

Oracle 12c: SQL

Chapter 13 *Views*

Objectives

- Create a view by using CREATE VIEW command or the CREATE OR REPLACE VIEW command
- Employ the FORCE and NOFORCE options
- State the purpose of the WITH CHECK OPTION constraint
- Explain the effect of the WITH READ ONLY option
- Update a record in a simple view
- Re-create a view

Objectives (continued)

- Explain the implication of an expression in a view for DML operations
- Update a record in a complex view
- Identify problems associated with adding records to a complex view
- Identify the key-preserved table underlying a complex view
- Drop a view
- Explain inline views and the use of ROWNUM to perform a “TOP-N” analysis
- Create a materialized view to replicate data

Views

- Permanent objects that store no data
- Store a query
- Two purposes
 - Reduce complex query requirements
 - Restrict users' access to sensitive data

Types of Views

VIEW TYPE	DESCRIPTION
Simple view	A view based upon a subquery that only references one table and does not include any group functions, expressions, or a GROUP BY clause
Complex view	A view based upon a subquery that retrieves or derives data from one or more tables—and may also contain functions or grouped data
Inline view	A subquery used in the FROM clause of a SELECT statement to create a “temporary” table that can be referenced by the SELECT and WHERE clauses of the outer statement
Materialized view	A view that replicates data by physically storing the results of the view query

Creating a View

- You use the CREATE VIEW keywords to create a view
- Use OR REPLACE if the view already exists
- Use FORCE if the underlying table does not exist at the time of creation
- Provide new column names if necessary

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW  
    viewname (columnname, ...)  
AS SELECT statement  
[WITH CHECK OPTION [CONSTRAINT constraintname]]  
[WITH READ ONLY];
```

Creating a View (continued)

- WITH CHECK OPTION constraint – if used, prevents data changes that will make the data subsequently inaccessible to the view
- WITH READ ONLY – prevents DML operations

Creating a Simple View

- Only references one table – no group functions, GROUP BY clause, or expressions

```
CREATE VIEW inventory  
  AS SELECT isbn, title, retail price  
      FROM books  
  WITH READ ONLY;
```


DML Operations on a Simple View

- Any DML operations are allowed through simple views unless created with WITH READ ONLY option
- DML operations that violate constraints on the underlying table are not allowed

Creating a Complex View

- A complex view may contain data from multiple tables or data created with the GROUP BY clause, functions, or expressions
- Type of DML operations allowed depends on various factors

```
CREATE VIEW prices  
AS SELECT isbn, title, cost, retail, retail-cost profit  
FROM books;
```

DML Operations on a Complex View with an Arithmetic Expression

Enter SQL Statement:

```
CREATE VIEW prices
AS SELECT isbn, title, cost, retail, retail-cost profit
FROM books;

SELECT *
FROM prices
WHERE title LIKE '%MICKEY%';

UPDATE prices
SET retail = 29.95
WHERE title LIKE '%MICKEY%';

