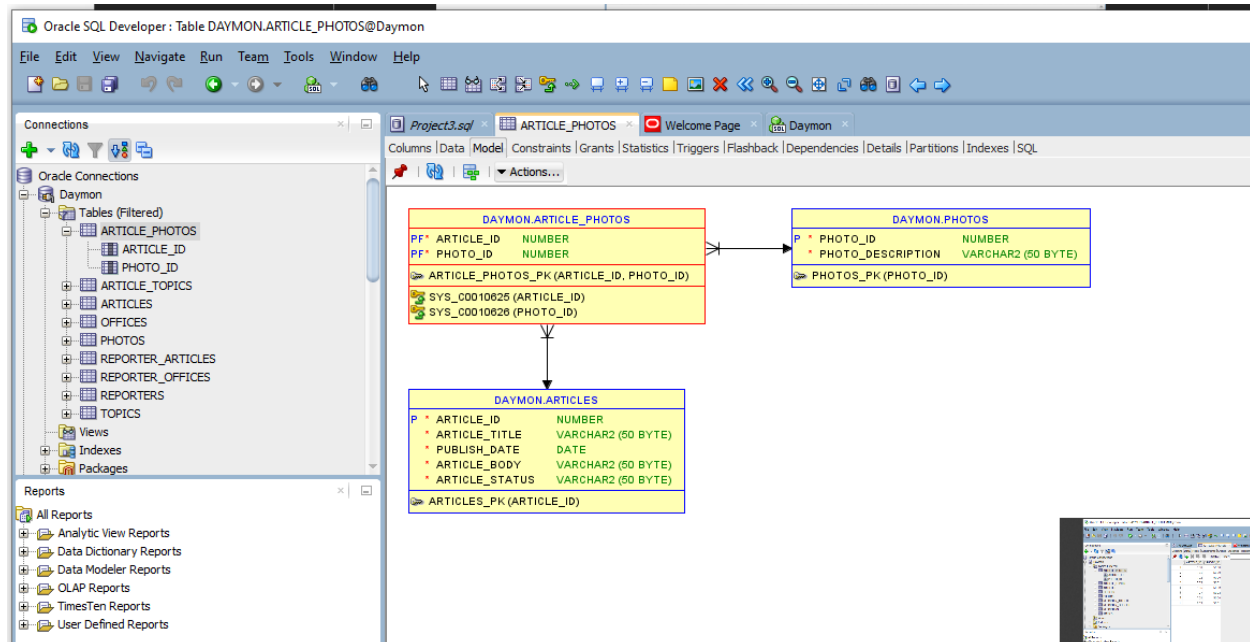
ARTICLE\_PHOTOS

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



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';
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the 'Worksheet' tab. The query is the same one as shown in the previous block. Below the query, the 'Query Result' tab shows the results of the query. The results are displayed in a table with three columns: ARTICLE\_TITLE, PUBLISH\_DATE, and REPORTER\_NAME. There are three rows of data.

	ARTICLE_TITLE	PUBLISH_DATE	REPORTER_NAME
1	Cooking with Obama	15-APR-23	James Wu
2	Hello Elmo	16-APR-23	Mary Bow
3	Goodwill	17-APR-23	Kat Mars

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%');
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like ARTICLE\_PHOTOS, ARTICLE\_TOPICS, ARTICLES, OFFICES, PHOTOS, REPORTER\_ARTICLES, REPORTER\_OFFICES, REPORTERS, TOPICS, and TOPICS. The main window shows a SQL worksheet with the following query:

```
WHERE t.topic_name IN ('sexism', 'immigration')
AND art.publish_date >= ADD_MONTHS(TO_DATE('2023-04-17', 'YYYY-MM-DD'), -6)
AND art.article_status = 'Published';

--Question 2
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%'));
```

The bottom pane shows the query results:

ARTICLE_TITLE	PUBLISH_DATE
1 CUNY's Role in Promoting Financial Literacy	30-NOV-22
2 Higher Education and Financial Planning	12-OCT-22

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'  
    )  
)
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like ARTICLE\_PHOTOS, ARTICLE\_TOPICS, ARTICLES, OFFICES, PHOTOS, REPORTER\_ARTICLES, REPORTER\_OFFICES, REPORTERS, and TOPICS. The main window shows a SQL worksheet with the following query:

```
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'  
)
```

Below the query, a comment indicates it is for "Question 4". The query is executed, and the results are shown in the "Query Result" pane:

TOPIC_ID	TOPIC_NAME
3	General
4	50 General
5	51 Cars
6	52 News
7	53 Misc
8	54 Yankees
9	55 education

The status bar at the bottom indicates "All Rows Fetched: 9 in 0.005 seconds".



