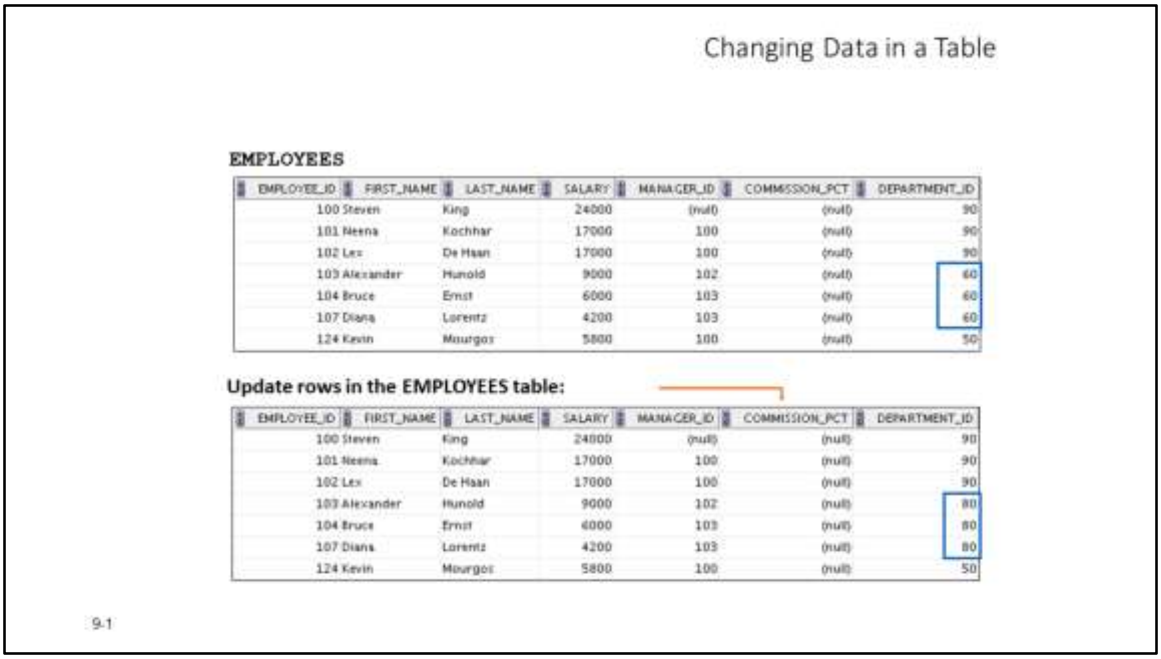


Changing Data in a Table

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	MANAGER_ID	COMMISSION_PCT	DEPARTMENT_ID
100	Steven	King	24000	(null)	(null)	90
101	Neena	Kochhar	17000	100	(null)	90
102	Lex	De Haan	17000	100	(null)	90
103	Alexander	Hunold	9000	102	(null)	60
104	Bruce	Ernst	6000	103	(null)	60
107	Diana	Lorentz	4200	103	(null)	60
124	Kevin	Mourgos	5800	100	(null)	50

