

About the Presentations

- The presentations cover the objectives found in the opening of each chapter.
- All chapter objectives are listed in the beginning of each presentation.
- You may customize the presentations to fit your class needs.
- Some figures from the chapters are included. A complete set of images from the book can be found on the Instructor Resources disc.

Oracle 12c: SQL

Chapter 5

Data Manipulation and Transaction Control

Objectives

- Use the INSERT command to add a record to an existing table
- Manage virtual columns in data manipulations
- Use quotes in data values
- Use a subquery to copy records from an existing table
- Use the UPDATE command to modify the existing rows of a table
- Use substitution variables with an UPDATE command

Objectives (continued)

- Delete records
- Manage transactions with transaction control commands COMMIT, ROLLBACK, and SAVEPOINT
- Differentiate between a shared lock and an exclusive lock
- Use the SELECT...FOR UPDATE command to create a shared lock

INSERT Command

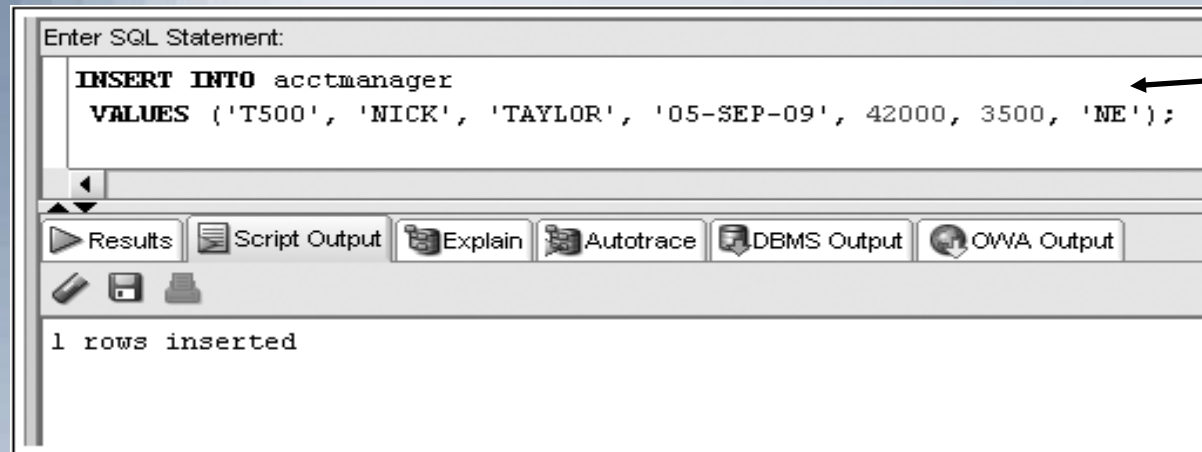
- Used to add rows to existing tables
- Identify the table in the INSERT INTO clause
- Specify data in the VALUES clause
- Can only add one row at a time to a table