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;
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like ARTICLE\_PHOTOS, ARTICLE\_TOPICS, ARTICLES, OFFICES, PHOTOS, REPORTER\_ARTICLES, REPORTER\_OFFICES, REPORTERS, and TOPICS. The main window shows a SQL worksheet with the following query:

```
--Question 4
SELECT t.topic_name, COUNT(*) AS article_count
FROM topics t
JOIN article_topics at ON t.topic_id = at.topic_id
JOIN articles a ON at.article_id = a.article_id
WHERE a.publish_date >= ADD_MONTHS(TRUNC(SYSDATE), -12)
GROUP BY t.topic_name
ORDER BY COUNT(*) DESC
FETCH FIRST 5 ROWS ONLY;
```

Below the query, the 'Query Result' pane shows the output:

TOPIC_NAME	ARTICLE_COUNT
1 General	4
2 Immigration	2
3 education	2
4 Brexit	2
5 Cars	1

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;
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like ARTICLE\_PHOTOS, ARTICLE\_TOPICS, ARTICLES, OFFICES, OFFICE\_ID, OFFICE\_NAME, OFFICE\_LOCATION, PHOTOS, REPORTER\_ARTICLES, REPORTER\_OFFICES, REPORTER\_ID, REPORTER\_NAME, TOPICS, TOPIC\_ID, and TOPIC\_NAME. The main pane shows a SQL worksheet with the following query:

```
--Question 5
SELECT r.reporter_name, TO_CHAR(a.publish_date, 'YYYY') AS year_published, COUNT(*) AS article_count
FROM reporters r
JOIN reporter_articles ra ON r.reporter_id = ra.reporter_id
JOIN articles a ON ra.article_id = a.article_id
WHERE a.publish_date >= ADD_MONTHS(TRUNC(SYSDATE), -60)
GROUP BY r.reporter_name, TO_CHAR(a.publish_date, 'YYYY')
ORDER BY r.reporter_name;
```

The bottom pane shows the query results in a table with 3 rows and 3 columns: REPORTER\_NAME, YEAR\_PUBLISHED, and ARTICLE\_COUNT.

REPORTER_NAME	YEAR_PUBLISHED	ARTICLE_COUNT
1 James Wu	2023	1
2 Kat Mars	2023	1
3 Mary Bow	2023	1

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;
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like ARTICLE\_PHOTOS, ARTICLE\_TOPICS, ARTICLES, OFFICES, PHOTOS, REPORTER\_ARTICLES, REPORTER\_OFFICES, REPORTERS, and TOPICS. The main window shows a SQL worksheet with two queries. Query 5 is a nested select query that filters for articles published in 2023 by reporters James Wu, Kat Mars, and Mary Bow. Query 6 is the query from the problem statement. The bottom pane shows the query results for Query 5.

```
--Question 5  
SELECT r.reporter_name, TO_CHAR(a.publish_date, 'YYYY') AS year_published, COUNT(*) AS article_count  
FROM reporters r  
JOIN reporter_articles ra ON r.reporter_id = ra.reporter_id  
JOIN articles a ON ra.article_id = a.article_id  
WHERE a.publish_date >= ADD_MONTHS(TRUNC(SYSDATE), -60)  
GROUP BY r.reporter_name, TO_CHAR(a.publish_date, 'YYYY')  
ORDER BY r.reporter_name;
```

```
--Question 6  
SELECT ar.article_title, ar.publish_date
```

REPORTER_NAME	YEAR_PUBLISHED	ARTICLE_COUNT
1 James Wu	2023	1
2 Kat Mars	2023	1
3 Mary Bow	2023	1

