



PLS_Integer replaces the older and slower Binary_Integer, which has virtually the same characteristics as PLS_Integer.

Note that an error will occur if either of the above are in a calculation which reaches the limit of their size, even if the result is put into a Number datatype ...

declare

```
l_binary1 binary_integer := 2147483647;
```

```
l_result number;
```

begin

```
l_result := l_binary1 + 2;
```

```
ORA-01426: numeric overflow
```



The Natural datatype is a subset of PLS_Integer and will allow any values between 0 and 2147483647, in other words no negative numbers ...

declare

```
l_natural natural := -1;
```

Results in ...

```
ORA-06502: PL/SQL: numeric or value error
```



The Naturaln datatype is also subset of PLS_Integer and will allow any values between 0 and 2147483647, in other words no negative numbers but unlike the Natural datatype, Naturaln will not allow nulls ...

declare

```
l_naturaln naturaln := null;
```

Results in ...

```
ORA-06550: line 3, column 31:  
PLS-00382: expression is of wrong type
```



The Positive datatype is also subset of PLS_Integer and will allow any values between 1 and 2147483647, in other words no negative or zero numbers (nulls are acceptable) ...

declare

```
l_int positive := 0;
```

Results in ...

```
ORA-06502: PL/SQL: numeric or value error
```



The Positiven datatype is very similar to the Positive except that it has an additional 'constraint' of not accepting nulls ...

declare

```
l_positiven positiven := null;
```

Results in ...

```
ORA-06550: line 3, column 31:  
PLS-00382: expression is of wrong type
```



The Signtype datatype can only accept four values, -1, 0, 1 and null and can be used as an alternative for boolean datatypes, there is no Signtypen ...

declare

```
l_sign signtype := 1;  
l_sign1 signtype := -1;  
l_sign2 signtype := null;  
l_sign3 signtype := 0;  
l_sign4 signtype := 2;
```

The final declaration results in ...

```
ORA-06502: PL/SQL: numeric or value error
```