

Oracle INTERVAL

Summary: in this tutorial, we'll introduce you to the Oracle **INTERVAL** data types and show you how to handle intervals effectively.

Introduction to Oracle data type

Oracle provides you with two date time data types: **DATE** (<https://www.oracletutorial.com/oracle-basics/oracle-date/>) and **TIMESTAMP** (<https://www.oracletutorial.com/oracle-basics/oracle-timestamp/>) for storing point-in-time data. In addition, It provides the **INTERVAL** data type that allows you to store periods of time.

There are two types of **INTERVAL** :

- **INTERVAL YEAR TO MONTH** – stores intervals using of year and month.
- **INTERVAL DAY TO SECOND** – stores intervals using days, hours, minutes, and seconds including fractional seconds.

Oracle INTERVAL YEAR TO MONTH

The **INTERVAL YEAR TO MONTH** data type allows you to store a period of time using the **YEAR** and **MONTH** fields.

The following illustrates an **INTERVAL YEAR TO MONTH** :

```
INTERVAL YEAR [(year_precision)] TO MONTH
```

The **year_precision** represents the number of digits in the **YEAR** field. It ranges from 0 to 9.

The **year_precision** is optional. If you omit the **year_precision** argument, it defaults to 2. In other words, by default, you can store up to a period of 99 years and 11 months, which must be less than 100 years.

Oracle INTERVAL YEAR TO MONTH literals

To specify literal values for the `INTERVAL YEAR TO MONTH` data type, you use the following format:

```
INTERVAL 'year[-month]' leading (precision) TO trailing
```

Where `leading` and `trailing` can be `YEAR` or `MONTH`.

The following are arguments:

`'year[-month]'`

The `year` and `month` are integers for the leading and trailing fields of the interval. If `leading` is `YEAR` and `trailing` is `MONTH`, then the `month` field ranges from 0 to 11.

The `trailing` field must be less than the `leading` field. For example, `INTERVAL '1-2' MONTH TO YEAR` is invalid because trailing is `YEAR` which is greater than the leading field which is `MONTH`.

`precision`

is the maximum number of digits in the `leading` field. The precision ranges from 0 to 9 and its default value is 2.

The following table illustrates examples of `INTERVAL YEAR TO MONTH` literals:

INTERVAL YEAR TO MONTH Literals	Meaning
INTERVAL '120-3' YEAR(3) TO MONTH	An interval of 120 years, 3 months; Must specify the leading field precision YEAR(3) because the value of the leading field is greater than the default precision (2 digits).
INTERVAL '105' YEAR(3)	An interval of 105 years 0 months.
INTERVAL '500' MONTH(3)	An interval of 500 months.

INTERVAL YEAR TO MONTH Literals	Meaning
INTERVAL '9' YEAR	9 years, which is equivalent to INTERVAL '9-0' YEAR TO MONTH
INTERVAL '40' MONTH	40 months or 3 years 4 months, which is equivalent to INTERVAL '3-4' YEAR TO MONTH
INTERVAL '180' YEAR	Invalid interval because '180' has 3 digits which are greater than the default precision (2)

Oracle INTERVAL YEAR TO MONTH example

First, let's create a new table named `candidates` for the demonstration:

```
CREATE TABLE candidates (  
    candidate_id NUMBER,  
    first_name VARCHAR2(50) NOT NULL,  
    last_name VARCHAR2(50) NOT NULL,  
    job_title VARCHAR2(255) NOT NULL,  
    year_of_experience INTERVAL YEAR TO MONTH,  
    PRIMARY KEY (candidate_id)  
);
```

In this table, we have the `year_of_experience` column whose the data type is `INTERVAL YEAR TO MONTH`.

Second, insert data into the `candidates` table:

```
INSERT INTO candidates (  
    first_name,  
    last_name,  
    job_title,  
    year_of_experience
```

```

    )
VALUES (
    'Camila',
    'Kramer',
    'SCM Manager',
    INTERVAL '10-2' YEAR TO MONTH
);

```

In this statement, we inserted an interval literal of 10 years 2 months into the `year_of_experience` column.

```

INSERT INTO candidates (
    first_name,
    last_name,
    job_title,
    year_of_experience
)
VALUES (
    'Keila',
    'Doyle',
    'SCM Staff',
    INTERVAL '9' MONTH
);

```

In this statement, we inserted an interval literal 10 months into the `year_of_experience` column.

Third, query data from the `INTERVAL YEAR TO MONTH` column:

```
SELECT * FROM candidates;
```

	CANDIDATE_ID	FIRST_NAME	LAST_NAME	JOB_TITLE	YEAR_OF_EXPERIENCE
1	1	Camila	Kramer	SCM Manager	+10-02
2	2	Keila	Doyle	SCM Staff	+00-09

Oracle INTERVAL DAY TO SECOND data type

The **INTERVAL DAY TO SECOND** stores a period of time in terms of days, hours, minutes, and seconds.

The following shows the syntax of the **INTERVAL DAY TO SECOND** data type:

```
INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)]
```

In this syntax:

- **day_precision** is the number of digits in the **DAY** field. It ranges from 0 to 9. By default, its value is set to 2.
- **fractional_seconds_precision** is the number of digits in the fractional part of the **SECOND** field. It ranges from 0 through 9. If you omit the **fractional_seconds_precision**, it defaults to 6.

Oracle INTERVAL DAY TO SECOND literals

The literal form of **INTERVAL DAY TO SECOND** is as follows:

```
INTERVAL leading (leading_precision) to trailing(fractional_seconds_precision)
```

The following table shows some examples of **INTERVAL DAY TO SECOND** literals:

INTERVAL DAY TO SECOND Literals	Meaning
INTERVAL '11 10:09:08.555' DAY TO SECOND(3)	11 days, 10 hours, 09 minutes, 08 seconds, and 555 thousandths of a second.
INTERVAL '11 10:09' DAY TO MINUTE	11 days, 10 hours, and 09 minutes.
INTERVAL '100 10' DAY(3) TO HOUR	100 days 10 hours.
INTERVAL '999' DAY(3)	999 days.

INTERVAL DAY TO SECOND Literals	Meaning
INTERVAL '09:08:07.6666666' HOUR TO SECOND(7)	9 hours, 08 minutes, and 7.6666666 seconds.
INTERVAL '09:30' HOUR TO MINUTE	9 hours and 30 minutes.
INTERVAL '8' HOUR	8 hours.
INTERVAL '15:30' MINUTE TO SECOND	15 minutes 30 seconds.
INTERVAL '30' MINUTE	30 minutes.
INTERVAL '5' DAY	5 days.
INTERVAL '40' HOUR	40 hours.
INTERVAL '15' MINUTE	15 minutes.
INTERVAL '250' HOUR(3)	250 hours.
INTERVAL '15.6789' SECOND(2,3)	Rounded to 15.679 seconds. Because the precision is 3, the fractional second '6789' is rounded to '679'

In this tutorial, you have learned how to use the Oracle INTERVAL data type to store periods of time in the tables.