INSERT Command Syntax

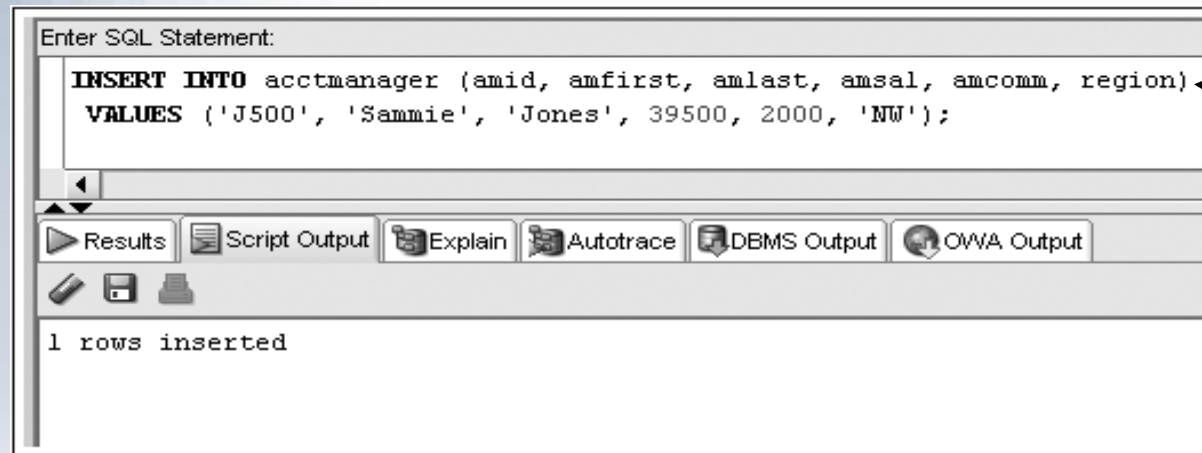
- Enclose nonnumeric data in single quotes
- If a column list is not provided, a value must be assigned to each column in the table

```
INSERT INTO tablename [(columnname, ...)]  
VALUES (datavalue, ...);
```

INSERT Command Examples



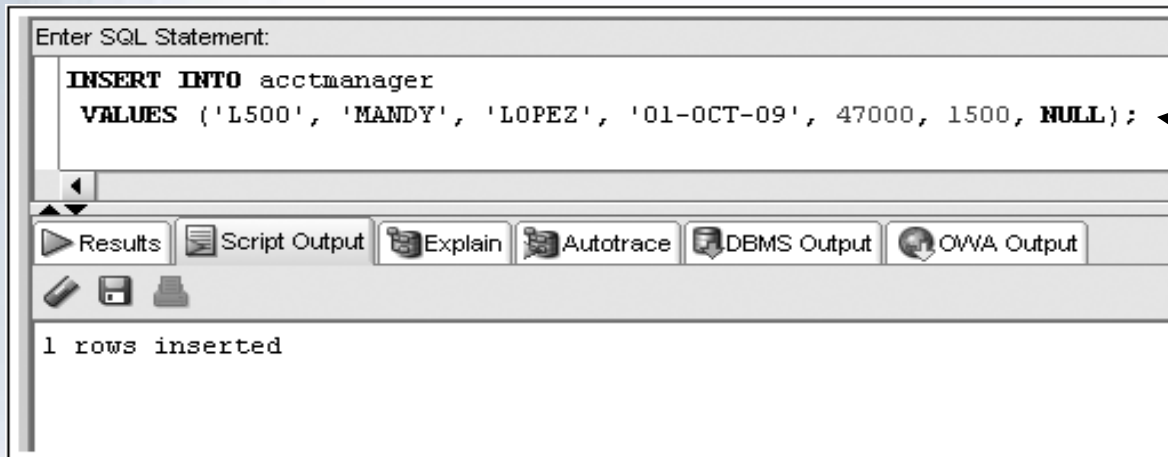
No Column List



Column List

Inserting NULL Value

- Omit column name from INSERT INTO clause column list
- Substitute two single quotation marks
- Use NULL keyword



The screenshot shows a web-based SQL interface. At the top, there is a text area labeled "Enter SQL Statement:" containing the following SQL code:

```
INSERT INTO acctmanager  
VALUES ('L500', 'MANDY', 'LOPEZ', '01-OCT-09', 47000, 1500, NULL);
```

Below the text area is a row of buttons: "Results", "Script Output", "Explain", "Autotrace", "DBMS Output", and "OWA Output". Below these buttons are three icons: a printer, a save icon, and a refresh icon. At the bottom of the interface, it displays the text "1 rows inserted".

