

# Oracle WHERE Clause

The **WHERE** clause specifies a search condition for rows returned by the **SELECT** statement.

```
SELECT
    select_list
FROM
    table_name
WHERE
    search_condition
ORDER BY
    sort_expression;
```

PRODUCTS
* PRODUCT_ID
PRODUCT_NAME
DESCRIPTION
STANDARD_COST
LIST_PRICE
CATEGORY_ID

## A) Selecting rows by using a simple equality operator

The following example returns only products whose names are **'Kingston'** :

```
SELECT
    product_name,
    description,
    list_price,
    category_id
FROM
    products
WHERE
    product_name = 'Kingston';
```

## B) Select rows using comparison operator

Operator	Description
=	Equality

Operator	Description
!=, <>	Inequality
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
IN	Equal to any value in a list of values
ANY/ SOME / ALL	Compare a value to a list or subquery. It must be preceded by another operator such as =, >, <.
NOT IN	Not equal to any value in a list of values
[NOT] BETWEEN <i>n</i> and <i>m</i>	Equivalent to [Not] >= <i>n</i> and <= <i>y</i> .
[NOT] EXISTS	Return true if subquery returns at least one row
IS [NOT] NULL	NULL test

to get products whose list prices are greater than 500,

```
SELECT
    product_name,
    list_price
FROM
    products
WHERE
    list_price > 500;
```

### C) Select rows that meet some conditions

to get all motherboards that belong to the category id 1 and have list prices greater than 500,

```
SELECT
    product_name,
    list_price
FROM
    products
WHERE
    list_price > 500
    AND category_id = 4;
```

### D) Selecting rows that have a value between two values

To find rows that have a value between two values, you use the **BETWEEN** operator in the **WHERE** clause.

```
SELECT
    product_name,
    list_price
FROM
    products
WHERE
    list_price BETWEEN 650 AND 680
ORDER BY
    list_price;
```

list\_price BETWEEN 650 AND 680

list\_price >= 650 AND list\_price <= 680

### E) Selecting rows that are in a list of values

To query rows that are in a list of values, you use the **IN** operator

```
SELECT
    product_name,
    category_id
FROM
    products
WHERE
    category_id IN(1, 4)
ORDER BY
    product_name;
```

category\_id IN (1, 4)

category\_id = 1 OR category\_id = 4

### F) Selecting rows which contain value as a part of a string

etrieves product whose name starts with **Asus**

```
SELECT
    product_name,
    list_price
FROM
    products
WHERE
    product_name LIKE 'Asus%'
ORDER BY
    list_price;
```