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;
```

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL worksheet with the following query:

```
--Question 7  
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;  
--Question 8
```

The query has been executed, and the results are displayed in the 'Query Result' pane. The results show 6 rows of data, all with the same reporter name, 'Mary Bow'.

REPORTER_NAME
1 Daymon Wu
2 James Wu
3 Jerry
4 Kat Mars
5 Mary Bow
6 Mary Bow

The 'Messages - Log' pane at the bottom shows the following message:

```
Messages | Logging Page | Statements  
Click on an identifier with the Control key down to perform "Go to Declaration"
```

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;
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like ARTICLES, REPORTER\_ARTICLES, and TOPICS. The main window shows a SQL worksheet with the following query:

```
--Question 8
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;

INSERT INTO reporter_articles (reporter_id, article_id)
VALUES (100, 800);
```

The bottom pane shows the query results in a table with columns ARTICLE\_TITLE and PUBLISH\_DATE. The result is:

ARTICLE_TITLE	PUBLISH_DATE
1 Yankees Win the Championship	18-MAY-23

The status bar at the bottom indicates "All Rows Fetched: 1 in 0.009 seconds".

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
);
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like ARTICLE\_PHOTOS, ARTICLE\_TOPICS, ARTICLES, OFFICES, PHOTOS, REPORTER\_ARTICLES, REPORTERS, and TOPICS. The main window shows a SQL worksheet with the following query:

```
--Question 9
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
);

INSERT INTO articles (article_id, article_title, publish_date, article_body, article_status)
VALUES (900, 'Artificial Intelligence', DATE '2023-05-18', 'Uhhoh', 'Published');
INSERT INTO articles (article_id, article_title, publish_date, article_body, article_status)
VALUES (901, 'Renewable Energy', DATE '2023-05-18', 'Recycle', 'Published');
```

Below the query, the 'Query Result' tab shows the output:

ARTICLE_TITLE	REPORTER_NAME
1 Artificial Intelligence	Jerry
2 Renewable Energy	Sam

The status bar at the bottom indicates 'Line: 362 Column: 3 | Insert | Modified | Windows: C'.

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'

)

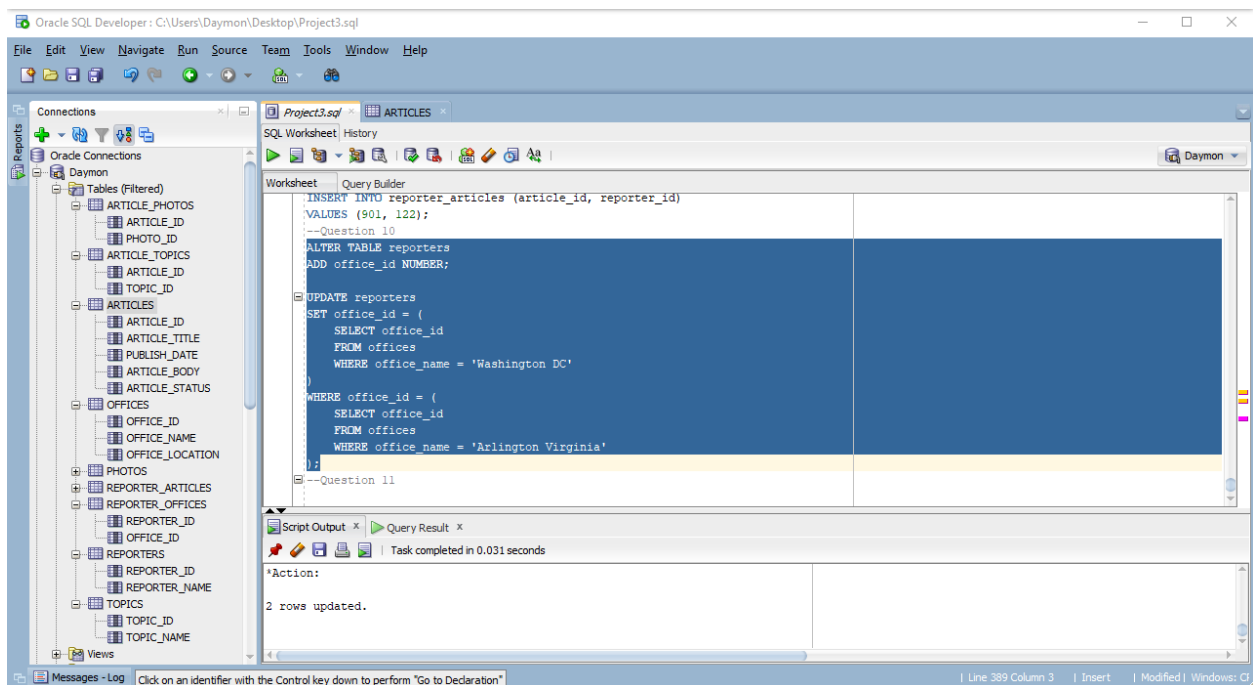
WHERE office\_id = (

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 image displays two side-by-side screenshots of the Oracle SQL Developer interface, demonstrating the execution of SQL statements and the resulting data.

**Left Screenshot:** The SQL Worksheet shows the following script:

```
SET AUTOCOMMIT OFF;
DELETE FROM topics;
SELECT COUNT(*) AS topic_count FROM topics;
INSERT INTO topics (topic_id, topic_name)
VALUES (1, 'Topic 1');
```

The Query Result window shows the output of the `SELECT COUNT(*)` query:

TOPIC_COUNT
1

The Messages - Log window shows the message: "15 rows deleted."

**Right Screenshot:** The SQL Worksheet shows the following script:

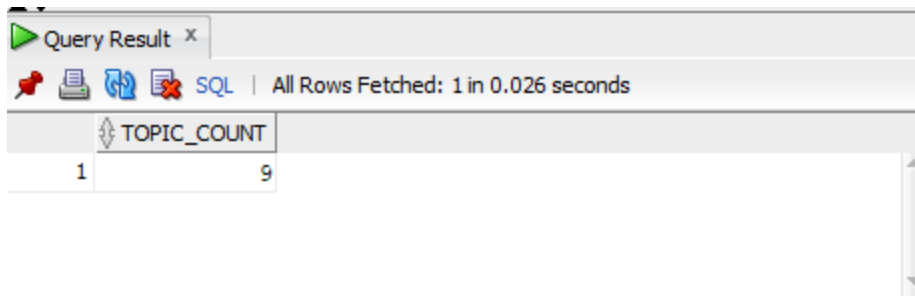
```
WHERE office_id = (
SELECT office_id
FROM offices
WHERE office_name = 'Arlington Virginia'
);
--Question 11
SET AUTOCOMMIT OFF;
DELETE FROM article_topics;
DELETE FROM topics;
SELECT COUNT(*) AS topic_count FROM topics;
INSERT INTO topics (topic_id, topic_name)
VALUES (1, 'Topic 1'), (2, 'Topic 2'), (3,
--Question 12
SELECT COUNT(*) AS topic_count FROM topics;
--Question 13
```

The Query Result window shows the output of the `SELECT COUNT(*)` query:

TOPIC_COUNT
1

The Messages - Log window shows the message: "Task completed in 0.006 seconds."

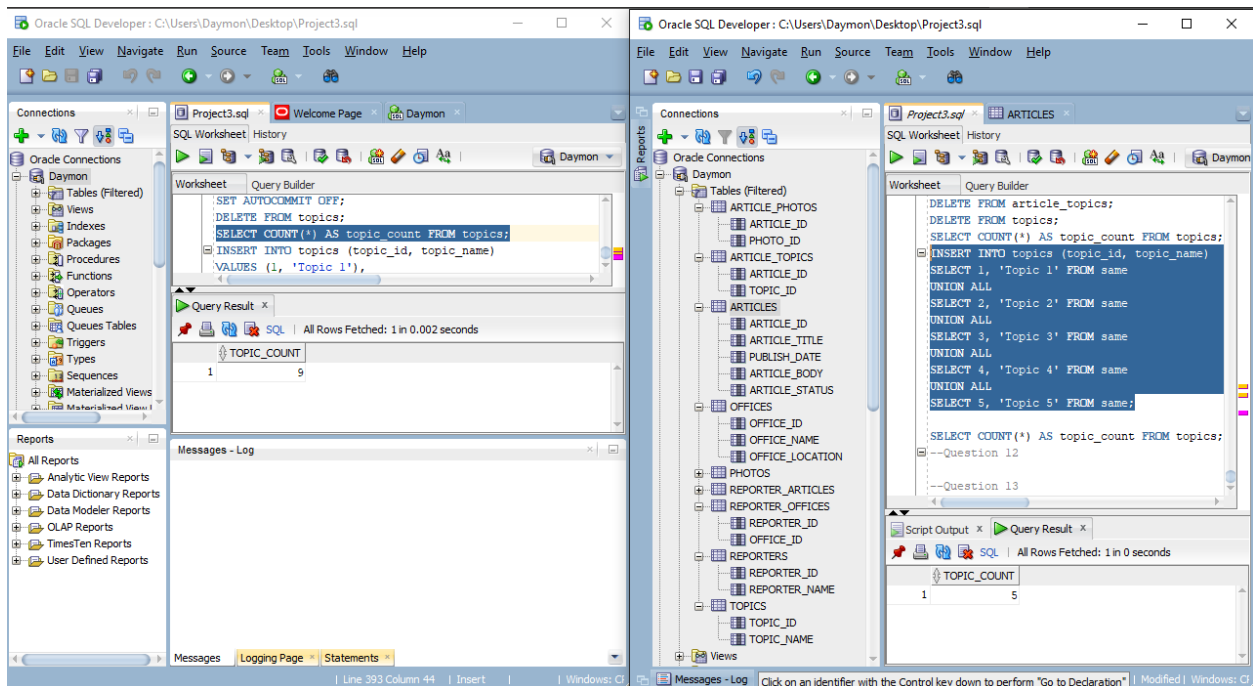




Query Result x

SQL | All Rows Fetched: 1 in 0.026 seconds

TOPIC_COUNT
9



Oracle SQL Developer: C:\Users\Daymon\Desktop\Project3.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections: Daymon

Tables (Filtered): TABLES, VIEWS, INDEXES, PACKAGES, PROCEDURES, FUNCTIONS, OPERATORS, QUEUES, TRIGGERS, SEQUENCES, TYPES, MATERIALIZED VIEWS

Worksheet: Project3.sql