NULL value input

ON NULL Clause

- Introduced in Oracle 12c
- Option with a DEFAULT setting

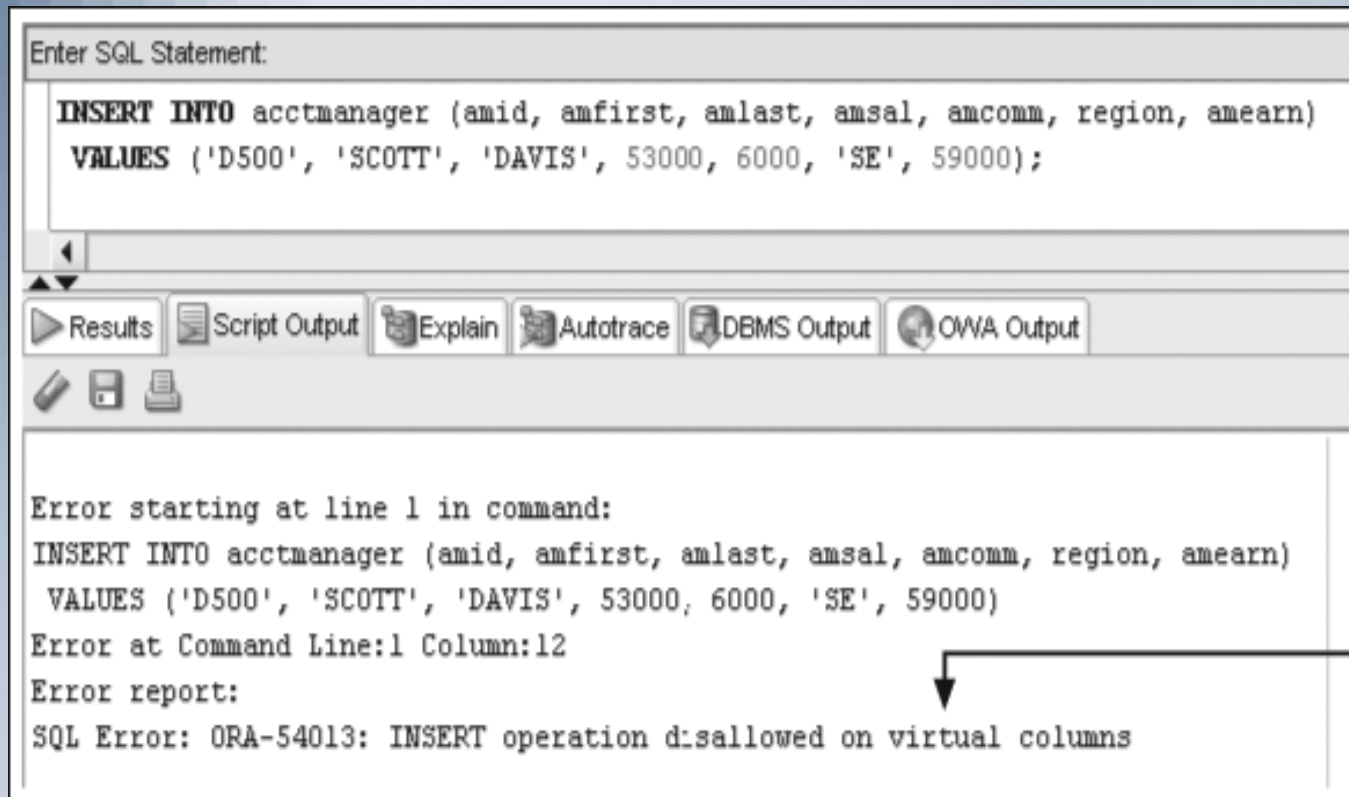
The screenshot shows the Oracle SQL Developer interface. The 'Query Builder' tab is active, displaying the following SQL code:

```
ALTER TABLE acctmanager  
  MODIFY (amsal DEFAULT ON NULL 0);  
  
INSERT INTO acctmanager (amid, amfirst, amlast, amedate, amsal, amcomm, region)  
  VALUES ('J500', 'SAMMIE', 'JONES', NULL, NULL, 2000, 'NW');  
  
SELECT * FROM acctmanager;
```

Below the code editor, the 'Script Output' window shows the results of the query. It indicates 'All Rows Fetched: 1 in 0 seconds'. The results are displayed in a table with the following columns: AMID, AMFIRST, AMLAST, AMEDATE, AMSAL, AMCOMM, and REGION.

	AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
1	J500	SAMMIE	JONES	(null)	0	2000	NW

Manage Virtual Column Input



The screenshot shows a web-based SQL interface. At the top, there is a text area labeled "Enter SQL Statement:" containing the following SQL command:

```
INSERT INTO acctmanager (amid, amfirst, amlast, amsal, amcomm, region, amearn)
VALUES ('D500', 'SCOTT', 'DAVIS', 53000, 6000, 'SE', 59000);
```

Below the text area is a toolbar with buttons for "Results", "Script Output", "Explain", "Autotrace", "DBMS Output", and "OWA Output". Below the toolbar is a large text area displaying the following error message:

```
Error starting at line 1 in command:
INSERT INTO acctmanager (amid, amfirst, amlast, amsal, amcomm, region, amearn)
VALUES ('D500', 'SCOTT', 'DAVIS', 53000, 6000, 'SE', 59000)
Error at Command Line:1 Column:12
Error report:
SQL Error: ORA-54013: INSERT operation disallowed on virtual columns
```

An arrow points from the text "Error indicating that an INSERT isn't allowed on a virtual column" to the error message "SQL Error: ORA-54013: INSERT operation disallowed on virtual columns".

Error indicating
that an INSERT
isn't allowed on
a virtual column

Constraint Violations

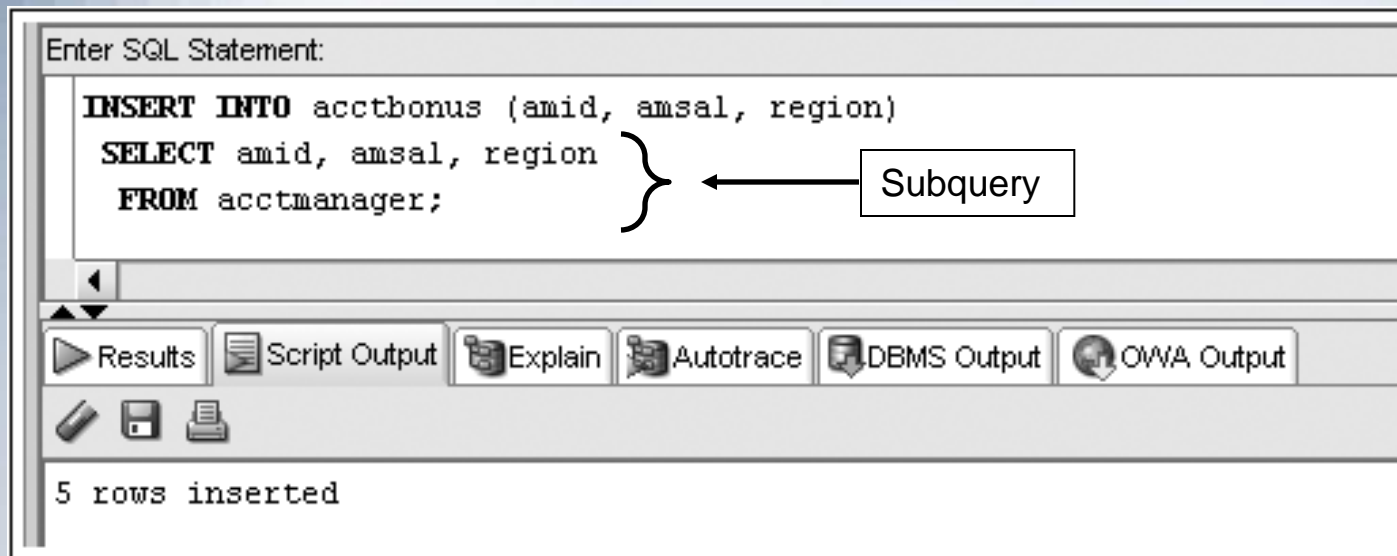
- When you add or modify table data, the data is checked for compliance with any applicable constraints