SELECT *
FROM prices
WHERE title LIKE '%MICKEY%';
```

Results Script Output Explain Autotrace DBMS Output OWA Output

CREATE VIEW succeeded.

ISBN	TITLE	COST	RETAIL	PROFIT
0401140733	REVENGE OF MICKEY	14.2	22	7.8

1 rows selected

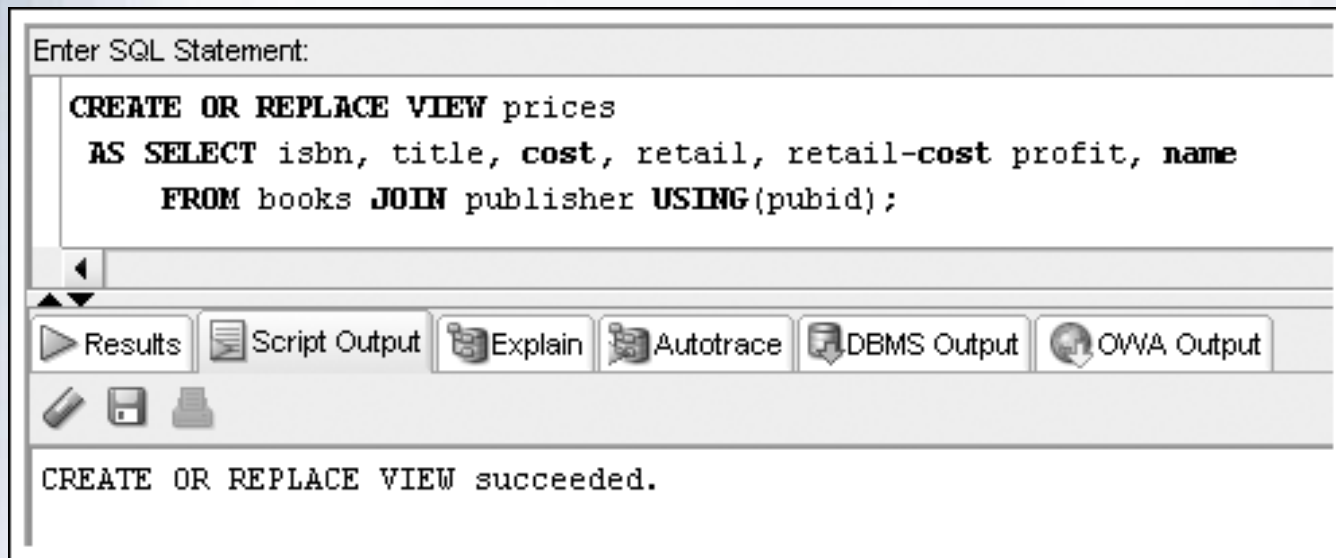
1 rows updated

ISBN	TITLE	COST	RETAIL	PROFIT
0401140733	REVENGE OF MICKEY	14.2	29.95	15.75

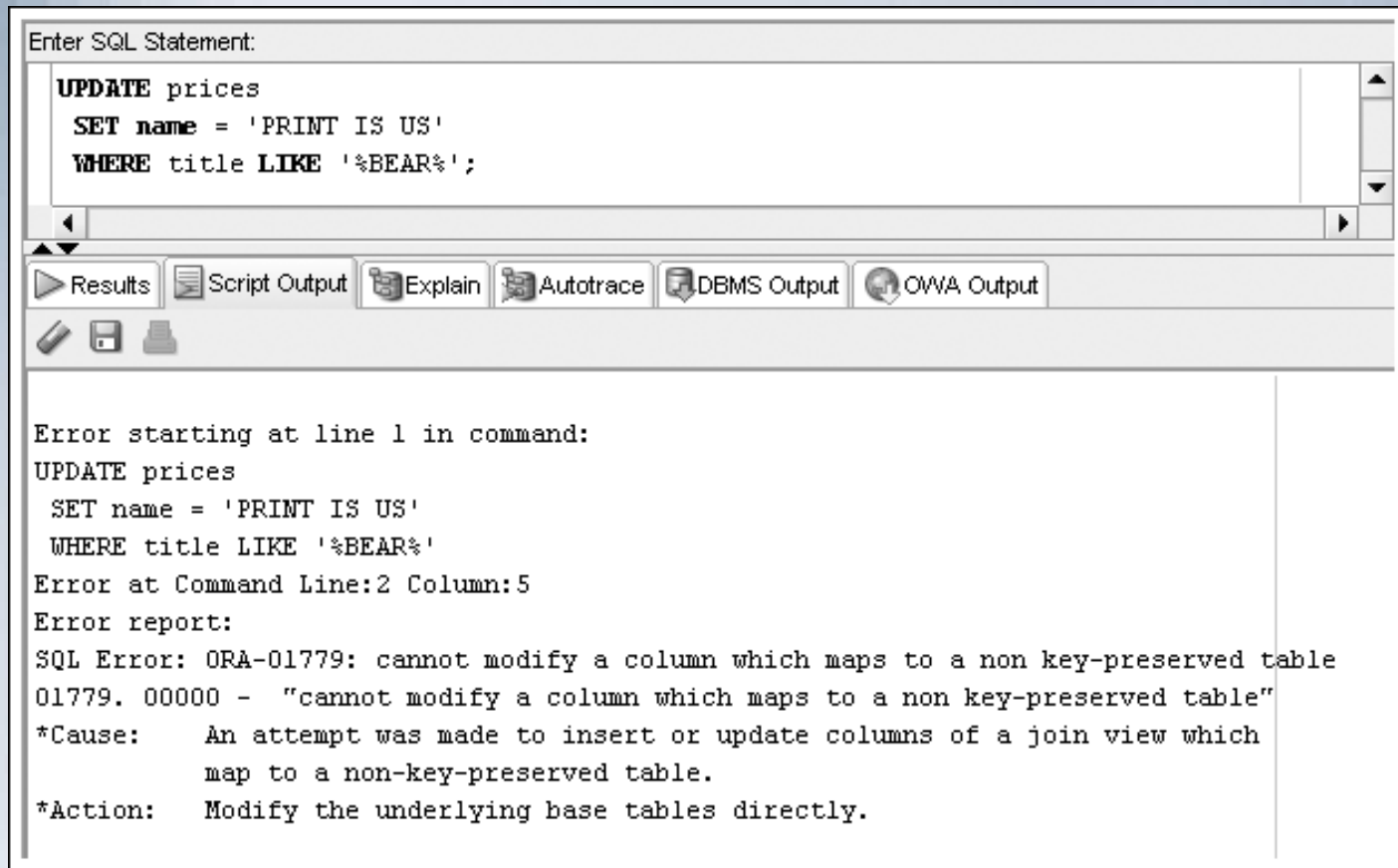
1 rows selected

DML Operations on a Complex View Containing Data from Multiple Tables

- DML operations cannot be performed on non-key-preserved tables, but they are permitted on key-preserved tables



DML Operations on a Complex View Containing Data from Multiple Tables (continued)



The screenshot shows a SQL*Plus window with the following components:

- Enter SQL Statement:** A text area containing the SQL command:

```
UPDATE prices
SET name = 'PRINT IS US'
WHERE title LIKE '%BEAR%';
```
- Toolbar:** Buttons for Results, Script Output, Explain, Autotrace, DBMS Output, and OWA Output.
- Error Message:** The main window displays the following error:

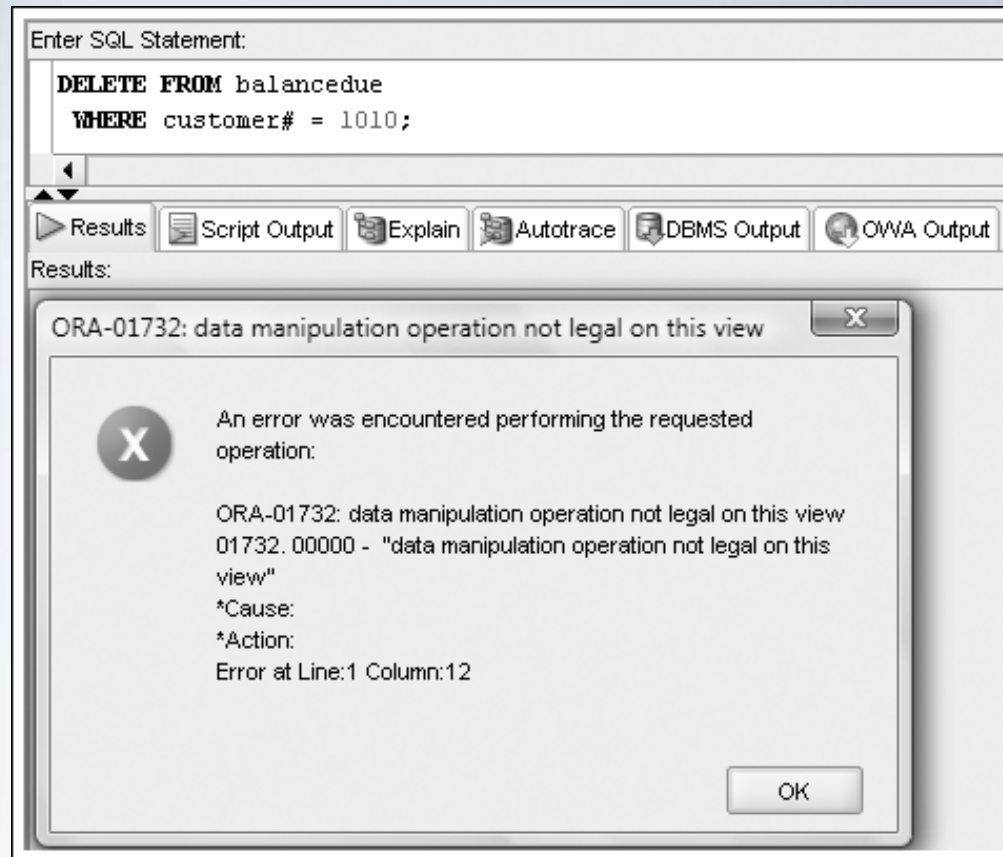
```
Error starting at line 1 in command:
UPDATE prices
  SET name = 'PRINT IS US'
  WHERE title LIKE '%BEAR%'
Error at Command Line:2 Column:5
Error report:
SQL Error: ORA-01779: cannot modify a column which maps to a non key-preserved table
ORA-01779. 00000 - "cannot modify a column which maps to a non key-preserved table"
*Cause:      An attempt was made to insert or update columns of a join view which
              map to a non-key-preserved table.
*Action:     Modify the underlying base tables directly.
```

DML Operations on a Complex View Containing Functions or Grouped Data

- DML operations are not permitted if the view includes a group function or a GROUP BY clause

