

# Lexical Units in a PL/SQL Block

## Lexical units:

- Are building blocks of any PL/SQL block
- Are sequences of characters including letters, numerals, tabs, spaces, returns, and symbols
- Can be classified as:
  - Identifiers: `v_fname`, `c_percent`
  - Delimiters: `;`, `,`, `+`, `-`
  - Literals: `John`, `428`, `True`
  - Comments: `--`, `/* */`

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

## Lexical Units in a PL/SQL Block

Lexical units include letters, numerals, special characters, tabs, spaces, returns, and symbols.

**Identifiers:** Identifiers are the names given to PL/SQL objects. You learned to identify valid and invalid identifiers. Recall that keywords cannot be used as identifiers.

**Quoted identifiers:**

Make identifiers case-sensitive.

Include characters such as spaces.

Use reserved words.

**Examples:**

`"begin date" DATE;`

`"end date" DATE;`

`"exception thrown" BOOLEAN DEFAULT TRUE;`

All subsequent usage of these variables should have double quotation marks. However, use of quoted identifiers is not recommended.

**Delimiters:** Delimiters are symbols that have special meaning. You already learned that the semicolon (`;`) is used to terminate a SQL or

3 - 1

PL/SQL statement. Therefore, ; is an example of a delimiter.

For more information, refer to the PL/SQL User's Guide and Reference.

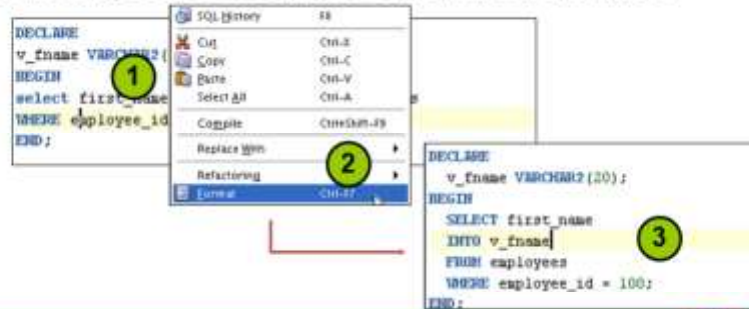
# PL/SQL Block Syntax and Guidelines

## – Using Literals

- Character and date literals must be enclosed in single quotation marks.
- Numbers can be simple values or in scientific notation.

```
v_name := 'Henderson';
```

## – Formatting Code: Statements can span several lines.



Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

#### PL/SQL Block Syntax and Guidelines

##### Using Literals

A literal is an explicit numeric, character string, date, or Boolean value that is not represented by an identifier.

Character literals include all printable characters in the PL/SQL character set: letters, numerals, spaces, and special symbols.

Numeric literals can be represented either by a simple value (for example, `-32.5`) or in scientific notation (for example, `2E5` means  $2 * 10^5 = 200,000$ ).

##### Formatting Code

In a PL/SQL block, a SQL statement can span several lines (as shown in example 3 in the slide).

You can format an unformatted SQL statement (as shown in example 1 in the slide) by using the SQL Worksheet shortcut menu. Right-click the active SQL Worksheet and, in the shortcut menu that appears, select the Format option (as shown in example 2).

**Note:** You can also use the shortcut key combination of Ctrl + F7 to format your code.

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

## Commenting Code

- Prefix single-line comments with two hyphens (--).
- Place a block comment between the symbols /\* and \*/.

Example:

```
DECLARE
+++
v_annual_sal NUMBER(9,2);
BEGIN
/* Compute the annual salary based on the
   monthly salary input from the user */
v_annual_sal := monthly_sal * 12;
-- The following line displays the annual salary
DBMS_OUTPUT.PUT_LINE (v_annual_sal);
END;
/
```

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

### Commenting Code

You should comment code to document each phase and to assist debugging.

In PL/SQL code:

A single-line comment is commonly prefixed with two hyphens (--)

You can also enclose a comment between the symbols /\* and \*/ Note:

For multiline comments, you can either precede each comment line with two hyphens, or use the block comment format.

Comments are strictly informational and do not enforce any conditions or behavior on the logic or data. Well-placed comments are extremely valuable for code readability and future code maintenance.

3 - 4