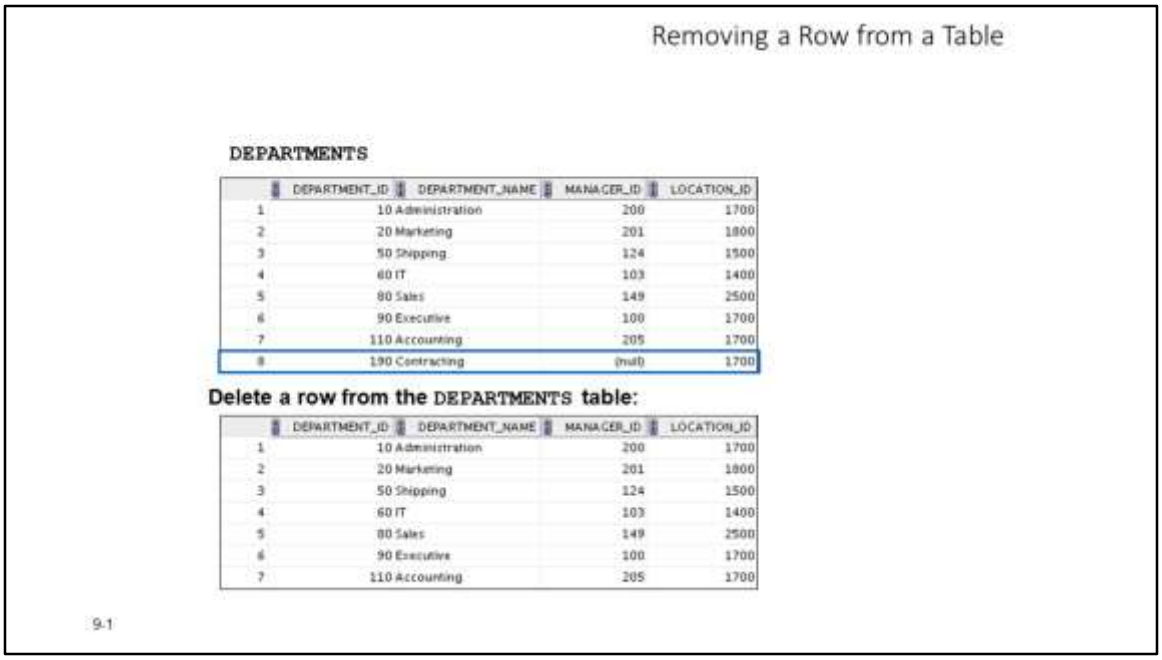
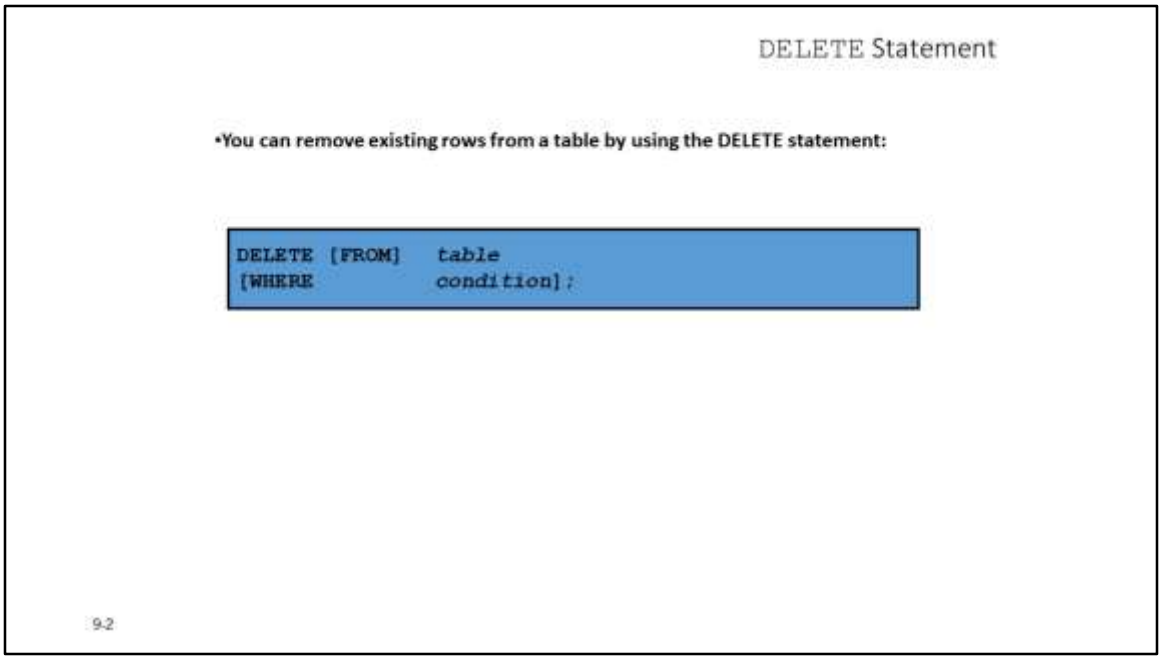


Removing a Row from a Table

The Contracting department has been removed from the `DEPARTMENTS` table (assuming no constraints on the `DEPARTMENTS` table are violated), as shown by the graphic in the slide.



