



# Database Management

BU.330.770

Session 4

Instructor: Changmi Jung, Ph.D.

# Announcement



» Assignment #2 is due next week (before the class)

- Available in Week 5 Module
- Late submissions are accepted within 48 hrs of the due but will receive only 50% of the points.

» Have you guys noticed we have a class Q&A and MS Teams?





# Data Manipulation & Transaction Control



# Check Point

» What is DDL (Data Definition Language)?

*create, Alter, drop*

» What is DML (Data Manipulation Language)?

*insert, update, delete*

DQL - DATA Query language → *select*

DCL → Data Control language *grant, revoke*



# Session 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
- » Delete records
- » Manage transactions with transaction control commands COMMIT, ROLLBACK, and SAVEPOINT



# 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 INTO tablename (column1, column2, ...)  
VALUES (value_to_column1, value_to_column2, ...);
```



# INSERT Command - VALUES

- » Enclose nonnumeric data in single quotes

```
INSERT INTO books (isbn, title)
VALUES (1001111111, 'COMPETING ON ANALYTICS');
```

Handwritten notes: An arrow points from the word "Notes" to the SQL code block. Red numbers 1 and 2 are placed above the *isbn* and *title* column names respectively. Red numbers 1 and 2 are placed below the numeric value and the string value in the VALUES clause respectively.

- » The data listed in the VALUES clause **must match the exact order of the columns** listed in the INSERT INTO clause

- 1) isbn = 1001111111, 2) title = 'COMPETING ON ANALYTICS'

- » If a column list is not provided, a value must be assigned to every column in the table in the correct order

```
INSERT INTO books 
VALUES (1001111111, 'COMPETING ON ANALYTICS', SYSDATE, 5,
13.99, 21.99, 0, 'COMPUTER');
```



# INSERT Command Examples

ID	Name	Employment Date	Salary	Commission	Region
T500	Nick Taylor	September 5, 2020	\$42,000.00	\$3,500.00	NE
J500	Sammie Jones	Today	\$39,500.00	\$2,000.00	NW
L500	Mandy Lopez	October 1, 2020	\$47,000.00	\$1,500.00	

Worksheet | Query Builder

```
INSERT INTO acctmanager
VALUES ('T500', 'NICK', 'TAYLOR', '05-SEP-20', 42000, 3500, 'NE');
```

```
INSERT INTO acctmanager (amid, amfirst, amlast, amsal, amcomm, region)
VALUES ('J500', 'Sammie', 'Jones', 39500, 2000, 'NW');
```

Query Result x | Script Output x

Task completed in 0.065 seconds

No Column List

VS.

Column List

1 row inserted.



Character data retains the case you use in the INSERT INTO command

1 row inserted.





# Activating the DEFAULT option

ID	Name	Employment Date	Salary	Commission	Region
J500	Sammie Jones	Today	\$39,500.00	\$2,000.00	NW

1. List columns in the INSERT statement but ignore the column where default value to be entered

```
INSERT INTO acctmanager (amid, amfirst, amlast, amsal, amcomm, region)
VALUES ('J500', 'SAMMIE', 'JONES', 39500, 2000, 'NW');
```

2. Use the DEFAULT keyword as the value for the column

```
INSERT INTO acctmanager (amid, amfirst, amlast, amedate, amsal, amcomm, region)
VALUES ('J500', 'SAMMIE', 'JONES', DEFAULT, 39500, 2000, 'NW');
```

Both 1 and 2 are the same as the following command:

```
INSERT INTO acctmanager (amid, amfirst, amlast, amedate, amsal, amcomm, region)
VALUES ('J500', 'SAMMIE', 'JONES', SYSDATE, 39500, 2000, 'NW');
```

# Activating the DEFAULT Example



1 →  
Ignore the  
column  
from the  
column list

```
Worksheet | Query Builder
INSERT INTO acctmanager (amid, amfirst, amlast, amsal, amcomm, region)
VALUES ('J501', 'SAMMIE', 'JONES', 39500, 2000, 'NW');
```

Script Output x | Query Result x  
Task completed in 0.069 seconds

Omitted 'amedate' column

1 row inserted.

2 →  
Use DEFAULT  
in VALUES  
clause