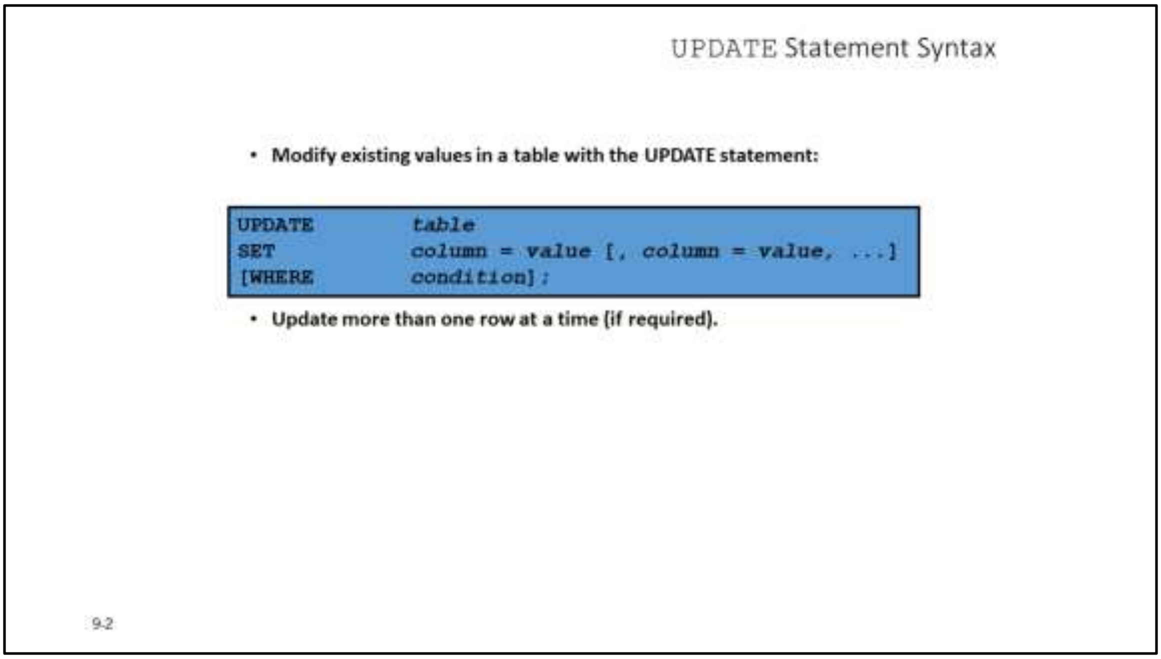
Update rows in the EMPLOYEES table:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	MANAGER_ID	COMMISSION_PCT	DEPARTMENT_ID
100	Steven	King	24000	(null)	(null)	90
101	Neena	Kochhar	17000	100	(null)	90
102	Lex	De Haan	17000	100	(null)	90
103	Alexander	Hunold	9000	102	(null)	80
104	Bruce	Ernst	6000	103	(null)	80
107	Diana	Lorentz	4200	103	(null)	80
124	Kevin	Mourgos	5800	100	(null)	50

9-1

Changing Data in a Table

The slide illustrates changing the department number for employees in department 60 to
department 80.



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).

9-2

UPDATE Statement Syntax

You can modify the existing values in a table by using the UPDATE statement.

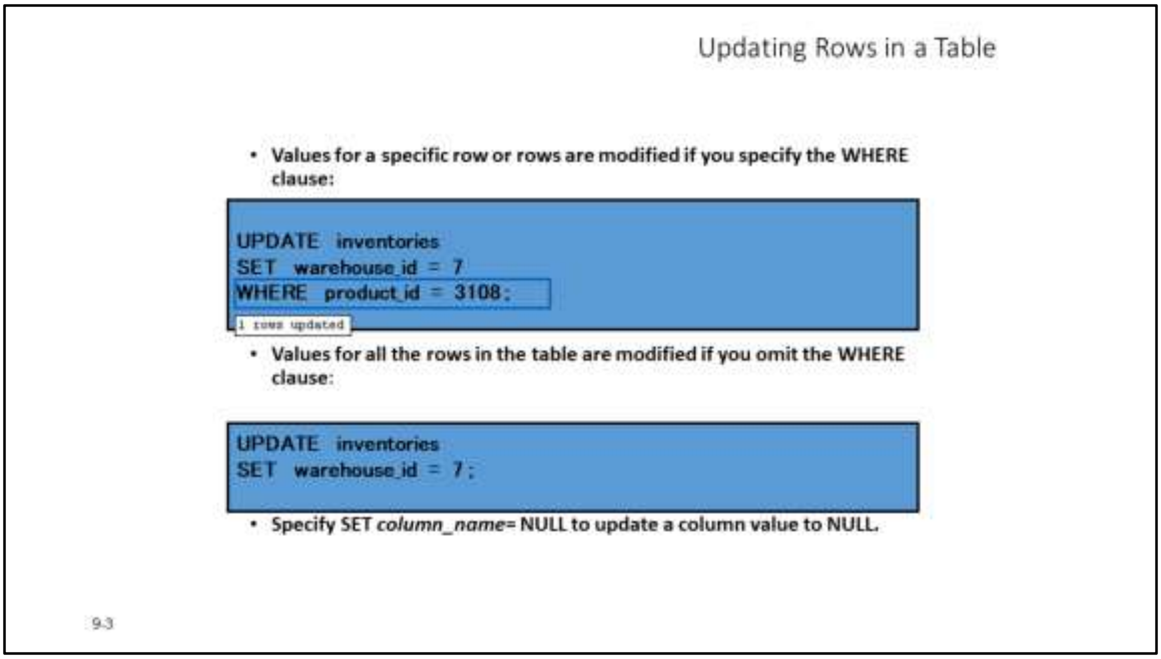
In the syntax:

table	Is the name of the table
column	Is the name of the column in the table to populate
value	Is the corresponding value or subquery for the column
condition	Identifies the rows to be updated and is composed of column names, expressions, constants, subqueries, and comparison operators

Confirm the update operation by querying the table to display the updated rows.