DELETE Statement Syntax

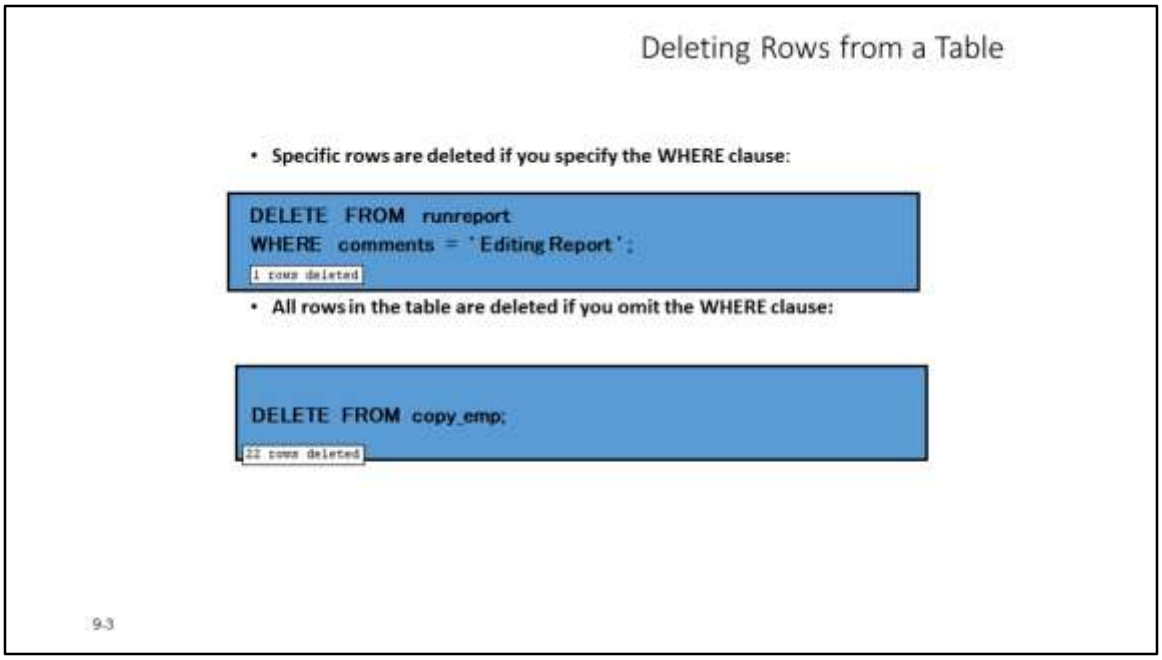
You can remove existing rows from a table by using the DELETE statement.

In the syntax:

table	Is the name of the table
condition	Identifies the rows to be deleted, and is composed of column names, expressions, constants, subqueries, and comparison operators

Note: If no rows are deleted, the message "0 rows deleted" is returned (on the Script Output tab in SQL Developer)

For more information, see the section on "DELETE" in *Oracle Database SQL Language Reference* for 10*g* or 11*g* database.



Deleting Rows from a Table

You can delete specific rows by specifying the `WHERE` clause in the `DELETE` statement. The first example in the slide deletes the row from the `RUNREPORT` table where the comment is 'Editing Report'. You can confirm the delete operation by displaying the deleted rows using the `SELECT` statement.

```
SELECT  *
FROM    runreport
WHERE   comments = 'Editing Report' ;
```

However, if you omit the `WHERE` clause, all rows in the table are deleted. The second example in the slide deletes all rows from the `COPY_EMP` table, because no `WHERE` clause was specified.

```
0 rows selected
```

