# Practices for Lesson 19: Manipulating Data Using Advanced Queries

Practices for Lesson 19: Overview

Practice Overview

This practice covers the following topics:

Performing multitable INSERTs

Performing MERGE operations

Performing flashback operations

Tracking row versions

**Note:** Before starting this practice, execute the **/home/oracle/labs/sql2/code\_ex/ cleanup\_scripts/cleanup\_19.sql** script.

Practice 19-1: Manipulating Data

Overview

In this practice, you perform multitable INSERT and MERGE operations, and the flashback operation, and track row versions.

**Note:** Execute the cleanup\_19.sql script from **/home/oracle/labs/sql2/code\_ex/ cleanup\_scripts/** before performing the following tasks.

Tasks

Run the lab\_19\_01.sql script in the labs folder to create the SAL\_HISTORY, MGR\_HISTORY and SPECIAL\_SAL tables.

Display the structure of the SAL\_HISTORY table.

Display the structure of the MGR\_HISTORY table.

Display the structure of the SPECIAL\_SAL table.

Write a query to do the following:

Retrieve details such as the employee ID, hire date, salary, and manager ID of those employees whose employee ID is less than 125 from the EMPLOYEES table.

If the salary is more than $20,000, insert details such as the employee ID and salary into the SPECIAL\_SAL table.

If the salary is less than $20,000:

Insert details such as the employee ID, hire date, and salary into the

SAL\_HISTORY table

Insert details such as the employee ID, manager ID, and salary into the

MGR\_HISTORY table

Display the records from the SPECIAL\_SAL table.

Display the records from the SAL\_HISTORY table.

**…**

Display the records from the MGR\_HISTORY table.

**…**

Run the lab\_19\_06\_a.sql script in the lab folder to create the SALES\_WEEK\_DATA

table.

Run the lab\_19\_06\_b.sql script in the lab folder to insert records into the

SALES\_WEEK\_DATA table.

Display the structure of the SALES\_WEEK\_DATA table.

Display the records from the SALES\_WEEK\_DATA table.

Run the lab\_19\_06\_e.sql script in the lab folder to create the EMP\_SALES\_INFO

table.

Display the structure of the EMP\_SALES\_INFO table.

Write a query to do the following:

Retrieve details such as employee ID, week ID, sales quantity on Monday, sales quantity on Tuesday, sales quantity on Wednesday, sales quantity on Thursday, and sales quantity on Friday from the SALES\_WEEK\_DATA table.

Build a transformation such that each record retrieved from the SALES\_WEEK\_DATA

table is converted into multiple records for the EMP\_SALES\_INFO table.

**Hint:** Use a pivoting INSERT statement.

Display the records from the EMP\_SALES\_INFO table.

You have the data of past employees stored in a flat file called emp.data. You want to store the names and email IDs of all employees, past and present, in a table. To do this, first create an external table called EMP\_DATA using the emp.dat source file in the emp\_dir directory. Use the lab\_19\_07.sql script to do this.

Run the lab\_19\_08.sql script to create the EMP\_HIST table.

Increase the size of the email column to 45.

Merge the data in the EMP\_DATA table created in the last lab into the data in the EMP\_HIST table. Assume that the data in the external EMP\_DATA table is the most upto-date. If a row in the EMP\_DATA table matches the EMP\_HIST table, update the email column of the EMP\_HIST table to match the EMP\_DATA table row. If a row in the EMP\_DATA table does not match, insert it into the EMP\_HIST table. Rows are considered matching when the employee’s first and last names are identical.

Retrieve the rows from EMP\_HIST after the merge.

**...**

Create the EMP2 table based on the following table instance chart. Enter the syntax in the SQL Worksheet. Then execute the statement to create the table. Confirm that the table is created.

Drop the EMP2 table.

Query the recycle bin to see whether the table is present.

Restore the EMP2 table to a state before the DROP statement.

Create the EMP3 table using the lab\_19\_13.sql script. In the EMP3 table, change the department for Kochhar to 60 and commit your change. Next, change the department for Kochhar to 50 and commit your change. Track the changes to Kochhar using the Row Versions feature.

Drop the EMP2 and EMP3 tables so that they cannot be restored. Check in the recycle bin.

Solution 19-1: Manipulating Data

Solution

Run the lab\_19\_01.sql script in the lab folder to create the SAL\_HISTORY, MGR\_HISTORY and SPECIAL\_SAL tables.

Display the structure of the SAL\_HISTORY table.

Display the structure of the MGR\_HISTORY table.

Display the structure of the SPECIAL\_SAL table.

Write a query to do the following:

Retrieve details such as the employee ID, hire date, salary, and manager ID of those employees whose employee ID is less than 125 from the EMPLOYEES table.

If the salary is more than $20,000, insert details such as the employee ID and salary into the SPECIAL\_SAL table.

If the salary is less than $20,000:

Insert details such as the employee ID, hire date, and salary into the

SAL\_HISTORY table

Insert details such as the employee ID, manager ID, and salary into the

MGR\_HISTORY table

Display the records from the SPECIAL\_SAL table.

Display the records from the SAL\_HISTORY table.

Display the records from the MGR\_HISTORY table.

Run the lab\_19\_06\_a.sql script in the lab folder to create the SALES\_WEEK\_DATA

table.

Run the lab\_19\_06\_b.sql script in the lab folder to insert records into the

SALES\_WEEK\_DATA table.

Display the structure of the SALES\_WEEK\_DATA table.

Display the records from the SALES\_WEEK\_DATA table.

Run the lab\_19\_06\_e.sql script in the lab folder to create the EMP\_SALES\_INFO

table.

Display the structure of the EMP\_SALES\_INFO table.

Write a query to do the following:

Retrieve details such as the employee ID, week ID, sales quantity on Monday, sales quantity on Tuesday, sales quantity on Wednesday, sales quantity on Thursday, and sales quantity on Friday from the SALES\_WEEK\_DATA table.

Build a transformation such that each record retrieved from the SALES\_WEEK\_DATA

table is converted into multiple records for the EMP\_SALES\_INFO table.

**Hint:** Use a pivoting INSERT statement.

Display the records from the SALES\_INFO table.

You have the data of past employees stored in a flat file called emp.data. You want to store the names and email IDs of all employees past and present in a table. To do this, first create an external table called EMP\_DATA by using the emp.dat source file in the emp\_dir directory. You can use the script in lab\_19\_07.sql to do this.

Run the lab\_19\_08.sql script to create the EMP\_HIST table.

Increase the size of the email column to 45.

Merge the data in the EMP\_DATA table created in the last lab into the data in the EMP\_HIST table. Assume that the data in the external EMP\_DATA table is the most upto-date. If a row in the EMP\_DATA table matches the EMP\_HIST table, update the email column of the EMP\_HIST table to match the EMP\_DATA table row. If a row in the

EMP\_DATA table does not match, insert it into the EMP\_HIST table. Rows are considered matching when the employee’s first and last names are identical.

Retrieve the rows from EMP\_HIST after the merge.

Create the EMP2 table based on the following table instance chart. Enter the syntax in the SQL Worksheet. Then execute the statement to create the table. Confirm that the table is created.

Drop the EMP2 table.

Query the recycle bin to see whether the table is present.

Restore the EMP2 table to a state before the DROP statement.

Create the EMP3 table by using the lab\_19\_13.sql script. In the EMP3 table, change the department for Kochhar to 60 and commit your change. Next, change the department for Kochhar to 50 and commit your change. Track the changes to Kochhar using the Row Versions feature.

Drop the EMP2 and EMP3 tables, so that they cannot be restored. Check in the recycle bin.