```
Worksheet | Query Builder
INSERT INTO acctmanager (amid, amfirst, amlast, amedate, amsal, amcomm, region)
VALUES ('J502', 'SAMMIE', 'JONES', DEFAULT, 39500, 2000, 'NW');
```

Script Output x | Query Result x  
Task completed in 0.069 seconds

1 row inserted.



# Inserting NULL Value

ID	Name	Employment Date	Salary	Commission	Region
L500	Mandy Lopez	October 1, 2020	\$47,000.00	\$1,500.00	<i>No data</i>

1. Omit column name from INSERT INTO clause column list

```
INSERT INTO acctmanager (amid, amfirst, amlast, amedate, amsal, amcomm)  
VALUES ('L500', 'MANDY', 'LOPEZ', '01-OCT-20', 47000, 1500);
```

Omit 'region' column in the column list

2. Substitute two single quotation marks in Value clause

```
INSERT INTO acctmanager  
VALUES ('L500', 'MANDY', 'LOPEZ', '01-OCT-20', 47000, 1500, "');
```

Two single quotations where the value for the region column should be

3. Use NULL keyword in Value clause

```
INSERT INTO acctmanager  
VALUES ('L500', 'MANDY', 'LOPEZ', '01-OCT-20', 47000, 1500, NULL);
```

NULL keyword where the value for the region column should be



# Inserting NULL Value Example

1   
Omit 'region'

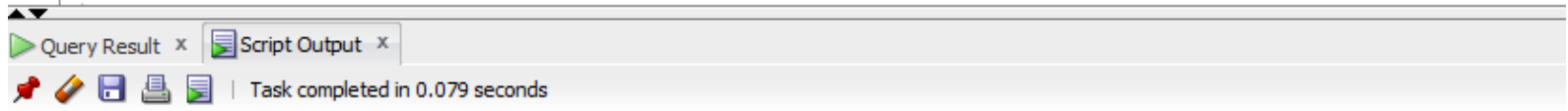
```
INSERT INTO acctmanager (amid, amfirst, amlast, amedate, amsal, amcomm)
VALUES ('L500', 'MANDY', 'LOPEZ', '01-OCT-20', 47000, 1500);
```

2   
Use single quotes

```
INSERT INTO acctmanager
VALUES ('L501', 'MANDY', 'LOPEZ', '01-OCT-20', 47000, 1500, ');
```

3   
Use NULL

```
INSERT INTO acctmanager
VALUES ('L502', 'MANDY', 'LOPEZ', '01-OCT-20', 47000, 1500, NULL);
```



1 row inserted.

1 row inserted.

1 row inserted.



# ON NULL Clause

- » Default column value will be overridden when you insert an explicit NULL value: using either two single quotes or keyword NULL
- » Prevent the overriding with **ON NULL** option in a DEFAULT setting

```
ALTER TABLE tablename  
MODIFY (columnname DEFAULT ON NULL defaultvalue);
```

Notes

Worksheet Query Builder

```
ALTER TABLE acctmanager  
MODIFY (amsal DEFAULT ON NULL 0);  
  
INSERT INTO acctmanager (amid, amfirst, amlast, amedate, amsal, amcomm, region)  
VALUES ('B500', 'CHARLIE', 'BROWN', '', NULL, 0, 'SW');  
  
SELECT * FROM acctmanager;
```

Script Output x Query Result x

SQL | All Rows Fetched: 9 in 0.015 seconds

	AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
1	B500	CHARLIE	BROWN	(null)	0	0	SW
2	T500	NICK	TAYLOR	05-SEP-20	42000	3500	NE
3	J500	Sammie	Jones	10-FEB-23	39500	2000	NW

# View DEFAULT Option



Worksheet    Query Builder

```
SELECT column_name, data_default, default_on_null
FROM user_tab_columns
WHERE table_name = 'ACCTMANAGER';
```

Script Output x    Query Result x

SQL | All Rows Fetched: 7 in 0.22 seconds

	COLUMN_NAME	DATA_TYPE	DATA_DEFAULT	DEFAULT_ON_NULL
1	AMID	CHAR	(null)	NO
2	AMFIRST	VARCHAR2	(null)	NO
3	AMLAST	VARCHAR2	(null)	NO
4	AMEDATE	DATE	SYSDATE	NO
5	AMSAL	NUMBER	0	YES
6	AMCOMM	NUMBER	0	NO
7	REGION	CHAR	(null)	NO

force default in  
Case of null.

default values



# ON NULL Clause

- » You can't set this DEFAULT ON NULL if the column already contains NULL value

