



One of the problems faced by Developers when defining local variables is getting the datatype and size the same as the underlying column in the table, in addition, over the course of time a PL/SQL script can become invalid because local variables which had been created to take tablecolumn values have not been updated when the table structure has changed.

A method of avoiding this problem is to use the *%type* declaration where the variable is given the column as its size and type.

## Simple DML - Declaring Variables Using %Type

---



The %type clause refers the PLSQL to the column structure directly instead of a datatype being used, %type uses the name of table and its column within the declaration ...

For example, instead of the Developer creating a variable with a datatype ...

```
l_section_no number(8);
```

it becomes ...

```
l_section_no section_details.sd_section_number%type;
```

## Simple DML - Declaring Variables Using %Type

---



The %type names the table separated by a dot followed by the column name, by using this method the Developer not only shows other Developers the intended recipient of the local variables value but no longer needs to worry about compatibility, in addition %type can be used to 'copy' previously declared local variables ...

```
l_date common_time.day_of_event%type;
```

```
l_day l_date%type;
```

```
l_month l_date%type;
```