Activating the DEFAULT option

- Include a column list in the INSERT statement ignoring the column to use the DEFAULT option
- Use the DEFAULT keyword as the value for the column

Inserting Data from an Existing Table

- Substitute subquery for VALUES clause



Modifying Existing Rows

- Modify rows using UPDATE command
- Use UPDATE command to:
 - Add values to an existing row (replace NULL values)
 - Change existing values

UPDATE Command

- UPDATE clause identifies table
- SET clause identifies column(s) being changed and new value(s)
- Optional WHERE clause specifies row(s) to be changed – if omitted, all rows will be updated!

UPDATE Command Syntax

```
UPDATE tablename  
SET columnname = new_datavalue, ...  
[WHERE condition];
```


UPDATE Command Example

Enter SQL Statement:

```
UPDATE acctmanager  
  SET amedate = '10-OCT-09',  
      region = 'S'  
  WHERE amid = '1500';
```

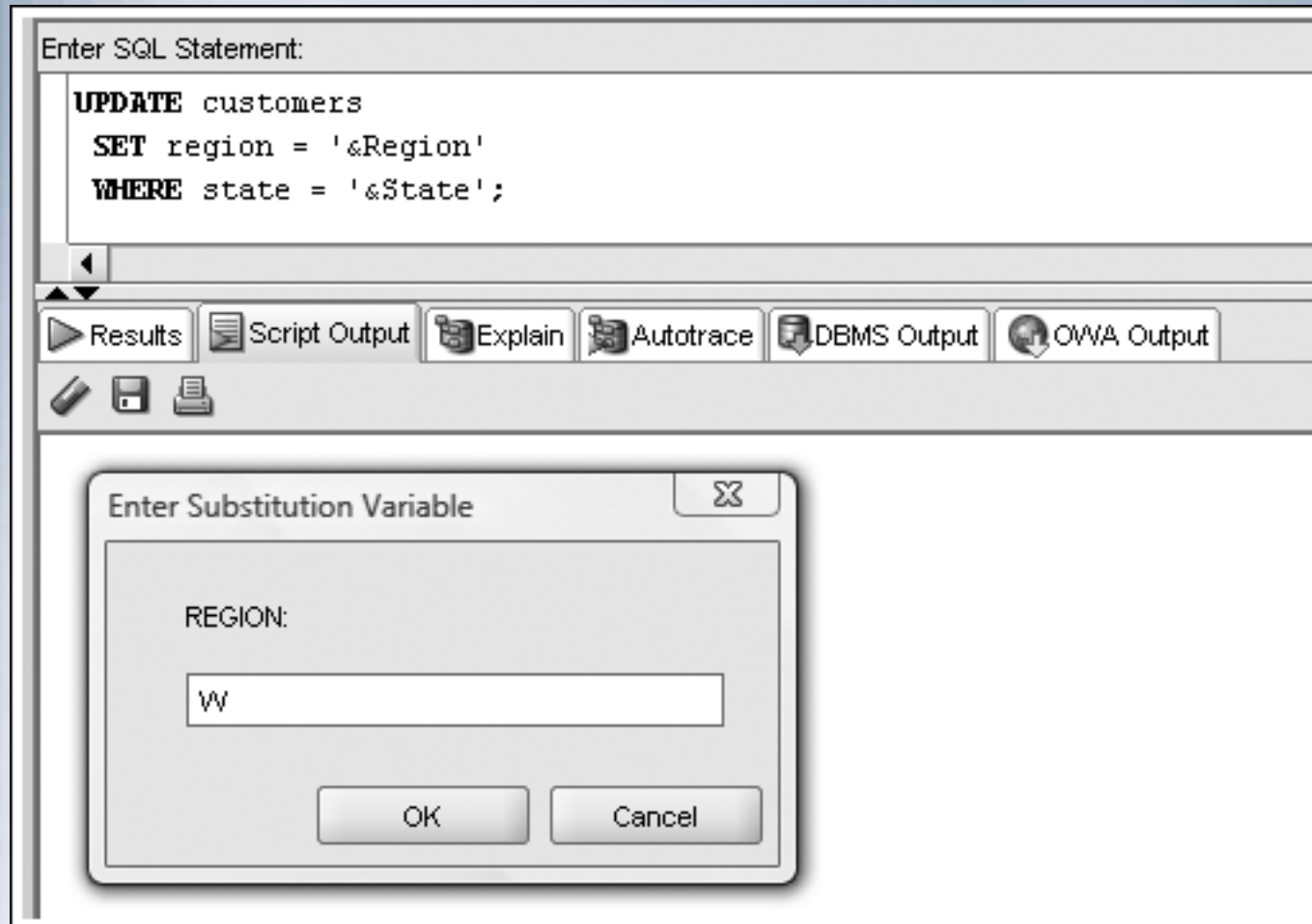
Results Script Output Explain Autotrace DBMS Output OWA Output