```
Worksheet | Query Builder

ALTER TABLE acctmanager
MODIFY amedate DEFAULT ON NULL SYSDATE;

Query Result x | Script Output x
Task completed in 0.094 seconds
```

Error starting at line : 82 in command -

```
ALTER TABLE acctmanager
MODIFY amedate DEFAULT ON NULL SYSDATE
```

Error report -

ORA-02296: cannot enable (ADMIN.) - null values found

02296. 00000 - "cannot enable (%s.%s) - null values found"

\*Cause: an alter table enable constraint failed because the table contains values that do not satisfy the constraint.

\*Action: Obvious

*We inserted NULL to Charlie Brown's amedate...*



# Manage Virtual Column Input

```
Worksheet | Query Builder
ALTER TABLE acctmanager
ADD amearn AS (amsal + amcomm);

DESC acctmanager;
```

Script Output x

Task completed in 0.22 seconds

Let's add a virtual column first

Table ACCTMANAGER altered.

Name	Null?	Type
-----	-----	-----
AMID	NOT NULL	CHAR(4)
AMFIRST	NOT NULL	VARCHAR2(12)
AMLAST	NOT NULL	VARCHAR2(12)
AMEDATE		DATE
AMSAL	NOT NULL	NUMBER(8,2)
AMCOMM		NUMBER(7,2)
REGION		CHAR(2)
AMEARN		NUMBER





# Manage Virtual Column Input

Worksheet | Query Builder

```
82 INSERT INTO acctmanager
83 VALUES ('D500', 'SCOTT', 'DAVIS', DEFAULT, 53000, 6000, 'SE', 59000);
84
```

Value to amend

Query Result x | Script Output x

Task completed in 0.061 seconds

Error starting at line : 82 in command -

```
INSERT INTO acctmanager
VALUES ('D500', 'SCOTT', 'DAVIS', DEFAULT, 53000, 6000, 'SE', 59000)
```

Error at Command Line : 82 Column : 1

Error report -

```
SQL Error: ORA-54013: INSERT operation disallowed on virtual columns
54013. 0000 - "INSERT operation disallowed on virtual columns"
```

\*Cause: Attempted to insert values into a virtual column

\*Action: Re-issue the statment without providing values for a virtual column

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

To correct this, remove the virtual column and value, then reissue the statement.



# Handling Single Quote

ID	Name	Employment Date	Salary	Commission	Region
O500	Peg O'Hara	Today	\$46,000.00	\$2,000.00	SW

```
Worksheet  Query Builder
INSERT INTO acctmanager (amid, amfirst, amlast, amsal, amcomm, region)
VALUES ('O500', 'PEG', 'O'HARA', 46000, 2000, 'SW');
```

Query Result x Script Output x  
Task completed in 0.095 seconds

*How to insert the name, O'Hara?*

Error starting at line : 94 in command -

```
INSERT INTO acctmanager (amid, amfirst, amlast, amsal, amcomm, region)
VALUES ('O500', 'PEG', 'O'HARA', 46000, 2000, 'SW');
```

*Notes*

Enter two single quotes together

Error at Command Line : 95 Colu

Error report -

SQL Error: ORA-00917: missing c  
00917. 00000 - "missing comma"

\*Cause:

\*Action:

```
Worksheet  Query Builder
INSERT INTO acctmanager (amid, amfirst, amlast, amsal, amcomm, region)
VALUES ('O500', 'PEG', 'O''HARA', 46000, 2000, 'SW');
```

Query Result x Script Output x  
Task completed in 0.065 seconds

1 row inserted.



# Inserting Data from an Existing Table

## » Substitute Subquery for VALUES clause

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

Don't need  
parentheses

No VALUES  
clause!!!

