```
********* QUERY **********
SELECT * FROM users;
**************************

 pk |  username  |    gecos     |    email     | user_on_line
----+------------+--------------+--------------+--------------
  2 | scotty     | scotty_gecos | scotty_email | NULL
  1 | myusername | mygecos      | myemail      | t
(2 rows)

********* QUERY **********
ALTER TABLE users
DROP COLUMN user_on_line;
**************************

ALTER TABLE
********* QUERY **********
ALTER TABLE users
ADD COLUMN user_on_line boolean;
**************************

ALTER TABLE
********* QUERY **********
UPDATE users
SET user_on_line = true
WHERE pk = 1;
**************************

UPDATE 1
********* QUERY **********
SELECT * FROM users WHERE user_on_line = true;
**************************

 pk |  username  |  gecos  |  email  | user_on_line
----+------------+---------+---------+--------------
  1 | myusername | mygecos | myemail | t
(1 row)

********* QUERY **********
SELECT * FROM users WHERE user_on_line IS NULL;
**************************

 pk | username |    gecos     |    email     | user_on_line
----+----------+--------------+--------------+--------------
  2 | scotty   | scotty_gecos | scotty_email | NULL
(1 row)

********* QUERY **********
SELECT 1.123456789::integer AS my_field;
**************************

 my_field
----------
        1
(1 row)

********* QUERY **********
SELECT 1.123456789::int4 AS my_field;
**************************
```

```
 my_field
----------
        1
(1 row)

********* QUERY **********
SELECT 1.123456789::bigint AS my_field;
**************************

 my_field
----------
        1
(1 row)

********* QUERY **********
SELECT 1.123456789::int8 AS my_field;
**************************

 my_field
----------
        1
(1 row)

********* QUERY **********
SELECT 1.123456789::smallint AS my_field;
**************************

 my_field
----------
        1
(1 row)

********* QUERY **********
SELECT 1.1234567889::int2 AS my_field;
**************************

 my_field
----------
        1
(1 row)

********* QUERY **********
SELECT 1.123456789::real AS my_field;
**************************

 my_field
-----------
 1.1234568
(1 row)

********* QUERY **********
SELECT 1.123456789::double precision AS my_field;
**************************

  my_field
-------------
 1.123456789
(1 row)

********* QUERY **********
```

```
SELECT 1.123456789::numeric(10,1) AS my_field;
**************************

 my_field
----------
      1.1
(1 row)

********* QUERY **********
SELECT 1.123456789::numeric(10,5) AS my_field;
**************************

 my_field
----------
  1.12346
(1 row)

********* QUERY **********
SELECT 1.123456789::numeric(10,9) AS my_field;
**************************

  my_field
-------------
 1.123456789
(1 row)

********* QUERY **********
DROP TABLE IF EXISTS new_tags;
**************************

DROP TABLE
********* QUERY **********
CREATE TABLE IF NOT EXISTS new_tags (
        pk integer NOT NULL PRIMARY KEY,
        tag char(10)
);
**************************

CREATE TABLE
********* QUERY **********
INSERT INTO new_tags
VALUES
        (1, 'first tag'),
        (2, 'tag');
**************************

INSERT 0 2
********* QUERY **********
SELECT
        pk,
        tag,
        length(tag),
        octet_length(tag),
        char_length(tag)
FROM new_tags;
**************************

 pk |    tag     | length | octet_length | char_length
----+------------+--------+--------------+-------------
  1 | first tag  |      9 |           10 |           9
  2 | tag        |      3 |            10 |           3
(2 rows)

********* QUERY **********
DROP TABLE IF EXISTS new_tags;
**************************

DROP TABLE
********* QUERY **********
CREATE TABLE IF NOT EXISTS new_tags (
        pk serial PRIMARY KEY,
        tag varchar(10)
);
**************************

CREATE TABLE
********* QUERY **********
INSERT INTO new_tags (tag)
VALUES
        ('first tag'),
        ('tag');
**************************

INSERT 0 2
********* QUERY **********
SELECT
        pk,
        tag,
        length(tag),
        octet_length(tag),
        char_length(tag)
FROM new_tags;
**************************

 pk |    tag    | length | octet_length | char_length
----+-----------+--------+--------------+-------------
  1 | first tag |      9 |            9 |           9
  2 | tag       |      3 |            3 |           3
(2 rows)

********* QUERY **********
INSERT INTO new_tags (tag)
VALUES ('this sentence has more than 10 characters');
**************************

********* QUERY **********
DROP TABLE IF EXISTS new_tags;
**************************

DROP TABLE
