



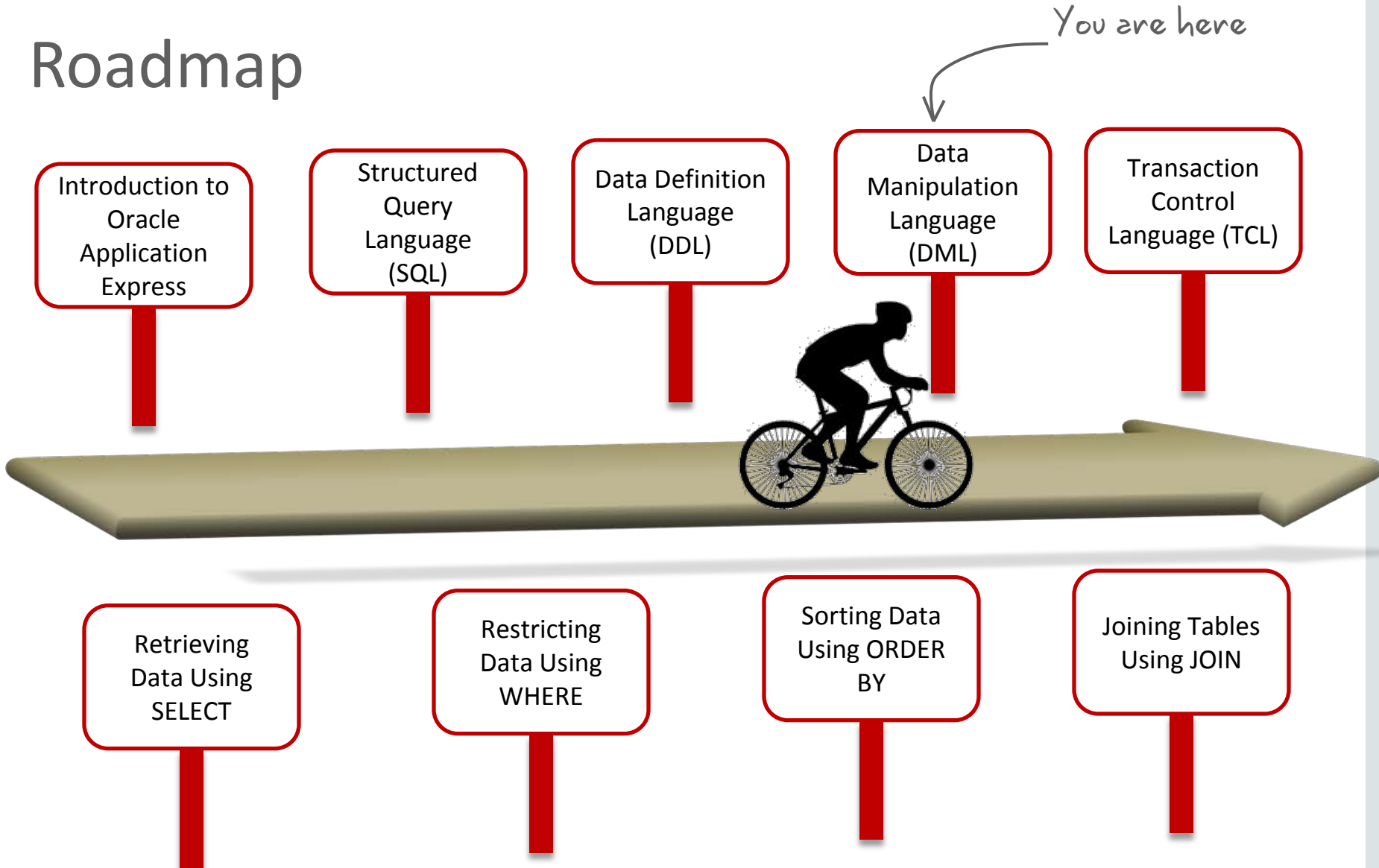
Database Foundations

6-4

Data Manipulation Language (DML)



Roadmap



Objectives

This lesson covers the following objectives:

- Describe the purpose of the data manipulation language (DML)
- Explain the DML operations that are required to manage a database's table data:
 - Insert
 - Update
 - Delete



Data Manipulation Language

- A DML statement is executed when you:
 - Add new rows to a table
 - Modify existing rows in a table
 - Remove existing rows from a table
- A transaction consists of a collection of DML statements that form a logical unit of work.



Adding a New Row to a Table

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
80	Sales	145	2500

New row

70	Public Relations	204	2700
----	------------------	-----	------



Insert new row into the DEPARTMENTS table.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500

INSERT Statement Syntax

- Add rows to a table by using the INSERT statement:

```
INSERT INTO  table [(column [, column...])]  
VALUES      (value [, value...]);
```

- With this syntax, only one row is inserted at a time.