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

Note: In general, use the primary key column in the WHERE clause to identify a single row for update. Using other columns can unexpectedly cause several rows to be updated. For example, identifying a single row in the EMPLOYEES table by name is dangerous, because more than one employee may have the same name.



Updating Rows in a Table

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

```
UPDATE  inventories
SET  warehouse_id = 7
WHERE  product_id = 3108;
```

```
1 rows updated
```

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

```
UPDATE  inventories
SET  warehouse_id = 7;
```

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

9-3

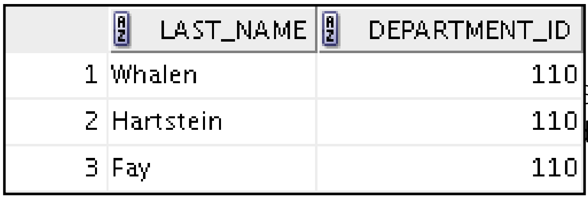
Updating Rows in a Table

The `UPDATE` statement modifies the values of a specific row or rows if the `WHERE` clause is specified. The example in the slide shows the transfer of product 3108 to warehouse 7.

If you omit the `WHERE` clause, values for all the rows in the table are modified.

Examine the updated rows in the `INVENTORIES` table.

```
SELECT *
FROM   inventories;
```



	LAST_NAME	DEPARTMENT_ID
1	Whalen	110
2	Hartstein	110
3	Fay	110

▪▪▪

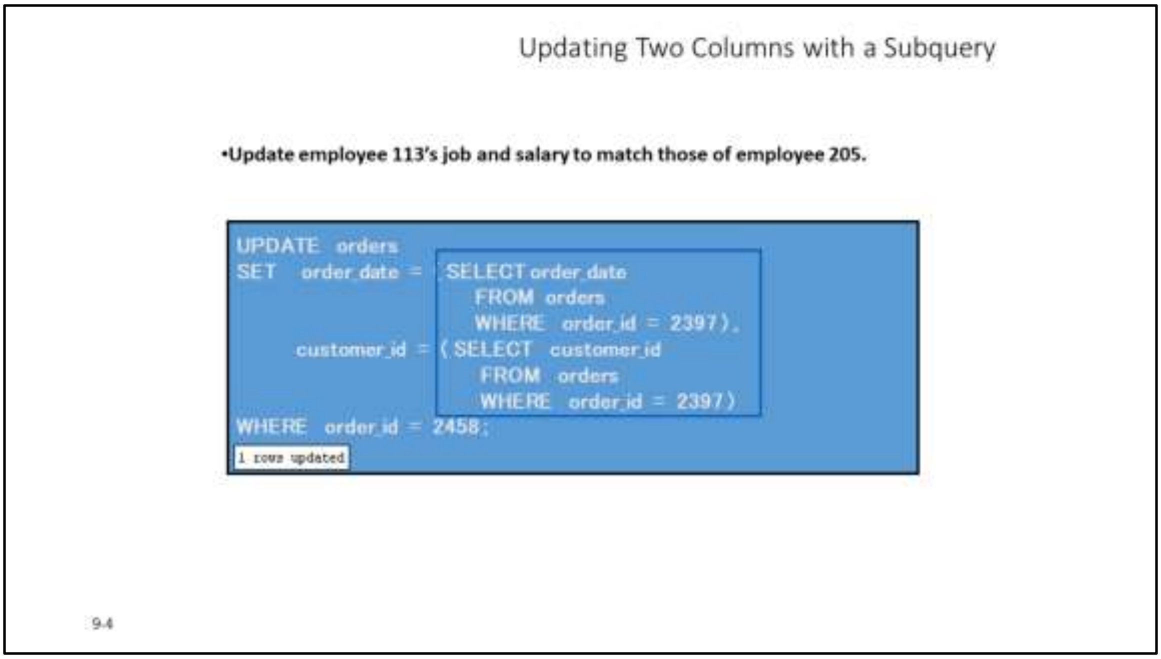
SA_REP has now changed his job to an

ls to be updated and the commission

```
UPDATE employees
SET job_id = 'IT_PROG', commission_pct = NULL
```

```
       WHERE employee_id = 114;
```

Note: The `COPY_EMP` table has the same data as the `EMPLOYEES` table.