```
CREATE VIEW balancedue
AS SELECT customer#, order#, SUM(quantity*retail) Amtdue
   FROM customers JOIN orders USING(customer#)
   JOIN orderitems USING(order#)
   JOIN books USING(isbn)
GROUP BY customer#, order#;
```

DML Operations on a Complex View Containing Functions or Grouped Data (continued)

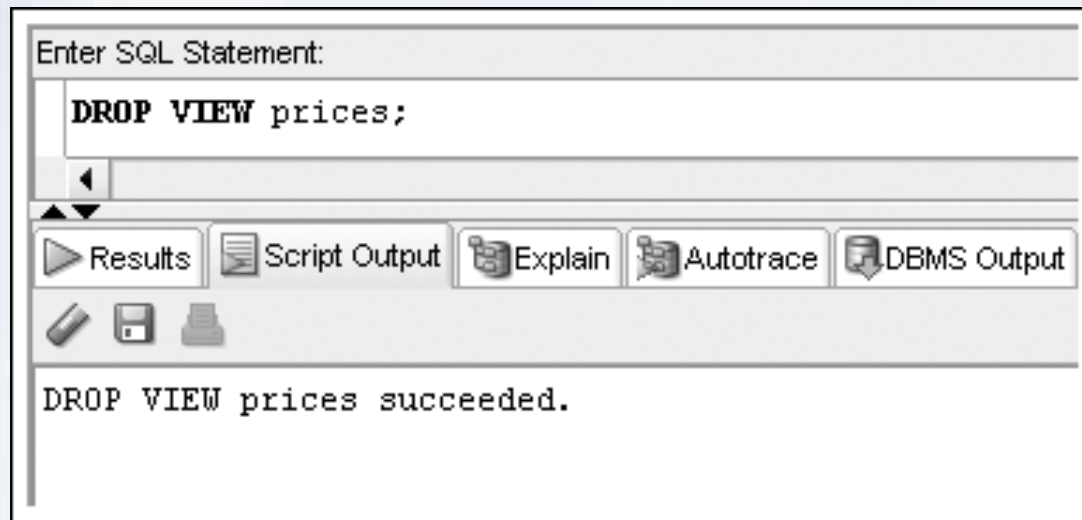


DML Operations on a Complex View Containing DISTINCT or ROWNUM

- DML operations on a view that contains the DISTINCT keyword or ROWNUM are not permitted

Dropping a View

- Use DROP VIEW command



Creating an Inline View

- An inline view is a temporary table created by using a subquery in the FROM clause
- It can only be referenced while the command is being executed
 - Most common usage – “TOP-N” analysis

“TOP-N” Analysis

- ORDER BY included to identify top values:
 - Descending for highest values
 - Ascending for lowest values
- Extract data based on ROWNUM

“TOP-N” Analysis (continued)

Enter SQL Statement:

```
SELECT title, profit
FROM (SELECT title, retail-cost profit
      FROM books
      ORDER BY retail-cost DESC)
WHERE ROWNUM <= 5;
```

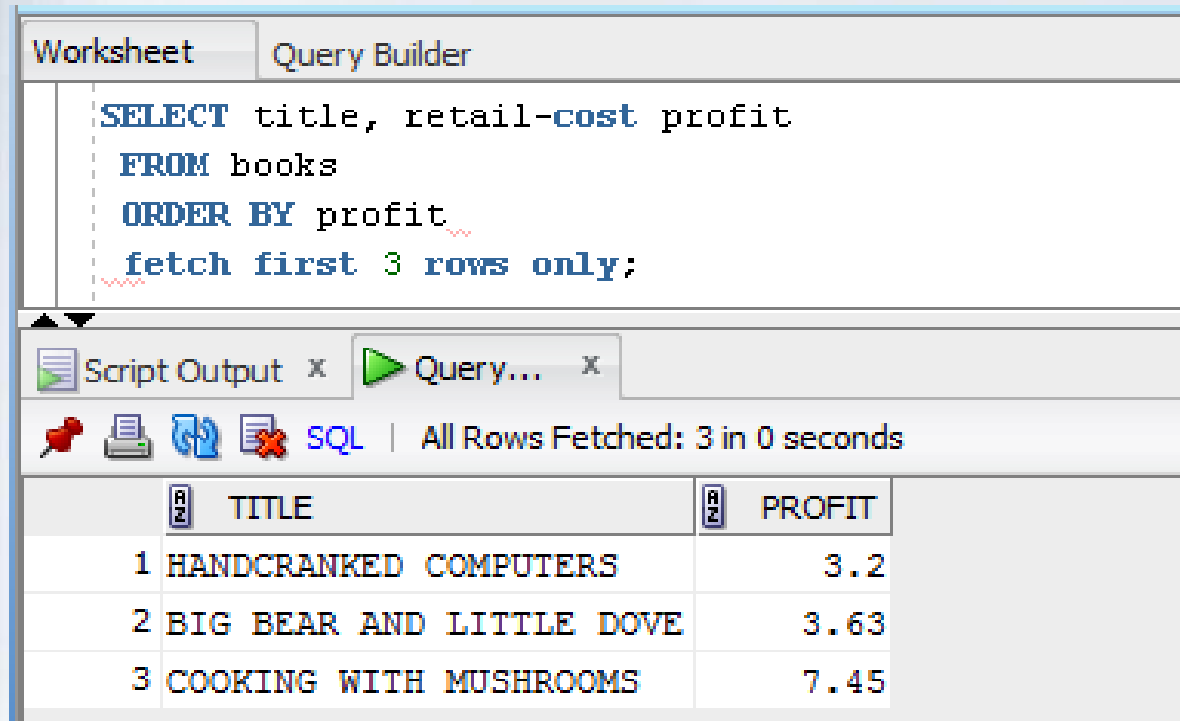
Results Script Output Explain Autotrace DBMS Output

Results:

	TITLE	PROFIT
1	PAINLESS CHILD-REARING	41.95
2	BODYBUILD IN 10 MINUTES A DAY	31.2
3	HOLY GRAIL OF ORACLE	28.7
4	DATABASE IMPLEMENTATION	24.55
5	BUILDING A CAR WITH TOOTHPICKS	22.15

“TOP-N” Analysis (continued)

- Oracle 12c introduces a new row limiting clause (# rows)



The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT title, retail-cost profit
FROM books
ORDER BY profit
fetch first 3 rows only;
```

Below the query editor, the 'Script Output' and 'Query...' tabs are visible. The status bar indicates 'All Rows Fetched: 3 in 0 seconds'. The results are displayed in a table with two columns: 'TITLE' and 'PROFIT'.

