# Oracle / PLSQL

**Oracle Basics**

**ALIASES**

column\_name AS alias\_name

table\_name alias\_name

ex:- first\_name || last\_name AS "CONTACT NAME"

products p

ex: -

SELECT p.product\_id, p.product\_name, categories.category\_name  
FROM products p  
INNER JOIN categories  
ON p.category\_id = categories.category\_id

ORDER BY p.product\_name ASC, categories.category\_name ASC;

# AND Condition

WHERE condition1

AND condition2

...

AND condition\_n;

Ex: -

UPDATE suppliers

SET supplier\_name = 'Apple'

WHERE supplier\_name = 'RIM'

AND offices = 8;

# Combining the AND and OR Conditions

## Note

* The Oracle AND & OR conditions allow you to test multiple conditions.
* Don't forget the order of operation parentheses!

SELECT supplier\_id

FROM suppliers

WHERE (supplier\_name = 'IBM')

OR (supplier\_name = 'Apple' AND state = 'Florida')

OR (supplier\_name = 'Best Buy' AND status = 'Active' AND state = 'California');

DELETE FROM suppliers

WHERE state = 'Florida'

AND (product = 'PC computers' OR supplier\_name = 'Dell');

# BETWEEN Condition

expression BETWEEN value1 AND value2;

ex: -

SELECT \*

FROM customers

WHERE customer\_id BETWEEN 4000 AND 4999;

==

SELECT \*

FROM customers

WHERE customer\_id >= 4000

AND customer\_id <= 4999;

SELECT \*

FROM order\_details

WHERE order\_date BETWEEN TO\_DATE ('2014/02/01', 'yyyy/mm/dd')

AND TO\_DATE ('2014/02/28', 'yyyy/mm/dd');

**Using NOT Operator**

SELECT \*

FROM customers

WHERE customer\_id NOT BETWEEN 3000 AND 3500;

==

SELECT \*

FROM customers

WHERE customer\_id < 3000

OR customer\_id > 3500;

# Comparison Operators



## Equality Operator

SELECT \*

FROM customers

WHERE last\_name = 'Anderson';

## Inequality Operator

In Oracle/PLSQL, you can use the <> or != operators to test for inequality in a query.

SELECT \*

FROM customers

WHERE last\_name <> 'Anderson';

SELECT \*

FROM customers

WHERE last\_name != 'Anderson';

## Greater Than Operator

SELECT \*

FROM suppliers

WHERE supplier\_id > 1000;

## Greater Than or Equal Operator

SELECT \*

FROM suppliers

WHERE supplier\_id >= 1000;

## Less Than Operator

SELECT \*

FROM employees

WHERE employee\_id < 99;

## Less Than or Equal Operator

SELECT \*

FROM employees

WHERE employee\_id <= 99;

## Advanced Operators

For the more advanced comparison operators, we've written specific tutorials to discuss each one individually. These topics will be covered later, or you can jump to one of these tutorials now.

* [IN ( )](https://www.techonthenet.com/oracle/in.php)
* [NOT](https://www.techonthenet.com/oracle/not.php)
* [BETWEEN](https://www.techonthenet.com/oracle/between.php)
* [IS NULL](https://www.techonthenet.com/oracle/isnull.php)
* [IS NOT NULL](https://www.techonthenet.com/oracle/isnotnull.php)
* [LIKE](https://www.techonthenet.com/oracle/like.php)
* [REGEXP\_LIKE](https://www.techonthenet.com/oracle/regexp_like.php)
* [EXISTS](https://www.techonthenet.com/oracle/exists.php)

# DELETE Statement

The Oracle DELETE statement is used to delete a single record or multiple records from a table in Oracle.

DELETE FROM table

[WHERE conditions];

## Note

* You do not need to list fields in the Oracle DELETE statement since you are deleting the entire row from the table.

DELETE FROM customers

WHERE last\_name = 'Smith';

DELETE FROM customers

WHERE last\_name = 'Anderson'

AND customer\_id > 25;

## Using EXISTS Clause

You may wish to delete records in one table based on values in another table. Since you can't list more than one table in the Oracle FROM clause when you are performing a delete, you can use the [Oracle EXISTS clause](https://www.techonthenet.com/oracle/exists.php).

DELETE FROM suppliers

WHERE EXISTS

( SELECT customers.customer\_name

FROM customers

WHERE customers.customer\_id = suppliers.supplier\_id

AND customer\_id > 25 );

# DISTINCT Clause

The Oracle DISTINCT clause is used to remove duplicates from the result set. The DISTINCT clause can only be used with [SELECT statements](https://www.techonthenet.com/oracle/select.php).

SELECT DISTINCT expressions

FROM tables

[WHERE conditions];

## Note

* When only one expression is provided in the DISTINCT clause, the query will return the unique values for that expression.
* When more than one expression is provided in the DISTINCT clause, the query will retrieve unique combinations for the expressions listed.
* In Oracle, the DISTINCT clause doesn't ignore NULL values. So when using the DISTINCT clause in your SQL statement, your result set will include NULL as a distinct value.
* SELECT DISTINCT state
* FROM customers
* WHERE last\_name = 'Smith';
* SELECT DISTINCT city, state
* FROM customers
* WHERE total\_orders > 10
* ORDER BY city;

# EXISTS Condition

The Oracle EXISTS condition is used in combination with a subquery and is considered "to be met" if the subquery returns at least one row. It can be used in a [SELECT](https://www.techonthenet.com/oracle/select.php),[INSERT](https://www.techonthenet.com/oracle/insert.php),[UPDATE](https://www.techonthenet.com/oracle/update.php), or[DELETE](https://www.techonthenet.com/oracle/delete.php) statement.

WHERE EXISTS ( subquery );

**subquery**

The subquery is a SELECT statement. If the subquery returns at least one record in its result set, the EXISTS clause will evaluate to true and the EXISTS condition will be met. If the subquery does not return any records, the EXISTS clause will evaluate to false and the EXISTS condition will not be met.

## Note

* Oracle SQL statements that use the Oracle EXISTS condition are very inefficient since the sub-query is RE-RUN for EVERY row in the outer query's table. There are more efficient ways to write most queries, that do not use the EXISTS condition.
* SELECT \*
* FROM customers
* WHERE EXISTS (SELECT \*
* FROM order\_details
* WHERE customers.customer\_id = order\_details.customer\_id);

## With SELECT Statement using NOT EXISTS

SELECT \*

FROM customers

WHERE NOT EXISTS (SELECT \*

FROM order\_details

WHERE customers.customer\_id = order\_details.customer\_id);

# FROM Clause

The Oracle/PLSQL FROM clause is used to list the tables and any join information required for the Oracle query.

SELECT \*

FROM homes

WHERE bathrooms >= 2

## ORDER BY home\_type ASC; Two tables with INNER JOIN

SELECT homes.home\_id, customers.last\_name, customers.first\_name

FROM customers

INNER JOIN homes

ON customers.customer\_id = homes.customer\_id

ORDER BY home\_id;

# GROUP BY Clause

The Oracle GROUP BY clause is used in a[SELECT statement](https://www.techonthenet.com/oracle/select.php) to collect data across multiple records and group the results by one or more columns.

SELECT expression1, expression2, ... expression\_n,

aggregate\_function (aggregate\_expression)

FROM tables

[WHERE conditions]

GROUP BY expression1, expression2, ... expression\_n;

EX: -

SELECT product, SUM(sale) AS "Total sales"

FROM order\_details

GROUP BY product;