Updating Two Columns with a Subquery

- Update employee 113's job and salary to match those of employee 205.

```
UPDATE  orders
SET   order_date =  (SELECT order_date
                       FROM  orders
                       WHERE  order_id = 2397),
      customer_id = (SELECT  customer_id
                       FROM  orders
                       WHERE  order_id = 2397)
WHERE  order_id = 2458;
```

```
1 rows updated
```

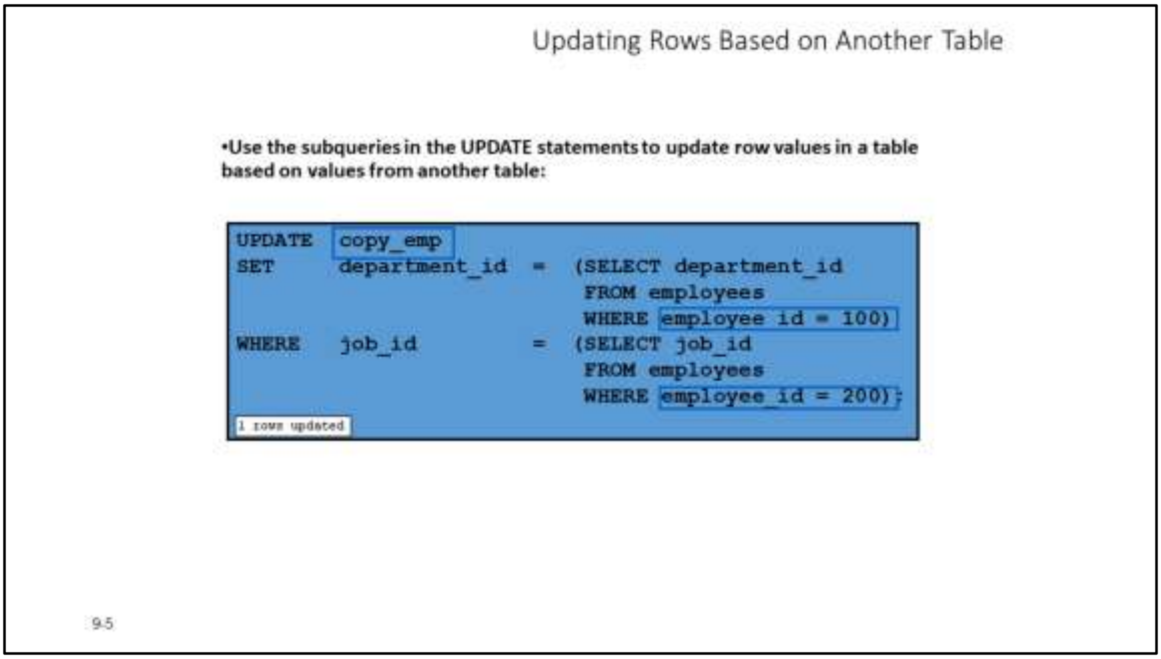
Updating Two Columns with a Subquery

You can update multiple columns in the SET clause of an UPDATE statement by writing multiple subqueries. The syntax is as follows:

```
    UPDATE table
SET      column  =
                                   (SELECT         column
                                    FROM table
                                    WHERE condition)
       [ ,
         column  =
                                   (SELECT         column
                                    FROM table
                                    WHERE condition)]
 [WHERE  condition ]       ;
```

The example in the slide can also be written as follows:

```
UPDATE orders
SET (order_date, customer_id)  = (SELECT  order_date,
  customer_id
                     FROM    orders
                     WHERE   order_id = 2397)
WHERE    order_id    =  2458;
```



Updating Rows Based on Another Table

- Use the subqueries in the UPDATE statements to update row values in a table based on values from another table:

```
UPDATE   copy_emp
SET      department_id  =  (SELECT department_id
                            FROM employees
                            WHERE employee_id = 100)
WHERE    job_id         =  (SELECT job_id
                            FROM employees
                            WHERE employee_id = 200);
```

```
1 rows updated
```

9-5

Updating Rows Based on Another Table

You can use the subqueries in the `UPDATE` statements to update values in a table. The example in the slide updates the `COPY_EMP` table based on the values from the `EMPLOYEES` table. It changes the department number of all employees with employee 200’s job ID to employee 100’s current department number.