```
Worksheet | Query Builder  
INSERT INTO acctbonus (amid, amsal, region)  
SELECT amid, amsal, region  
FROM acctmanager;
```

Subquery

Notes.

Query Result x | Script Output x  
Task completed in 0.079 seconds

5 rows inserted.



# Modifying Existing Rows

- » Modify rows using **UPDATE** command
- » Use UPDATE command to:
  - Add values to an existing row (replace NULL in some columns)
  - Change existing values

*Assign an account manager to a marketing region*



*A customer moved to a new address*

*An employee's information is incorrectly entered (e.g. employment date)*



# UPDATE Command Syntax

```
UPDATE tablename
SET columnname1 = new_datavalue1,
    columnname2 = new_datavalue2,
    ...
[WHERE condition];
```

No ~~tes~~

- » 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 Example

Worksheet    Query Builder

UPDATE acctmanager

SET amedate = '01-AUG-20',  
amsal = 58000

WHERE amid = 'B500';

SELECT \*

FROM acctmanager

WHERE amid = 'B500';

Script Output x    Query Result x

SQL | All Rows Fetched: 1 in 0.014 seconds

	AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
1	B500	CHARLIE	BROWN	01-AUG-20	58000	0	SW

Charlie Brown's employment date is Null,  
and his salary is zero (we inserted Null  
during the practice)

*Let's change the date to Aug. 1,  
2020 and salary to 58,000*

# UPDATE Command Example



Worksheet    Query Builder

```
UPDATE acctmanager
SET amedate = '15-OCT-20',
    region = 'S'
WHERE amid = 'L500';

SELECT *
FROM acctmanager
WHERE amid = 'L500';
```

Script Output x    Query Result x

SQL | All Rows Fetched: 1 in 0.014 seconds

	AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
1	L500	MANDY	LOPEZ	15-OCT-20	47000	1500	S

*Let's change Mandy Lopez's employment date (amedate) to Oct. 15, 2020 and region to 'S.'*

*From 01-OCT-20*

*From null*



# Substitution Variables

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



What if I need to update the region column for all customers in the Customers table?



# Substitution Variable Example



```
Worksheet | Query Builder

ALTER TABLE customers
ADD region CHAR(2);

UPDATE customers
SET region = '&Customer_Region'
WHERE state = '&Customer_State';
```

Query Result x | Script Output x

Task completed in 11.982 seconds

```
old:UPDATE customers
SET region = '&Customer_Region'
WHERE state = '&Customer_State'
new:UPDATE customers
SET region = 'W'
WHERE state = 'CA'
```

3 rows updated.

Enter Substitution Variable

Enter value for Customer\_Region:

W

OK Cancel

Enter Substitution Variable

Enter value for Customer\_State:

CA

OK Cancel



# Deleting Rows

- » DELETE command removes a row (rows) from a table
- » DELETE applies to an entire row (can't be applied to specific columns in a row)

```
Worksheet | Query Builder
DELETE FROM acctmanager
WHERE amid = 'J500';
```

Query Result x | Script Output x

Task completed in 0.038 seconds

WHERE clause determines which row(s) is(are) removed

1 row deleted.

Sammie Jones (amid = J500) moved to the Customer Service department - no longer listed in the ACCTMANAGER table

```
Worksheet | Query Builder
DELETE FROM acctmanager
WHERE amid IN ('L501', 'L502')
```

Query Result x | Script Output x

Task completed in 0.026 seconds

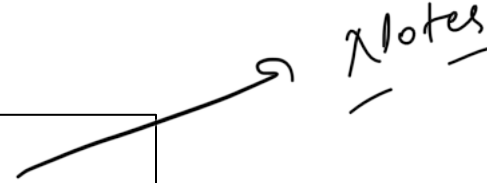
2 rows deleted.



# DELETE Command – Omitting WHERE Clause

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

