**Date and Time function**

1. age() function :- age function are used to find the age with use’s two timestamp . Age () function are two type of syntax .

* age(timestamp,timestamp),the first timestamp is current timestamp and second one is which timestamp to find age.
* Example :- postgres=# SELECT AGE(timestamp '2001-04-10', timestamp '1957-06-13');
* age
* -------------------------
* 43 years 9 mons 27 days
* (1 row)
* age(timestamp) : postgres=# select age(timestamp '1994-07-21');
* age
* -------------------------
* 26 years 9 mons 30 days
* (1 row)

2. Find current date/time .

* CURRENT\_DATE Delivers current date.

Example : - postgres=# select current\_date;

current\_date

--------------

2021-05-20

(1 row)

* CURRENT\_TIME Delivers values with time zone.

Example :- postgres=# select current\_time;

current\_time

-----------------------

16:55:30.246204+05:30

(1 row)

* CURRENT\_TIMESTAMP Delivers values with time zone.

Example :- postgres=# select current\_timestamp;

current\_timestamp

----------------------------------

2021-05-20 16:55:37.580135+05:30

(1 row)

* CURRENT\_TIME(precision) Optionally takes a precision parameter, which causes the result to be rounded to that many fractional digits in the seconds field.

Example :-postgres=# select current\_time(3);

current\_time

--------------------

17:06:02.607+05:30

(1 row)

* CURRENT\_TIMESTAMP(precision) Optionally takes a precision parameter, which causes the result to be rounded to that many fractional digits in the seconds field.

Example : - postgres=# select current\_timestamp(2);

current\_timestamp

------------------------------

2021-05-20 17:04:31.39+05:30

(1 row)

* LOCALTIME Delivers values without time zone.

Example : - postgres=# select localtime;

localtime

----------------

17:08:07.47499

(1 row)

* LOCALTIMESTAMP Delivers values without time zone.

Example : -postgres=# select localtimestamp;

localtimestamp

----------------------------

2021-05-20 17:09:32.690486

(1 row)

* LOCALTIME(precision) Optionally takes a precision parameter, which causes the result to be rounded to that many fractional digits in the seconds field.

Example : - postgres=# select localtime(1);

localtime

------------

17:08:16.1

(1 row)

* LOCALTIMESTAMP(precision) Optionally takes a precision parameter, which causes the result to be rounded to that many fractional digits in the seconds field.

Example :-postgres=# select localtimestamp(2);

localtimestamp

------------------------

2021-05-20 17:12:02.89

(1 row)

* now() show the current date ,time and timezone.

Example : - postgres=# select NOW();

now

----------------------------------

2021-05-20 17:15:12.093219+05:30

(1 row)

* timeofday() show day month time and year

Example : - postgres=# select timeofday();

timeofday

-------------------------------------

Thu May 20 17:15:31.199069 2021 IST

(1 row)

* transaction\_timestamp() show the current date ,time and timezone.
* postgres=# SELECT NOW(), TRANSACTION\_TIMESTAMP();
* now | transaction\_timestamp
* ----------------------------------+----------------------------------
* 2021-05-20 17:23:37.861598+05:30 | 2021-05-20 17:23:37.861598+05:30
* (1 row)
* statement\_timestamp()
* Example : - postgres=# SELECT statement\_timestamp();

statement\_timestamp

----------------------------------

2021-05-20 17:30:58.639136+05:30

(1 row)

* clock\_timestamp()
* postgres=# SELECT clock\_timestamp();
* clock\_timestamp
* ----------------------------------
* 2021-05-20 17:32:21.469996+05:30
* (1 row)

3. EXTRACT(field from timestamp), EXTRACT(field from interval)

* The PostgreSQL EXTRACT() function retrieves a field such as a year, month, and day from a date/time value.

**Syntax : -** The following illustrates the syntax of the EXTRACT() function:

EXTRACT(field FROM source)

Arguments

The PostgreSQL EXTRACT() function requires two arguments:

**1) field : -** The field argument specifies which field to extract from the date/time value. The following table illustrates the valid field values:

Field Value TIMESTAMP Interval

1. CENTURY The century The number of centuries
2. DAY The day of the month (1-31) The number of days
3. DECADE The decade that is the year divided by 10 Sames as TIMESTAMP
4. DOW The day of week Sunday (0) to Saturday (6) N/A
5. DOY The day of year that ranges from 1 to 366 N/A
6. EPOCH The number of seconds since 1970-01-01 The total number

00:00:00 UTC of seconds in the interval

1. HOUR The hour (0-23) The number of hours
2. ISODOW Day of week based on ISO 8601 N/A

Monday (1) to Sunday (7)

1. ISOYEAR ISO 8601 week number of year N/A
2. MICROSECONDS The seconds field, including fractional Sames as TIMESTAMP

parts, multiplied by 1000000

1. MILLENNIUM The millennium The number of millennium
2. MILLISECONDS The seconds field, including fractional Sames as TIMESTAMP

parts, multiplied by 1000

1. MINUTE The minute (0-59) The number of minutes
2. MONTH Month, 1-12 The number of months, modulo (0-11)
3. QUARTER Quarter of the year The number of quarters
4. SECOND The second The number of seconds
5. TIMEZONE The timezone offset from UTC, N/A

measured in seconds

1. TIMEZONE\_HOUR The hour component of the time zone offset N/A
2. TIMEZONE\_MINUTE The minute component of the time zone offset N/A
3. WEEK The number of the ISO 8601 week-numbering week of the year N/A
4. YEAR The year Sames as TIMESTAMP

**2) source**

The source is a value of type TIMESTAMP or INTERVAL. If you pass a DATE value, the function will cast it to a TIMESTAMP value.

**Return value :-** The EXTRACT() function returns a double precision value.

A) Epostgres=# select Extract (year from timestamp '2021-07-31');

date\_part

-----------

2018

(1 row)xtracting from a TIMESTAMP examples

postgres=# select extract(quarter from timestamp '2021-10-19');

date\_part

-----------

4

(1 row)

postgres=# select extract(quarter from timestamp '2021-04-19');

date\_part

-----------

2