	TITLE	PROFIT
1	HANDCRANKED COMPUTERS	3.2
2	BIG BEAR AND LITTLE DOVE	3.63
3	COOKING WITH MUSHROOMS	7.45

“TOP-N” Analysis (continued)

- Oracle 12c introduces a new row limiting clause (percent of rows)

The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT title, retail-cost profit
FROM books
ORDER BY profit
fetch first 25 percent rows only;
```

Below the query editor, the 'Query Result' tab is active, showing the results of the query. The status bar indicates 'All Rows Fetched: 4 in 0.046 seconds'. The results are displayed in a table with two columns: 'TITLE' and 'PROFIT'.

	TITLE	PROFIT
1	HANDCRANKED COMPUTERS	3.2
2	BIG BEAR AND LITTLE DOVE	3.63
3	COOKING WITH MUSHROOMS	7.45
4	REVENGE OF MICKEY	7.8


Cross & Outer Apply Joins





- A column of the joining table may be used to produce the result set of the inline view

Worksheet

Query Builder

```
SELECT b.isbn, b.title, sales
FROM books b CROSS APPLY (SELECT SUM(quantity*paideach) Sales
FROM orderitems o
WHERE b.isbn = o.isbn
GROUP BY b.isbn)
ORDER BY b.title;
```

 Query Result x

    SQL | All Rows Fetched: 11 in 0.047 seconds

	ISBN	TITLE	SALES
1	8117949391	BIG BEAR AND LITTLE DOVE	35.8
2	1059831198	BODYBUILD IN 10 MINUTES A DAY	30.95
3	3437212490	COOKING WITH MUSHROOMS	159.6
4	0040170110	DATABASE IMPLEMENTATION	301.65

Materialized Views

- Replicate data
- Store data retrieved from view query
- Referred to as “snapshots”

Materialized Views (continued)

Enter SQL Statement:

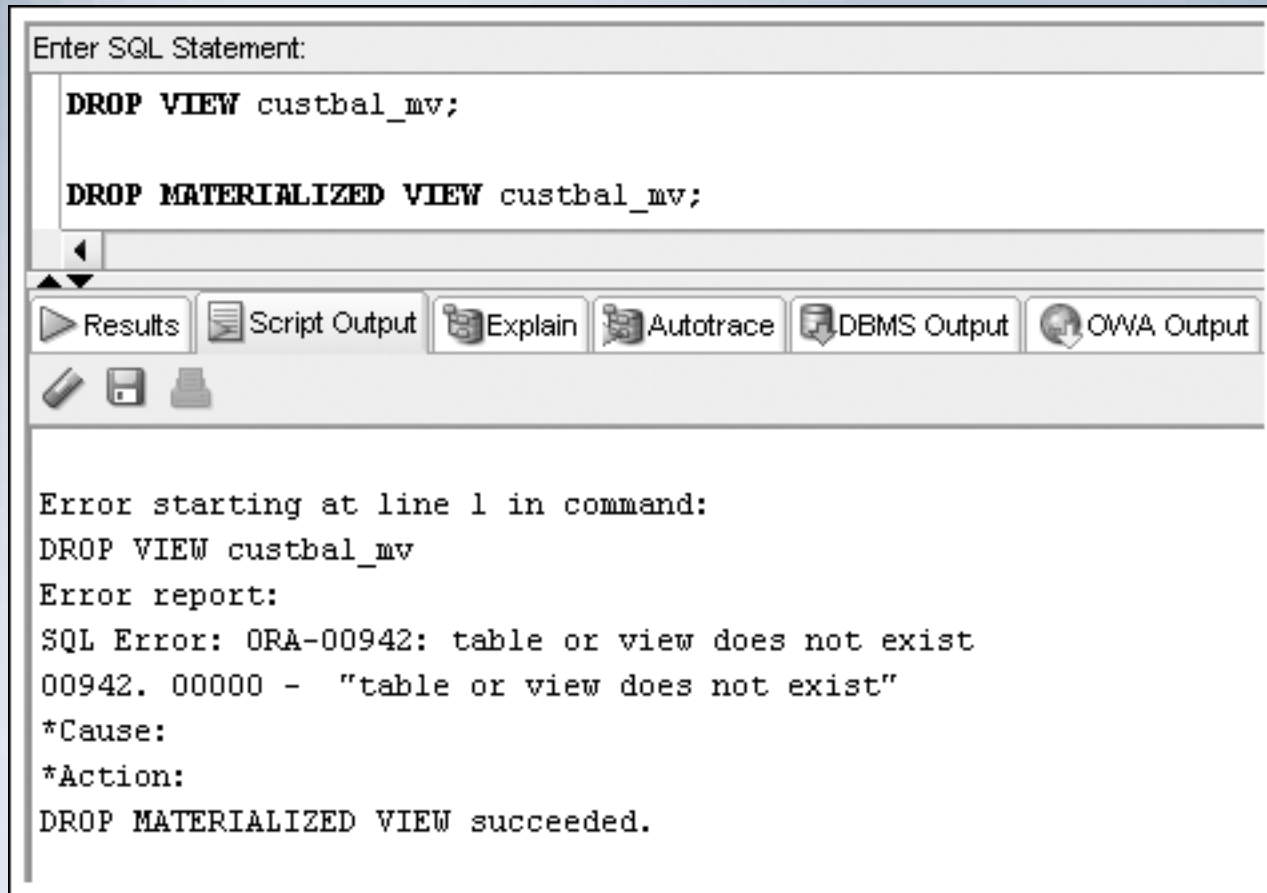
```
CREATE MATERIALIZED VIEW custbal_mv  
REFRESH COMPLETE  
START WITH SYSDATE NEXT SYSDATE+7  
AS SELECT customer#, city, state, order#, SUM(quantity*retail) Amtdue  
FROM customers JOIN orders USING(customer#)  
JOIN orderitems USING(order#)  
JOIN books USING(isbn)  