```
SET AUTOCOMMIT OFF;
DELETE FROM topics;
SELECT COUNT(*) AS topic_count FROM topics;
INSERT INTO topics (topic_id, topic_name)
VALUES (1, 'Topic 1');
```

Query Result x

SQL | All Rows Fetched: 1 in 0.002 seconds

TOPIC_COUNT
9

Reports: All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports

Messages - Log

Oracle SQL Developer: C:\Users\Daymon\Desktop\Project3.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections: Daymon

Tables (Filtered): ARTICLE\_PHOTOS, ARTICLE\_ID, PHOTO\_ID, ARTICLE\_TOPICS, ARTICLE\_ID, TOPIC\_ID, ARTICLES, ARTICLE\_ID, ARTICLE\_TITLE, PUBLISH\_DATE, ARTICLE\_BODY, ARTICLE\_STATUS, OFFICES, OFFICE\_ID, OFFICE\_NAME, OFFICE\_LOCATION, PHOTOS, REPORTER\_ARTICLES, REPORTER\_OFFICES, REPORTER\_ID, OFFICE\_ID, REPORTERS, REPORTER\_ID, REPORTER\_NAME, TOPICS, TOPIC\_ID, TOPIC\_NAME

Worksheet: Project3.sql

```
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;
```

Script Output x

Query Result x

SQL | All Rows Fetched: 1 in 0 seconds

TOPIC_COUNT
5

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 of your 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');

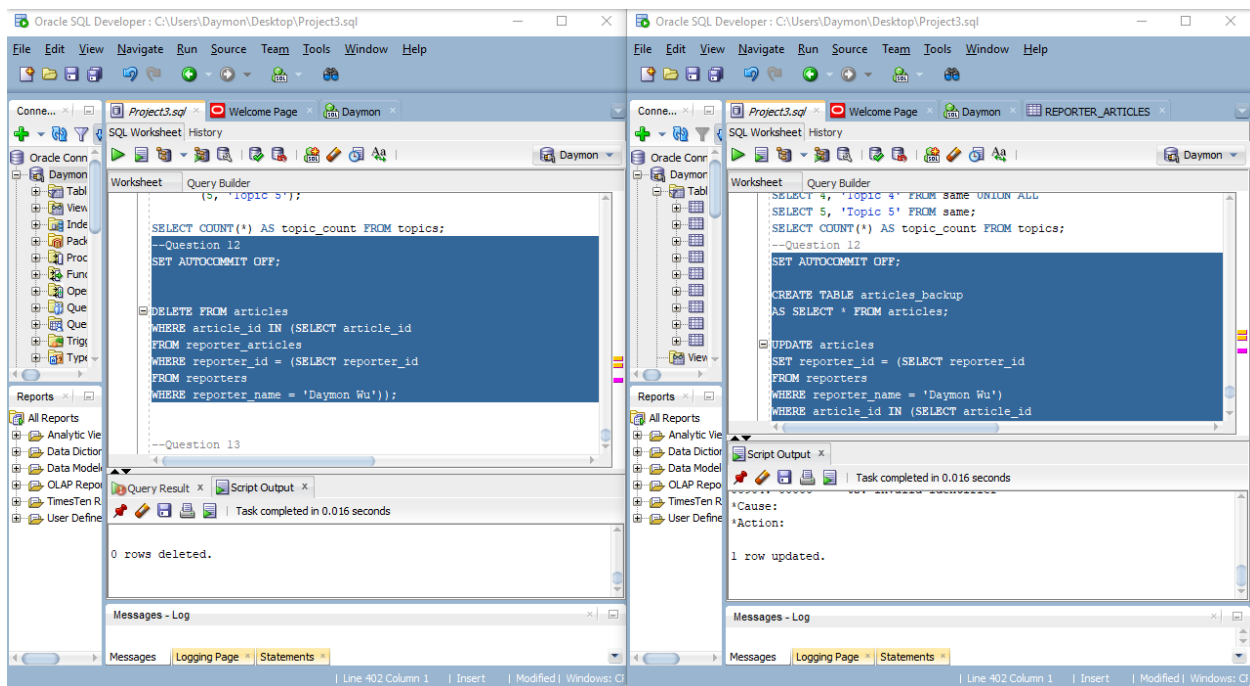
DELETE FROM articles;

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;

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script with the following content:

```
--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;
```

The Script Output window shows the results of the DESCRIBE statements:

```
Task completed in 1.912 seconds

REPORTER_ID  NOT NULL NUMBER
REPORTER_NAME NOT NULL VARCHAR2(50)
OFFICE_ID    NUMBER
Name        Null?   Type
-----
OFFICE_ID    NOT NULL NUMBER
OFFICE_NAME  NOT NULL VARCHAR2(50)
OFFICE_LOCATION NOT NULL VARCHAR2(50)
Name        Null?   Type
-----
REPORTER_ID  NOT NULL NUMBER
OFFICE_ID    NOT NULL NUMBER
Name        Null?   Type
-----
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)
Name        Null?   Type
-----
REPORTER_ID  NOT NULL NUMBER
ARTICLE_ID   NOT NULL NUMBER
Name        Null?   Type
-----
TOPIC_ID     NOT NULL NUMBER
TOPIC_NAME   NOT NULL VARCHAR2(50)
Name        Null?   Type
-----
ARTICLE_ID   NOT NULL NUMBER
TOPIC_ID     NOT NULL NUMBER
Name        Null?   Type
-----
PHOTO_ID     NOT NULL NUMBER
PHOTO_DESCRIPTION NOT NULL VARCHAR2(50)
Name        Null?   Type
-----
ARTICLE_ID   NOT NULL NUMBER
PHOTO_ID     NOT NULL NUMBER
```

The interface also includes a left-hand pane with a tree view of database objects and a bottom status bar showing the current line and column.

15. Display the version of Oracle. Enter:

SELECT \*

FROM v\$version;

