

# Overview of basic arithmetic operators

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL



**Brian Piccolo**

Sr. Director, Digital Strategy

# Topics

- Overview of basic arithmetic operators
- The `CURRENT_DATE` , `CURRENT_TIMESTAMP` , `NOW()` functions
- The `AGE()` function
- The `EXTRACT()` , `DATE_PART()` , and `DATE_TRUNC()` functions

# Adding and subtracting date / time data

```
SELECT date '2005-09-11' - date '2005-09-10';
```

```
+-----+  
| integer |  
+-----+  
| 1       |  
+-----+
```

# Adding and subtracting date / time data

```
SELECT date '2005-09-11' + integer 3;
```

```
+-----+  
| interval |  
|-----|  
| 2005-09-14 |  
+-----+
```

# Adding and subtracting date / time data

```
SELECT date '2005-09-11 00:00:00' - date '2005-09-09 12:00:00';
```

```
+-----+  
| interval |  
|-----|  
| 1 day 12:00:00 |  
+-----+
```

# Calculating time periods with AGE

```
SELECT AGE(timestamp '2005-09-11 00:00:00', timestamp '2005-09-09 12:00:00');
```

```
+-----+
| interval |
|-----|
| 1 day 12:00:00 |
+-----+
```

# DVDs, really??

```
SELECT
```

```
    AGE(rental_date)
```

```
FROM rental;
```

```
+-----+
| age                |
+-----+
| 13 years 11 mons 12 days 01:06:30 |
| 13 years 11 mons 12 days 01:05:27 |
| 13 years 11 mons 12 days 00:56:21 |
+-----+
```

# Date / time arithmetic using INTERVALs

```
SELECT rental_date + INTERVAL '3 days' as expected_return
FROM rental;
```

```
+-----+
| expected_return |
|-----|
| 2005-05-27 22:53:30 |
+-----+
```



# Date / time arithmetic using INTERVALs

```
SELECT timestamp '2019-05-01' + 21 * INTERVAL '1 day';
```

```
+-----+
| timestamp without timezone |
|-----|
| 2019-05-22 00:00:00        |
+-----+
```

# Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

# Functions for retrieving current date/time

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL



**Brian Piccolo**

Sr. Director, Digital Strategy

# Retrieving the current timestamp

```
SELECT NOW();
```

```
+-----+
| now() |
|-----|
| 2019-04-19 02:51:18.448641+00 |
+-----+
```

# Retrieving the current timestamp

```
SELECT NOW()::timestamp;
```

```
+-----+  
| now() |  
+-----+  
| 2019-04-19 02:51:18.448641 |  
+-----+
```

# Retrieving the current timestamp

## PostgreSQL specific casting

```
SELECT NOW()::timestamp;
```

## CAST() function

```
SELECT CAST(NOW() as timestamp);
```

# Retrieving the current timestamp

```
SELECT CURRENT_TIMESTAMP;
```

```
+-----+
| current_timestamp |
|-----|
| 2019-04-19 02:51:18.448641+00 |
+-----+
```

# Retrieving the current timestamp

```
SELECT CURRENT_TIMESTAMP(2);
```

```
+-----+
| current_timestamp |
|-----|
| 2019-04-19 02:51:18.44+00 |
+-----+
```



# Current date and time

```
SELECT CURRENT_DATE;
```

```
+-----+  
| current_date |  
|-----|  
| 2019-04-19   |  
+-----+
```

# Current date and time

```
SELECT CURRENT_TIME;
```

```
+-----+
| current_time |
|-----|
| 04:06:30.929845+00:00 |
+-----+
```

# Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL

# Extracting and transforming date / time data

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL



**Brian Piccolo**

Sr. Director, Digital Strategy

# Extracting and transforming date and time data

Exploring the `EXTRACT()` , `DATE_PART()` and `DATE_TRUNC()` functions

- Transactional timestamp precision not useful for analysis

```
2005-05-13 08:53:53
```

- Often need to extract parts of timestamps

```
2005 or 5 or 2 or Friday
```

- Or convert / truncate timestamp precision to standardize

```
2005-05-13 00:00:00
```

# Extracting and transforming date / time data

- `EXTRACT( field FROM source )`

```
SELECT EXTRACT(quarter FROM timestamp '2005-01-24 05:12:00') AS quarter;
```

- `DATE_PART('field', source)`

```
SELECT DATE_PART('quarter' FROM timestamp '2005-01-24 05:12:00') AS quarter;
```

```
+-----+  
| quarter |  
+-----+  
| 1       |  
+-----+
```

# Extracting sub-fields from timestamp data

Transactional data from DVD Rentals *payment* table

```
SELECT * FROM payment;
```

```
+-----+-----+-----+-----+-----+-----+
| payment_id | customer_id | staff_id | rental_id | amount | payment_date |
|-----|-----|-----|-----|-----|-----|
| 1 | 1 | 1 | 76 | 2.99 | 2005-05-25 11:30:37 |
| 2 | 1 | 1 | 573 | 0.99 | 2005-05-28 10:35:23 |
| 3 | 1 | 1 | 1185 | 5.99 | 2005-06-15 0:54:12 |
+-----+-----+-----+-----+-----+-----+

```

# Extracting sub-fields from timestamp data

Data from *payment* table by year and quarter

```
SELECT
  EXTRACT(quarter FROM payment_date) AS quarter,
  EXTRACT(year FROM payment_date) AS year,
  SUM(amount) AS total_payments
FROM
  payment
GROUP BY 1, 2;
```

Results

```
+-----+
| quarter | year | total_payments |
|-----|-----|-----|
| 2       | 2005 | 14456.31      |
| 3       | 2005 | 52446.02      |
| 1       | 2006 | 514.18        |
+-----+
```



# Truncating timestamps using DATE\_TRUNC()

The `DATE_TRUNC()` function will truncate timestamp or interval data types.

- Truncate timestamp '2005-05-21 15:30:30' by year

```
SELECT DATE_TRUNC('year', TIMESTAMP '2005-05-21 15:30:30');
```

```
Result: 2005-01-01 00:00:00
```

- Truncate timestamp '2005-05-21 15:30:30' by month

```
SELECT DATE_TRUNC('month', TIMESTAMP '2005-05-21 15:30:30');
```

```
Result: 2005-05-01 00:00:00
```

# Let's practice!

FUNCTIONS FOR MANIPULATING DATA IN POSTGRESQL