```
DELETE FROM acctmanager;
```



*Recap: Difference between  
DELETE FROM tablename vs. TRUNCATE TABLE tablename?*



# 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

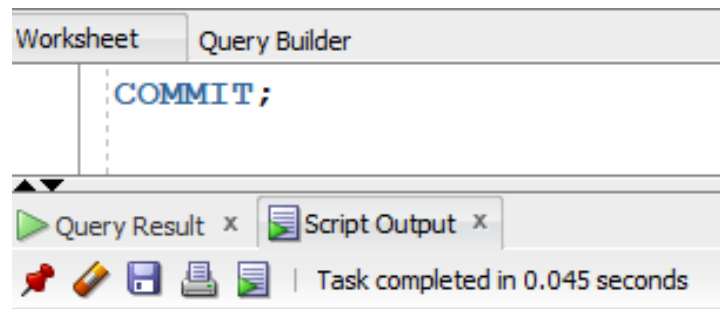
Imagine you are withdrawing \$1,000 and transferring the money to another account. Before transferring but after withdrawing, the system crashes. Would you lose \$1,000 from your account?



# COMMIT Command



- » Explicit COMMIT occurs by executing COMMIT;
- » Implicit COMMIT occurs when (1) DDL command is executed or (2) user properly exits system  
*CREATE TABLE.... ALTER TABLE... TRUNCATE TABLE...*
- » Permanently updates table(s) and allows other users to view changes



Commit complete.



# ROLLBACK Command

- » Used to “undo” changes that have not been committed
- » Occurs when:
  - ROLLBACK; is executed
  - System restarts after a crash
- » DDL commands can’t be rolled back (auto committed)
- » SAVEPOINT marks a specific spot within the transaction
- » We can ROLLBACK to a SAVEPOINT to undo part of the transaction

# DDL vs. DML



Worksheet Query Builder



```
CREATE TABLE bookauthor1  
AS (SELECT *  
FROM bookauthor);
```

```
SELECT *  
FROM bookauthor;
```

```
SELECT *  
FROM bookauthor1;
```

```
TRUNCATE TABLE bookauthor1;
```

```
DELETE FROM bookauthor;
```

```
ROLLBACK;
```

Notes

Both BOOKAUTHOR and  
BOOKAUTHOR1 have 20 records

→ DDL is auto-committed (can't ROLLBACK)

→ DML can be undone if not committed

Check which command is undone



# Transaction Control Practice (1/3)

First, let's set `amcomm = 0` for all for our convenience

Worksheet | Query Builder

```
UPDATE acctmanager
SET amcomm = 0;

SELECT *
FROM acctmanager;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 5 in 0.018 seconds

	AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
1	T500	NICK	TAYLOR	05-SEP-20	42000	0	NE
2	B500	CHARLIE	BROWN	01-AUG-20	58000	0	SW
3	D500	SCOTT	DAVIS	09-FEB-24	53000	0	SE
4	O500	PEG	O'Hara	09-FEB-24	46000	0	SW
5	L500	MANDY	LOPEZ	15-OCT-20	47000	0	S

Worksheet | Query Builder

```
COMMIT;
```

Query Result x | Script Output x

Task completed in 0.045 seconds

Commit complete.



# Transaction Control Practice (2/3)



Worksheet Query Builder

1

```
UPDATE acctmanager  
SET amcomm = 1000  
WHERE amid = 'L500';
```

COMMIT; → Permanent save with COMMIT

2

```
UPDATE acctmanager  
SET amcomm = 2000  
WHERE amid = 'T500';
```

SAVEPOINT one; → Create a SAVEPOINT

3

```
UPDATE acctmanager  
SET amcomm = 3000  
WHERE amid = 'B500';
```

SAVEPOINT two; → Create a SAVEPOINT

4

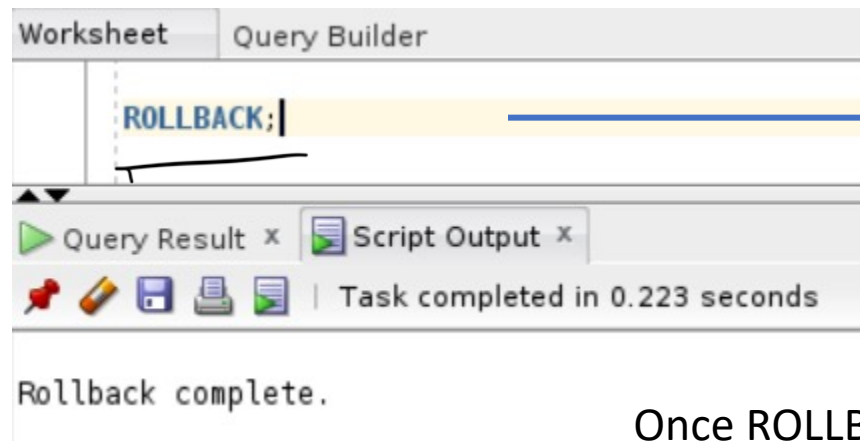
```
UPDATE acctmanager  
SET amcomm = 4000  
WHERE amid = 'D500';
```



What will happen if we issue a ROLLBACK command?