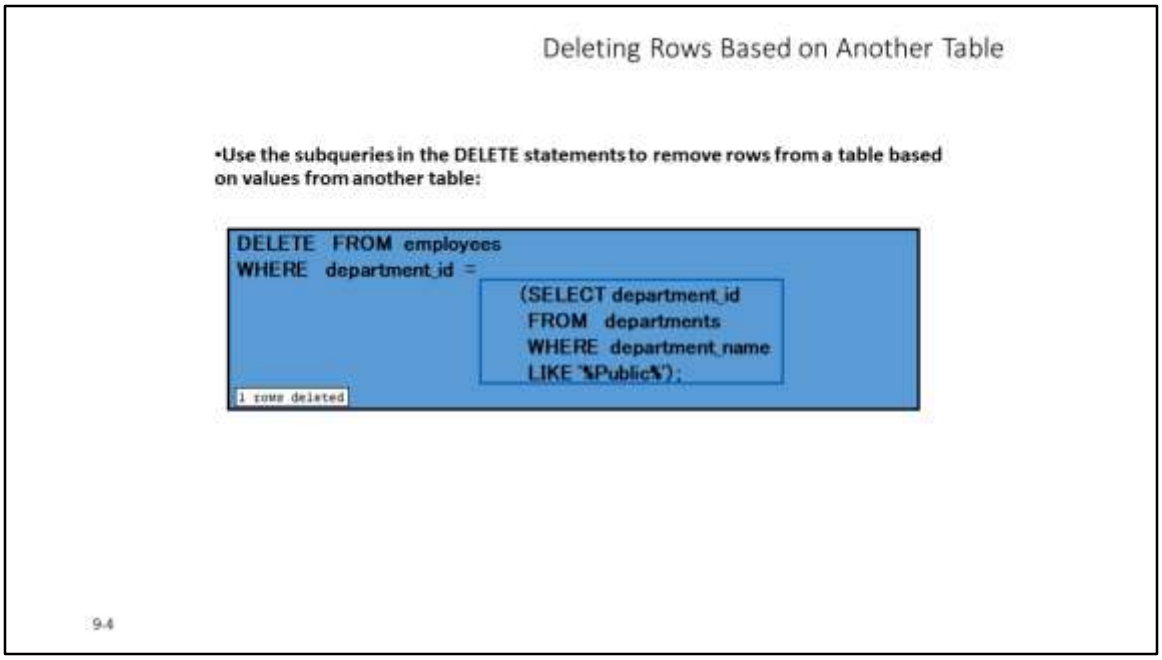
Remove rows identified in the `WHERE` clause.

```
DELETE FROM  employees WHERE employee_id = 114;
```

```
DELETE FROM  departments WHERE department_id IN (30, 40);
```

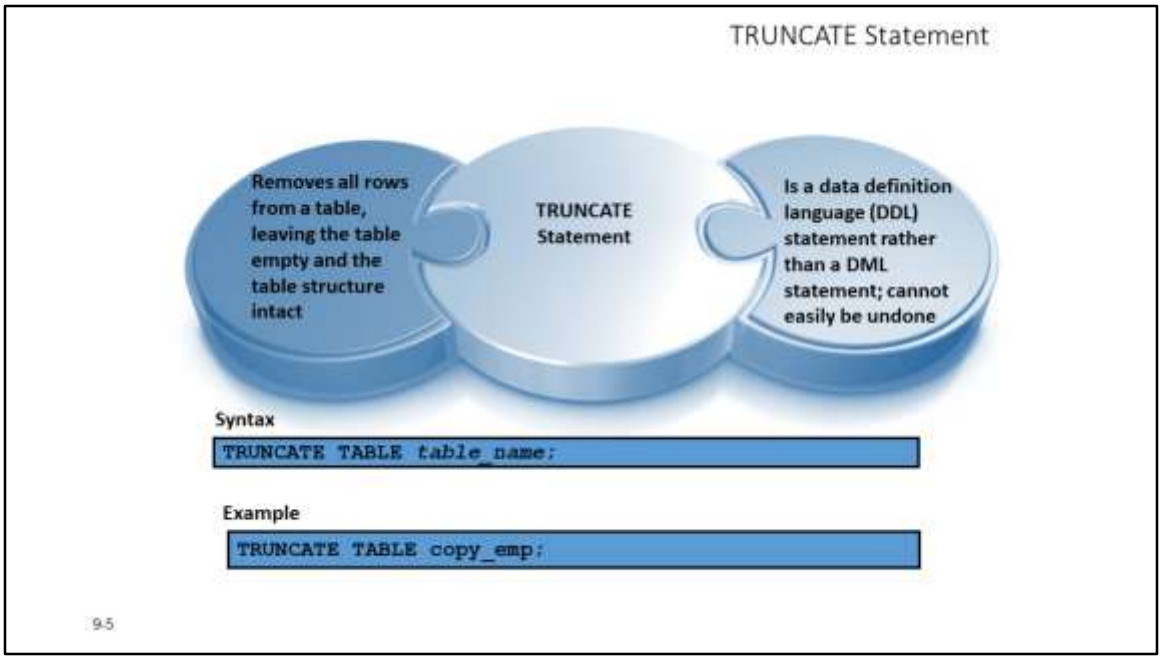
```
1 rows deleted
```

```
2 rows deleted
```



Deleting Rows Based on Another Table

You can use the subqueries to delete rows from a table based on values from another table. The example in the slide deletes all the employees in a department, where the department name contains the string `Public`.
The subquery searches the `DEPARTMENTS` table to find the department number based on the department name containing the string `Public`. The subquery then feeds the department number to the main query, which deletes rows of data from the `EMPLOYEES` table based on this department number.



TRUNCATE Statement

A more efficient method of emptying a table is by using the TRUNCATE statement.

You can use the TRUNCATE statement to quickly remove all rows from a table or cluster. Removing rows with the TRUNCATE statement is faster than removing them with the DELETE statement for the following reasons:

- The TRUNCATE statement is a data definition language (DDL) statement and generates no rollback information. Rollback information is covered later in this lesson.
- Truncating a table does not fire the delete triggers of the table.

If the table is the parent of a referential integrity constraint, you cannot truncate the table. You need to disable the constraint before issuing the TRUNCATE statement. Disabling constraints is covered in the lesson titled "Using DDL Statements to Create and Manage Tables."