ORACLE®

ACADEMY

Database Programming with PL/SQL

3-1
Review of SQL DML

Objectives

This lesson covers the following objectives:

- Insert data into a database table
- Update data in a database table
- Delete data from a database table
- Merge data into a database table

Purpose

- When you create, change, or delete an object in a database, the language you use is referred to as data definition language (DDL).
- When you change data within an object (inserting rows, deleting rows, changing column values within a row), it is called data manipulation language (DML).
- This lesson reviews basic SQL DML statements.
- Later, you will use DML statements in your PL/SQL code to modify data.

Data Manipulation Language (DML)

- You can use DML commands to modify the data in a database table.
- The DML commands are `INSERT`, `UPDATE`, `DELETE`, and `MERGE`.

INSERT

- You use the INSERT statement to add new rows to a table.
- It requires at least two items:
  - The name of the table
  - Values for each of the columns in the table
  - (Optional, but recommended) The names of the columns that will receive a value when the row is inserted.
  - If this option is used, then the list of values must match the list of columns.

INSERT Explicit Syntax

- The syntax shown explicitly lists each column in the table that can not be NULL plus the column for the employee's first name.
- The values for each column must be listed in the same order as the columns are listed.

```
INSERT INTO employees (employee_id, first_name,
      last_name, email, hire_date, job_id)
   VALUES (305, 'Kareem', 'Naser',
      'naserk@oracle.com', SYSDATE, 'SR_SA_REP');
```

INSERT Implicit Syntax

- Another way to insert values in a table is to implicitly add them without listing the column names.
- The values must match the order in which the columns appear in the table and a value must be provided for each column.

```
INSERT INTO employees
   VALUES (305, 'Kareem', 'Naser',
      'naserk@oracle.com', '111-222-3333', SYSDATE,
      'SR_SA_REP', 7000, NULL, NULL, NULL);
```

UPDATE

- The UPDATE statement is used to modify existing rows in a table.
- It requires at least three items:
  - The name of the table
  - The name of at least one column to modify
  - A value for each column being modified
  - (Optional) a WHERE clause that identifies the row or rows to be modified

UPDATE Syntax

- A single column can be modified.
- The WHERE clause identifies the row to be modified.

```
UPDATE employees
   SET salary = 11000
   WHERE employee_id = 176;
```

- An UPDATE statement can modify multiple columns.

```
UPDATE employees
   SET salary = 11000, commission_pct = .3
   WHERE employee_id = 176;
```

DELETE

- You use the DELETE statement to remove existing rows in a table.
- The statement requires at least one item:
  - The name of the table
  - (Optional) a WHERE clause that identifies the row or rows to be deleted
- Please note, if the WHERE clause is omitted, ALL rows will be deleted.
- Be very careful when running a DELETE statement without a WHERE clause. Situations requiring that will be rare.

DELETE Syntax

- The WHERE clause identifies the row or rows to be deleted.

```
DELETE FROM employees
  WHERE employee_id = 149;
```

```
DELETE FROM employees
  WHERE department_id = 80;
```

- Be careful with the DELETE statement.
- If the WHERE clause is omitted, ALL rows will be deleted.
- Very few situations will require a DELETE statement without a WHERE clause.

MERGE

- The MERGE statement will INSERT a new row into a target table or UPDATE existing data in a target table, based on a comparison of the data in the two tables.
- The WHEN clause determines the action to be taken.
- For example, if a specific row exists in the source table, but there is no matching row in the target table, the row from the source table will be inserted into the target table.
- If the matching row does exist in the target table, but some data in the source table is different for that row, the target table may be updated with the different data.

MERGE Usage and Syntax

- To set up our MERGE example, consider a situation where we need to calculate annual bonuses for employees earning less than $10,000 USD.
- First, we create a table called bonuses.

```
CREATE TABLE bonuses
   (employee_id NUMBER(6,0) NOT NULL,
    bonus NUMBER(8,2) DEFAULT 0);
```

MERGE Usage and Syntax

We then populate the table with the employee ids of all employees with a salary less than $10,000 USD.

```
INSERT INTO bonuses(employee_id)
   (SELECT employee_id FROM employees
   WHERE salary < 10000);
```

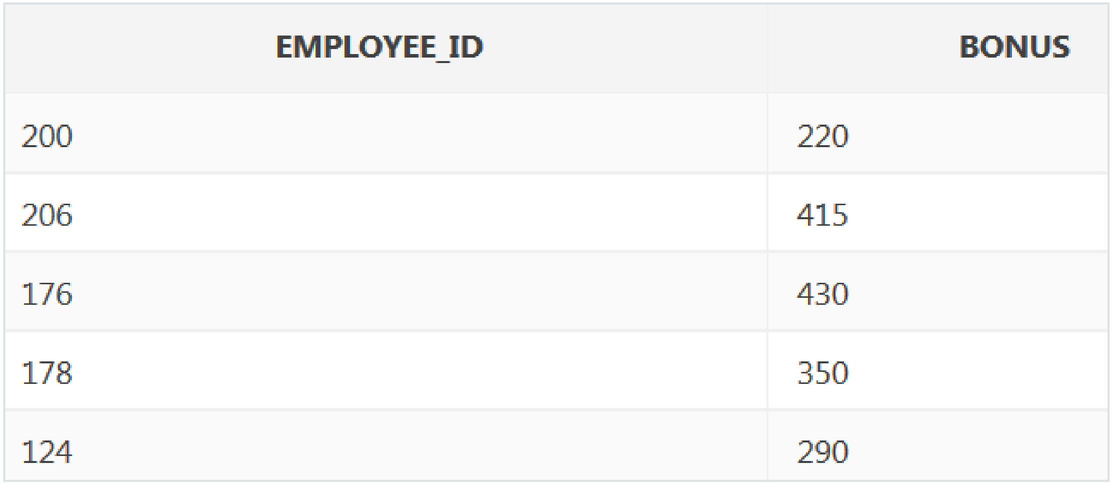
MERGE Usage and Syntax

- Each employee with a salary less than $10,000 USD is to receive a bonus of 5% of their salary.
- To use the salary column from the employees table to calculate the amount of the bonus and update the bonus column in the bonuses table, we use the following MERGE statement.

```
MERGE INTO bonuses b
   USING employees e
   ON (b.employee_id = e.employee_id)
   WHEN MATCHED THEN
      UPDATE SET b.bonus = e.salary * .05;
```

MERGE Usage and Syntax

- The resulting bonus table will look something like this:



EMPLOYEE_ID	BONUS
200	220
206	415
176	430
178	350
124	290

- If you attempt to duplicate this example, your results may vary depending on the data in your employees table.

Terminology

Key terms used in this lesson included:

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- DELETE
- INSERT
- MERGE
- UPDATE

Summary

In this lesson, you should have learned how to:

- Insert data into a database table
- Update data in a database table
- Delete data from a database table
- Merge data into a database table

ORACLE® ACADEMY