********* QUERY **********
CREATE TABLE IF NOT EXISTS new_tags (
        pk integer GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
        tag text
);
**************************

CREATE TABLE
********* QUERY **********
INSERT INTO new_tags (tag)
```

```
VALUES
        ('first tag'),
        ('tag'),
        ('this sentence has more than 10 characters');
**************************

INSERT 0 3
********* QUERY **********
SELECT
        pk,
        tag,
        length(tag),
        octet_length(tag),
        char_length(tag)
FROM new_tags;
**************************

 pk |                    tag                    | length | octet_length | c
har_length
----+-------------------------------------------+--------+--------------+--
-----------
  1 | first tag                                 |      9 |            9 |
         9
  2 | tag                                       |      3 |            3 |
         3
  3 | this sentence has more than 10 characters |     41 |           41 |
        41
(3 rows)

********* QUERY **********
SELECT setting FROM pg_settings WHERE name = 'DateStyle';
**************************

 setting
----------
 ISO, MDY
(1 row)

********* QUERY **********
SELECT '12-31-2022'::date;
**************************

    date
------------
 2022-12-31
(1 row)

********* QUERY **********
SELECT to_date('31/12/2022', 'dd/mm//yyyy');
**************************

  to_date
------------
 2022-12-31
(1 row)

********* QUERY **********
SELECT to_date('31/12/22', 'dd/mm/yy');
**************************
```

```
  to_date
------------
 2022-12-31
(1 row)

********* QUERY **********
SELECT
        pk,
        title,
        created_on
FROM posts;
**************************

 pk |     title      |          created_on
----+----------------+-------------------------------
  5 | my orange      | 2023-07-17 11:05:27.216009+03
  8 | my tomato      | 2023-07-17 11:05:27.216009+03
  7 | Re:my orange   | 2023-07-17 11:05:27.216009+03
  9 | my new orange  | 2023-07-17 12:46:07.00367+03
  6 | my new apple   | 2023-07-17 11:05:27.216009+03
(5 rows)

                                      Table "public.posts"
     Column     |           Type           | Collation | Nullable |
   Default
----------------+--------------------------+-----------+----------+--------
----------------------
 pk             | integer                  |           | not null | generat
ed always as identity
 title          | text                     |           |          |
 content        | text                     |           |          |
 author         | integer                  |           | not null |
 category       | integer                  |           | not null |
 reply_to       | integer                  |           |          |
 created_on     | timestamp with time zone |           |          | CURRENT
_TIMESTAMP
 last_edited_on | timestamp with time zone |           |          | CURRENT
_TIMESTAMP
 editable       | boolean                  |           |          | true
Indexes:
    "posts_pkey" PRIMARY KEY, btree (pk)
Foreign-key constraints:
    "posts_author_fkey" FOREIGN KEY (author) REFERENCES users(pk)
    "posts_category_fkey" FOREIGN KEY (category) REFERENCES categories(pk)
    "posts_reply_to_fkey" FOREIGN KEY (reply_to) REFERENCES posts(pk)
Referenced by:
    TABLE "j_posts_tags" CONSTRAINT "j_posts_tags_post_pk_fkey" FOREIGN KEY
 (post_pk) REFERENCES posts(pk)
    TABLE "posts" CONSTRAINT "posts_reply_to_fkey" FOREIGN KEY (reply_to) R
EFERENCES posts(pk)

********* QUERY **********
SELECT
        pk,
        title,
        created_on::date,
        to_char(created_on, 'dd-mm-yyyy') AS european_format_date
FROM posts;
**************************
```

```
 pk |     title      | created_on | european_format_date
----+----------------+------------+----------------------
  5 | my orange      | 2023-07-17 | 17-07-2023
  8 | my tomato      | 2023-07-17 | 17-07-2023
  7 | Re:my orange   | 2023-07-17 | 17-07-2023
  9 | my new orange | 2023-07-17 | 17-07-2023
  6 | my new apple  | 2023-07-17 | 17-07-2023
(5 rows)

********* QUERY **********
DROP TABLE IF EXISTS new_posts;
**************************

DROP TABLE
********* QUERY **********
CREATE TABLE IF NOT EXISTS new_posts AS
SELECT
        pk,
        title,
        created_on::timestamp with time zone AS created_on_t,
        created_on::timestamp without time zone AS created_on_nt
FROM posts;
**************************

SELECT 5
                           Table "public.new_posts"
    Column     |            Type             | Collation | Nullable | Defau
lt