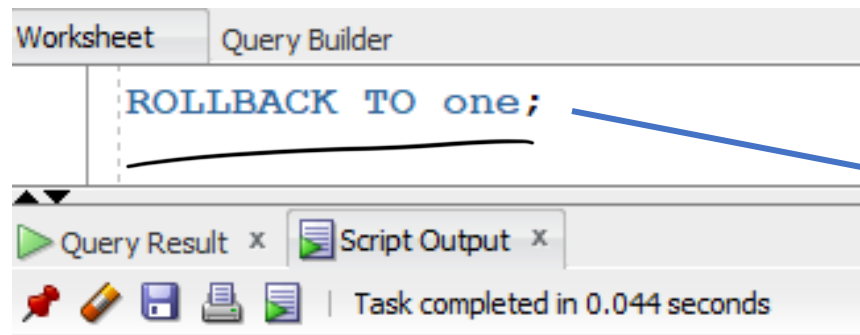
What if we want to undo only the last UPDATE?

# Transaction Control Practice (3/3)



Undo all DML actions after COMMIT

Once ROLLBACK is executed, SAVEPOINT must be re-established.



Only undo DML actions performed after SAVEPOINT

Rollback complete.



# Results of ROLLBACK (to SAVEPOINT)

AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
T500	NICK	TAYLOR	05-SEP-20	42000	1000	NE
L500	MANDY	LOPEZ	15-OCT-20	47000	2000	S
B500	CHARLIE	BROWN	01-AUG-20	58000	3000	SW
D500	SCOTT	DAVIS	10-FEB-24	53000	4000	SE
O500	PEG	O'Hara	10-FEB-24	46000	0	SW

1  
2  
3  
4

➡ After all updates

AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
T500	NICK	TAYLOR	05-SEP-20	42000	1000	NE
L500	MANDY	LOPEZ	15-OCT-20	47000	0	S
B500	CHARLIE	BROWN	01-AUG-20	58000	0	SW
D500	SCOTT	DAVIS	10-FEB-24	53000	0	SE
O500	PEG	O'Hara	10-FEB-24	46000	0	SW

1

➡ After **ROLLBACK;**

All reset to original value except for (1)

AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
T500	NICK	TAYLOR	05-SEP-20	42000	1000	NE
L500	MANDY	LOPEZ	15-OCT-20	47000	2000	S
B500	CHARLIE	BROWN	01-AUG-20	58000	0	SW
D500	SCOTT	DAVIS	10-FEB-24	53000	0	SE
O500	PEG	O'Hara	10-FEB-24	46000	0	SW

1  
2

➡ After **ROLLBACK to one;**

All reset to original value except for (1) and (2)

AMID	AMFIRST	AMLAST	AMEDATE	AMSAL	AMCOMM	REGION
T500	NICK	TAYLOR	05-SEP-20	42000	1000	NE
L500	MANDY	LOPEZ	15-OCT-20	47000	2000	S
B500	CHARLIE	BROWN	01-AUG-20	58000	3000	SW
D500	SCOTT	DAVIS	10-FEB-24	53000	0	SE
O500	PEG	O'Hara	10-FEB-24	46000	0	SW

1  
2  
3

➡ After **ROLLBACK to two;**

Only the update (4) is reset to zero

# Summary (1/4)



- » Data manipulation language (DML) includes the INSERT, UPDATE, DELETE, and TCL includes 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



## Summary (2/4)

- » 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
- » You can use a NULL value in an INSERT INTO command by omitting the column from the column list of the INSERT INTO clause, entering two single quotes (without a space) in the position of the NULL value, or including the keyword NULL.
- » In a DML statement, two single quotes together must be used to represent a single quote in a value *Remember, Peg O'Hara*
- » 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

# Summary (3/4)



- » 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
- » Use the DELETE command to remove records from a table; if the WHERE clause is omitted, all rows in the table are deleted

# Summary (4/4)



- » 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

# Let's Check Our Learning!

The Kahoot! logo is centered on a dark purple rectangular background. The word "Kahoot!" is written in a bold, white, sans-serif font. A lighter purple arrow points from the right towards the text.

**Kahoot!**