GROUP BY customer#, city, state, order#;
```

 Results  Script Output  Explain  Autotrace  DBMS Output  OWA Output



CREATE MATERIALIZED succeeded.

Materialized Views (continued)



Summary

- A view is a temporary or virtual table that is used to retrieve data that exists in the underlying database tables
- The view query must be executed each time the view is used
- A view can be used to simplify queries or to restrict access to sensitive data
- A view is created with the CREATE VIEW command
- A view cannot be modified; to change a view, it must be dropped and then re-created, or the CREATE OR REPLACE VIEW command must be used

Summary (continued)

- Any DML operation can be performed on a simple query if it does not violate a constraint
- A view that contains expressions or functions, or that joins multiple tables, is considered a complex view
- A complex view can be used to update only one table; the table must be a key-preserved table
- Data cannot be added to a view column that contains an expression
- DML operations are not permitted on non-key-preserved tables

Summary (continued)

- DML operations are not permitted on views that include group functions, a GROUP BY clause, the ROWNUM pseudocolumn, or the DISTINCT keyword
- Oracle 12c assigns a row number to every row in a table to indicate its position in the table; the row number can be referenced by the keyword ROWNUM
- A view can be dropped with the DROPVIEW command; the data is not affected, because it exists in the original tables
- An inline view can be used only by the current statement and can include an ORDER BY clause
- “TOP-N” analysis uses the row number of sorted data to determine a range of top values
- Materialized views physically store view query results