1 rows updated

Substitution Variables

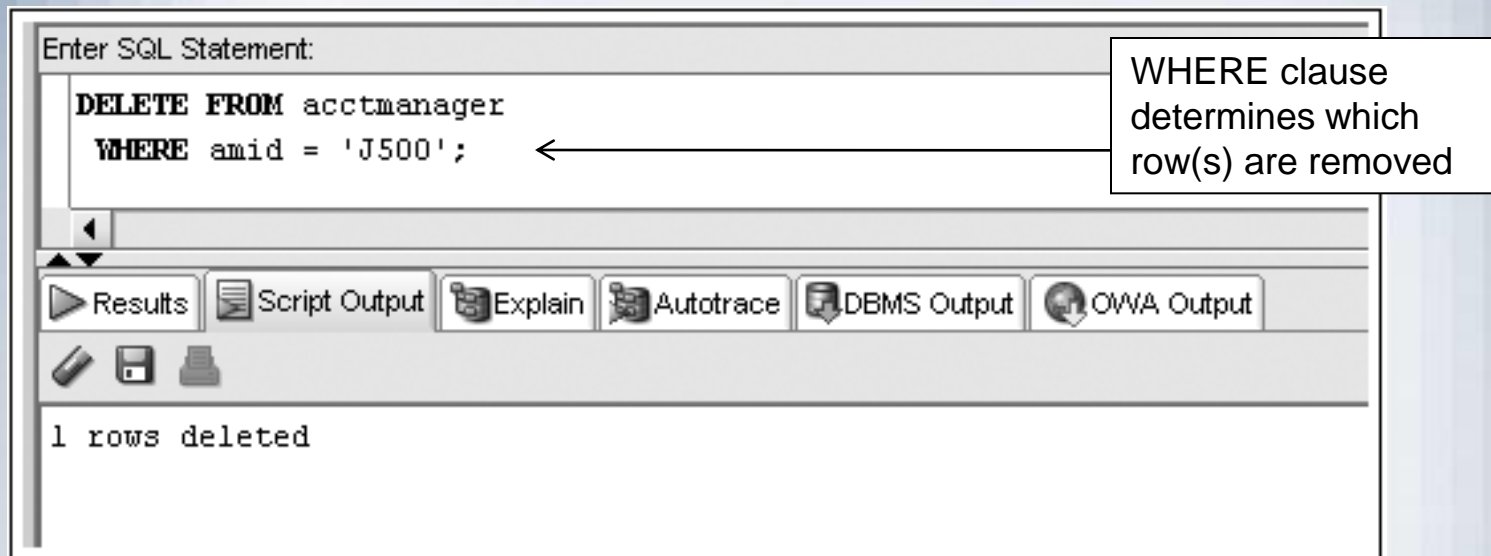
- Prompts user for value
- Identified by ampersand (&) preceding variable name
- Can be used to create interactive scripts