---------------+-----------------------------+-----------+----------+------
---
 pk            | integer                     |           |          |
 title         | text                        |           |          |
 created_on_t  | timestamp with time zone    |           |          |
 created_on_nt | timestamp without time zone |           |          |

********* QUERY **********
SELECT * FROM new_posts;
**************************

 pk |     title      |         created_on_t          |      created_on_nt

----+----------------+-------------------------------+---------------------
------
  5 | my orange      | 2023-07-17 11:05:27.216009+03 | 2023-07-17 11:05:27.2
16009
  8 | my tomato      | 2023-07-17 11:05:27.216009+03 | 2023-07-17 11:05:27.2
16009
  7 | Re:my orange   | 2023-07-17 11:05:27.216009+03 | 2023-07-17 11:05:27.2
16009
  9 | my new orange | 2023-07-17 12:46:07.00367+03  | 2023-07-17 12:46:07.0
0367
  6 | my new apple  | 2023-07-17 11:05:27.216009+03 | 2023-07-17 11:05:27.2
16009
(5 rows)

********* QUERY **********
SHOW timezone;
**************************

   TimeZone
```

```
---------------
 Europe/Moscow
(1 row)

********* QUERY **********
SET timezone='GMT';
**************************

SET
********* QUERY **********
SHOW timezone;
**************************

 TimeZone
----------
 GMT
(1 row)

********* QUERY **********
SELECT * FROM new_posts;
**************************

 pk |     title     |         created_on_t          |       created_on_nt

----+---------------+-------------------------------+---------------------
------
  5 | my orange     | 2023-07-17 08:05:27.216009+00 | 2023-07-17 11:05:27.2
16009
  8 | my tomato     | 2023-07-17 08:05:27.216009+00 | 2023-07-17 11:05:27.2
16009
  7 | Re:my orange  | 2023-07-17 08:05:27.216009+00 | 2023-07-17 11:05:27.2
16009
  9 | my new orange | 2023-07-17 09:46:07.00367+00  | 2023-07-17 12:46:07.0
0367
  6 | my new apple  | 2023-07-17 08:05:27.216009+00 | 2023-07-17 11:05:27.2
16009
(5 rows)

********* QUERY **********
SET timezone='Europe/Moscow';
**************************

SET
********* QUERY **********
SHOW timezone;
**************************

   TimeZone
---------------
 Europe/Moscow
(1 row)

********* QUERY **********
SELECT
        p.pk,
        p.title,
        u.username,
        c.title AS category
FROM posts AS p
INNER JOIN users AS u ON u.pk = p.author
```

```
LEFT JOIN categories AS c ON c.pk = p.category
ORDER BY 1;
**************************

 pk |     title     |  username  | category
----+---------------+------------+----------
  5 | my orange     | myusername | orange
  6 | my new apple  | myusername | apple
  7 | Re:my orange  | scotty     | orange
  8 | my tomato     | scotty     | tomato
  9 | my new orange | myusername | orange
(5 rows)

********* QUERY **********
SELECT
        p.pk,
        p.title,
        hstore(ARRAY['username', u.username, 'category', c.title])
FROM posts AS p
INNER JOIN users AS u ON u.pk = p.author
LEFT JOIN categories AS c ON c.pk = p.category
ORDER BY 1;
**************************

 pk |     title     |                     hstore
----+---------------+------------------------------------------------
  5 | my orange     | "category"=>"orange", "username"=>"myusername"
  6 | my new apple  | "category"=>"apple", "username"=>"myusername"
  7 | Re:my orange  | "category"=>"orange", "username"=>"scotty"
  8 | my tomato     | "category"=>"tomato", "username"=>"scotty"
  9 | my new orange | "category"=>"orange", "username"=>"myusername"
(5 rows)

********* QUERY **********
DROP TABLE IF EXISTS posts_options;
**************************

DROP TABLE
********* QUERY **********
CREATE TABLE IF NOT EXISTS posts_options AS
SELECT
        p.pk,
        p.title,
        hstore(ARRAY['username', u.username, 'category', c.title]) AS optio
ns
FROM posts AS p
INNER JOIN users AS u ON u.pk = p.author
LEFT JOIN categories AS c ON c.pk = p.category
ORDER BY 1;
**************************

SELECT 5
            Table "public.posts_options"
 Column  |  Type   | Collation | Nullable | Default
---------+---------+-----------+----------+---------
 pk      | integer |           |          |
 title   | text    |           |          |
 options | hstore  |           |          |

********* QUERY **********
```

```
SELECT * FROM posts_options
WHERE options->'category' = 'orange';
**************************

 pk |     title     |                     options
----+---------------+------------------------------------------------
  5 | my orange     | "category"=>"orange", "username"=>"myusername"
  7 | Re:my orange  | "category"=>"orange", "username"=>"scotty"