Inserting Rows

- Insert a row containing values for each column.

```
INSERT INTO departments
VALUES (80, 'Marketing', 400, 1900);
```

- List values in the default order of the columns in the table.
- Alternatively, list the columns in the INSERT clause.

```
INSERT INTO departments(department_id,
                        department_name, manager_id, location_id)
VALUES (70, 'Public Relations', 100, 1700);
```

- Enclose character and date values within single quotation marks.

Case Scenario: Inserting Rows

I understand that the `INSERT` statement is used to add rows to a table. Is it possible to insert a row if some columns do not have values?



Student



Faculty

Absolutely. You can use the `INSERT` statement to specify the columns for which values will not be entered.

Inserting Rows with Null Values

- Implicit method: Omit the column from the column list.

```
INSERT INTO departments (department_id,  
                          department_name)  
VALUES (30, 'Purchasing');
```

- Explicit method: Specify the NULL keyword in the VALUES clause.

```
INSERT INTO departments  
VALUES (100, 'Finance', NULL, NULL);
```

Case Scenario: Inserting Rows

```
1 row(s) inserted.  
1 row(s) inserted.  
1 row(s) inserted.
```

```
INSERT INTO AUTHOR VALUES ('AN0002', 'Oscar Wilde', NULL);  
INSERT INTO AUTHOR VALUES ('AN0003', 'George Shaw', NULL);  
INSERT INTO AUTHOR VALUES ('AN0004', 'Leo', NULL);
```

Rows inserted
successfully



Inserting Special Values

The `SYSDATE` function records the current date and time.

```
INSERT INTO employees (employee_id,  
                        first_name, last_name,  
                        email, phone_number,  
                        hire_date, job_id, salary,  
                        commission_pct, manager_id,  
                        department_id)  
VALUES (113,  
        'Louis', 'Popp',  
        'LPOPP', '515.124.4567',  
        SYSDATE, 'AC_ACCOUNT', 6900,  
        NULL, 205, 110);
```

Inserting Specific Date and Time Values

- Add an employee.

```
INSERT INTO employees
VALUES      (114,
            'Den', 'Raphealy',
            'DRAPHEAL', '515.127.4561',
            TO_DATE('Dec 7, 2002', 'MON DD, YYYY'),
            'SA_REP', 11000, 0.2, 100, 60);
```

1 rows inserted

- Verify your addition.

114	Den	Raphealy	DRAPHEAL	515.127.4561	07-DEC-02
-----	-----	----------	----------	--------------	-----------

Changing Data in a Table

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-06	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-07	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-05	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-06	IT_PROG	4800	-	103	60

Update rows in the EMPLOYEES table:



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-06	IT_PROG	9000	-	102	80
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-07	IT_PROG	6000	-	103	80
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-05	IT_PROG	4800	-	103	80
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-06	IT_PROG	4800	-	103	60

UPDATE Statement Syntax

- Modify existing values in a table with the UPDATE statement:

```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE     condition];
```

- Update more than one row at a time (if required).

Updating Rows in a Table

- Values for a specific row or rows are modified if you specify the `WHERE` clause:

```
UPDATE employees
SET    department_id = 50
WHERE  employee_id = 113;
```

- Values for all the rows in the table are modified if you omit the `WHERE` clause:

```
UPDATE    copy_emp
SET       department_id = 110;
```

- Specify `SET column_name= NULL` to update a column value to `NULL`.

Violating Constraints

```
UPDATE employees  
SET    department_id = 55  
WHERE  department_id = 110;
```

```
ORA-02291: integrity constraint (APEXWS2.EMP_DEPT_FK) violated - parent  
key not found
```

Department 55 does not exist.

Removing a Row from a Table

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700

Delete a row from the
DEPARTMENTS table:

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
80	Sales	145	2500
90	Executive	100	1700

DELETE Statement

You can remove existing rows from a table by using the DELETE statement:

```
DELETE [FROM] table  
[WHERE condition];
```

Deleting Rows from a Table

- Specific rows are deleted if you include the `WHERE` clause:

```
DELETE FROM departments
WHERE department_name = 'Finance';
```

- All rows in the table are deleted if you omit the `WHERE` clause:

```
DELETE FROM copy_emp;
```

Violating Constraints

You cannot delete a row that contains a primary key which is used as a foreign key in another table.

```
DELETE FROM departments  
WHERE department_id = 60;
```

```
ORA-02292: integrity constraint (APEXWS2.JHIST_DEPT_FK) violated - child  
record found
```

Case Scenario: Deleting Rows

Is it possible to delete all rows in a table but leave the structure intact?



Student



Faculty

Absolutely. You can use the TRUNCATE statement to do that.

TRUNCATE Statement

- Removes all rows from a table, leaving the table empty and the table structure intact
- Is a DDL statement rather than a DML statement; cannot easily be undone

Syntax:

```
TRUNCATE TABLE table_name;
```

Example:

```
TRUNCATE TABLE copy_emp;
```

Summary

In this lesson, you should have learned how to:

- Describe the purpose of DML
- List the DML operations that are required to manage a database's table data:
 - Insert
 - Update
 - Delete