Substitution Variable Example



Deleting Rows

- DELETE command removes a row from a table



DELETE Command – Omitting WHERE Clause

- Omitting WHERE clause removes all rows
- Example below removes all rows from the acctmanager2 table

```
DELETE FROM acctmanager;
```

Transaction Control Statements

- Results of data manipulation language (DML) are not permanently updated to a table until explicit or implicit COMMIT occurs
- Transaction control statements can:
 - Commit data through COMMIT command
 - Undo data changes through ROLLBACK command

COMMIT Command

- Explicit COMMIT occurs by executing COMMIT;
- Implicit COMMIT occurs when DDL command is executed or user properly exits system
- Permanently updates table(s) and allows other users to view changes

ROLLBACK Command

- Used to “undo” changes that have not been committed
- Occurs when:
 - ROLLBACK; is executed
 - System restarts after a crash
- SAVEPOINT marks a specific spot within the transaction
- Can ROLLBACK to a SAVEPOINT to undo part of the transaction

Transaction Control Example

Enter SQL Statement:

```
UPDATE acctmanager
  SET region = 'E'
  WHERE amid = 'M500';
COMMIT;
UPDATE acctmanager
  SET region = 'E'
  WHERE amid = 'T500';
UPDATE acctmanager
  SET region = 'E'
  WHERE amid = 'L500';
SAVEPOINT ONE;
UPDATE acctmanager
  SET amcomm = 6600
  WHERE amid = 'T500';
```

Permanent save with COMMIT

Create a SAVEPOINT

Results Script Output Explain Autotrace DBMS Output OWA Output

1 rows updated
COMMIT succeeded.
1 rows updated
1 rows updated
SAVEPOINT ONE succeeded.
1 rows updated

Transaction Control Example (continued)