  9 | my new orange | "category"=>"orange", "username"=>"myusername"
(3 rows)

********* QUERY **********
INSERT INTO posts_options (pk, title, options)
VALUES (7, 'my last post', '"enabled"=>"false"');
**************************

INSERT 0 1
********* QUERY **********
SELECT * FROM posts_options;
**************************

 pk |     title     |                     options
----+---------------+------------------------------------------------
  5 | my orange     | "category"=>"orange", "username"=>"myusername"
  6 | my new apple  | "category"=>"apple", "username"=>"myusername"
  7 | Re:my orange  | "category"=>"orange", "username"=>"scotty"
  8 | my tomato     | "category"=>"tomato", "username"=>"scotty"
  9 | my new orange | "category"=>"orange", "username"=>"myusername"
  7 | my last post  | "enabled"=>"false"
(6 rows)

********* QUERY **********
SELECT
        p.pk,
        p.title,
        t.tag
FROM posts AS p
LEFT JOIN j_posts_tags AS jpt ON p.pk = jpt.post_pk
LEFT JOIN tags AS t ON t.pk = jpt.tag_pk
ORDER BY 1;
**************************

 pk |     title     |    tag
----+---------------+------------
  5 | my orange     | vegetables
  5 | my orange     | fruits
  6 | my new apple  | fruits
  7 | Re:my orange  |
  8 | my tomato     |
  9 | my new orange | fruits
(6 rows)

********* QUERY **********
SELECT
        p.pk,
        p.title,
        string_agg(t.tag, ', ') AS tag
FROM posts AS p
LEFT JOIN j_posts_tags AS jpt ON p.pk = jpt.post_pk
LEFT JOIN tags AS t ON t.pk = jpt.tag_pk
```

```
GROUP BY 1, 2
ORDER BY 1;
**************************

 pk |     title     |        tag
----+---------------+--------------------
  5 | my orange     | vegetables, fruits
  6 | my new apple  | fruits
  7 | Re:my orange  |
  8 | my tomato     |
  9 | my new orange | fruits
(5 rows)

********* QUERY **********
WITH cte AS (
SELECT
        p.pk,
        p.title,
        string_agg(t.tag, ', ') AS tag
FROM posts AS p
LEFT JOIN j_posts_tags AS jpt ON p.pk = jpt.post_pk
LEFT JOIN tags AS t ON t.pk = jpt.tag_pk
GROUP BY 1, 2
ORDER BY 1)
SELECT row_to_json(q) AS json_data
FROM cte AS q;
**************************

                        json_data
---------------------------------------------------------
 {"pk":5,"title":"my orange","tag":"vegetables, fruits"}
 {"pk":6,"title":"my new apple","tag":"fruits"}
 {"pk":7,"title":"Re:my orange","tag":null}
 {"pk":8,"title":"my tomato","tag":null}
 {"pk":9,"title":"my new orange","tag":"fruits"}
(5 rows)

********* QUERY **********
DROP TABLE IF EXISTS post_json;
**************************

DROP TABLE
********* QUERY **********
CREATE TABLE IF NOT EXISTS post_json (json_data jsonb);
**************************

CREATE TABLE
              Table "public.post_json"
  Column   | Type  | Collation | Nullable | Default
-----------+-------+-----------+----------+---------
 json_data | jsonb |           |          |

********* QUERY **********
INSERT INTO post_json (json_data)
SELECT row_to_json(q) FROM (
SELECT
        p.pk,
        p.title,
        string_agg(t.tag, ', ') AS tag
FROM posts AS p
LEFT JOIN j_posts_tags AS jpt ON p.pk = jpt.post_pk
LEFT JOIN tags AS t ON t.pk = jpt.tag_pk
GROUP BY 1, 2
ORDER BY 1) AS q;
**************************

INSERT 0 5
********* QUERY **********
SELECT jsonb_pretty(json_data) FROM post_json;
**************************

           jsonb_pretty
----------------------------------
 {                               +
     "pk": 5,                    +
     "tag": "vegetables, fruits",+
     "title": "my orange"        +
 }
 {                               +
     "pk": 6,                    +
     "tag": "fruits",            +
     "title": "my new apple"     +
 }
 {                               +
     "pk": 7,                    +
     "tag": null,                +
     "title": "Re:my orange"     +
 }
 {                               +
     "pk": 8,                    +
     "tag": null,                +
     "title": "my tomato"        +
 }
 {                               +
     "pk": 9,                    +
     "tag": "fruits",            +
     "title": "my new orange"    +
 }
(5 rows)

********* QUERY **********
SELECT jsonb_pretty(json_data) FROM post_json
WHERE json_data @> '{"tag":"fruits"}';
**************************

         jsonb_pretty
------------------------------
 {                           +
     "pk": 6,                +
     "tag": "fruits",        +
     "title": "my new apple" +
 }
 {                           +
     "pk": 9,                +
     "tag": "fruits",        +
     "title": "my new orange"+
 }
(2 rows)
```