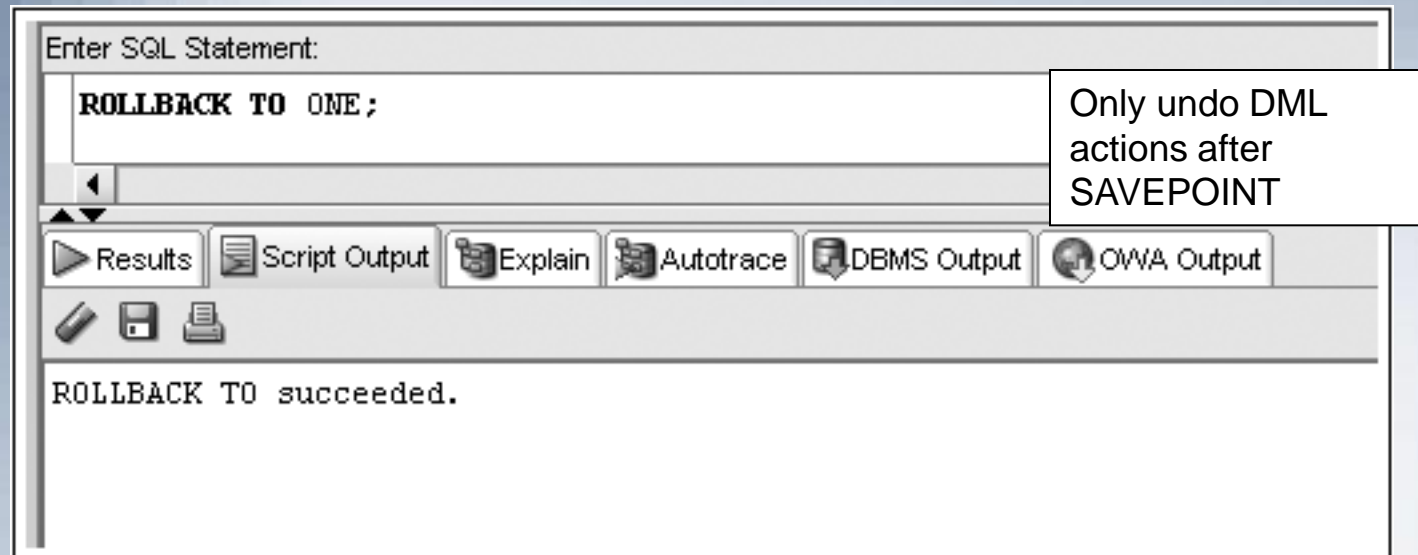


Table Locks

- Prevent users from changing same data or objects
- Two types
 - Shared – prevents DML operations on a portion of table
 - Exclusive – locks table preventing other exclusive or shared locks

LOCK TABLE Command Shared Lock

- Locks portion of table affected by DML operation
- Implicitly occurs during UPDATE or DELETE operations
- Explicitly occurs through LOCK TABLE command with SHARE MODE option
- Released when COMMIT (implicit or explicit) or ROLLBACK occurs

LOCK TABLE Command Exclusive Lock

- Implicitly locks table for DDL operations – CREATE or ALTER TABLE
- Explicitly locked through LOCK TABLE command with EXCLUSIVE MODE option
- Released after execution of DDL operation or after user exits system

SELECT...FOR UPDATE Command

- Creates shared lock on retrieved portion of table
- Prevents one user from changing a row while another user is selecting rows to be changed
- Released through implicit or explicit commit

SELECT...FOR UPDATE Command Syntax

```
SELECT columnnames,...  
FROM tablename, ...  
[WHERE condition]  
FOR UPDATE;
```

Summary

- Data manipulation language (DML) includes the INSERT, UPDATE, DELETE, COMMIT, and ROLLBACK commands
- The INSERT INTO command is used to add new rows to an existing table
- The column list specified in the INSERT INTO clause must match the order of data entered in the VALUES clause
- A virtual column must be ignored in all DML actions because the database system generates this column value automatically
- You can use a NULL value in an INSERT INTO command by including the keyword NULL, omitting the column from the column list of the INSERT INTO clause, or entering two single quotes (without a space) in the position of the NULL value

Summary (continued)

- To assign a DEFAULT option value, a column must be excluded from the column list in an INSERT statement or the keyword DEFAULT must be included as the value for the column
- In a DML statement, two single quotes together must be used to represent a single quote in a value
- If rows are copied from a table and entered in an existing table by using a subquery in the INSERT INTO command, the VALUES clause must be omitted because it's irrelevant
- You can change the contents of a row or group of rows with the UPDATE command
- You can use substitution variables to allow you to execute the same command several times with different data values

Summary (continued)

- DML operations aren't stored permanently in a table until a COMMIT command is issued implicitly or explicitly
- A transaction consists of a set of DML operations committed as a block
- Uncommitted DML operations can be undone by issuing the ROLLBACK command
- A SAVEPOINT serves as a marker for a point in a transaction and allows rolling back only a portion of the transaction
- Use the DELETE command to remove records from a table; if the WHERE clause is omitted, all rows in the table are deleted
- Table locks can be used to prevent users from mistakenly overwriting changes made by other users

Summary (continued)

- Table locks can be in SHARE mode or EXCLUSIVE mode
- EXCLUSIVE mode is the most restrictive table lock and prevents any other user from placing any locks on the same table
- A lock is released when a transaction control statement is issued, a DDL statement is executed, or the user exits the system by using the EXIT command
- SHARE mode allows other users to place shared locks on other portions of the table, but it prevents users from placing an exclusive lock on the table
- The SELECT . . . FOR UPDATE command can be used to place a shared lock for a specific row or rows; the lock isn't released unless a DDL command is issued or the user exits the system