Postico
Iterm2
Oh my zsh
Mockaroo

```
test=# CREATE TABLE person (
test(# id BIGSERIAL NOT NULL PRIMARY KEY,
test(# first_name VARCHAR(50) NOT NULL,
test(# last_name VARCHAR(50) NOT NULL,
test(# gender VARCHAR(7) NOT NULL,
test(# date_of_birth DATE NOT NULL,
test(# email VARCHAR(150) );
CREATE TABLE
test=# \D
invalid command \D
Try \? for help.
test=# \d
           List of relations
 Schema |     Name      |   Type   | Owner
--------+---------------+----------+-------
 public | person        | table    | Nick
 public | person_id_seq | sequence | Nick
(2 rows)

test=# \d person
                              Table "public.person"
   Column     |         Type          | Collation | Nullable |             Default
--------------+-----------------------+-----------+----------+---------------------------------
 id           | bigint                |           | not null | nextval('person_id_seq'::regclass)
 first_name   | character varying(50) |           | not null |
 last_name    | character varying(50) |           | not null |
 gender       | character varying(7)  |           | not null |
 date_of_birth | date                 |           | not null |
 email        | character varying(150) |          |          |
Indexes:
    "person_pkey" PRIMARY KEY, btree (id)

test=#

test=# \dt
       List of relations
 Schema |  Name  | Type  | Owner
--------+--------+-------+-------
 public | person | table | Nick
```

(1 row)

test=# INSERT INTO person (first_name, last_name, gender, date_of_birth)
test-# VALUES ('Anne', 'Smith', 'FEMALE', date '1988-01-09');
test'#
test=# INSERT INTO person (first_name, last_name, gender, date_of_birth, email) VALUES
('Jake', 'Jones', 'MALE', date '1990-12-31', 'jake@gmail.com');
INSERT 0 1
test=#

Creating databases
Adding columns
Creating sql database through mockaroo and uploading into vs code to edit and then uploading database into Iterm2

test=# SELECT FROM person;
test=# SELECT first_name FROM person;
test=# SELECT first_name, last_name FROM person;
test=# SELECT email FROM person;
test=# SELECT date_of_birth FROM person;
test=# SELECT * FROM person ORDER BY country_of_birth ACS;
ERROR: syntax error at or near "ACS"
LINE 1: SELECT * FROM person ORDER BY country_of_birth ACS;
                                     ^
test=# SELECT * FROM person ORDER BY country_of_birth;
test=# SELECT * FROM person ORDER BY country_of_birth DESC;
test=# SELECT * FROM person ORDER BY id;
ERROR: column "id" does not exist
LINE 1: SELECT * FROM person ORDER BY id;
                            ^
test=# SELECT * FROM person;
test=# SELECT * FROM person ORDER by email;
test=# SELECT * FROM person ORDER by date_of_birth DESC;
test=# SELECT country_of_birth FROM person ORDER by country_of_birth;
test=# SELECT DISTINCT country_of_birth FROM person ORDER BY country_of_birth;
test=# SELECT DISTINCT country_of_birth FROM person ORDER BY country_of_birth DESC;

test=# SELECT * FROM person WHERE gender = 'Male' AND (country_of_birth = 'Poland' OR
country_of_birth = 'China') AND last_name = 'Pietersma';   first_name |
last_name | email | gender | date_of_birth | country_of_birth

test=# SELECT * FROM person WHERE country_of_birth LIKE 'P%';
test=# SELECT * FROM person WHERE country_of_birth ILIKE 'p%';

```
SELECT * FROM person WHERE email LIKE '________@';
SELECT * FROM person WHERE email LIKE '%@google.com';
SELECT *
test-# FROM person
test-# WHERE country_of_birth IN ('China', 'Brazil', 'France');
SELECT * FROM person OFFSET 5 LIMIT 5;
SELECT country_of_birth, COUNT(*)  FROM person GROUP BY country_of_birth HAVING
COUNT(*) > 50 ORDER BY country_of_birth;
 country_of_birth | count
SELECT country_of_birth, COUNT(*)  FROM person GROUP BY country_of_birth HAVING
COUNT(*) > 5 ORDER BY country_of_birth;

/I  followed by path to access file in terminal.k

SELECT make, model, MIN(price) FROM car GROUP BY make, model;

SELECT id, make, model,ROUND(price, price * .10, 2), ROUND(price - (price * .10), 2  FROM car;

SELECT id, make, model, price AS original_price, ROUND(price, price * .10, 2) AS ten_percent,
ROUND(price - (price * .10), 2) AS discount_after_10_percent FROM car;

SELECT COALESCE(email, 'Email not provided') FROM person;

SELECT COALESCE( 10 / NULLIF(0, 0), 0);

SELECT NOW()::DATE;
SELECT NOW()::TIME;
SELECT NOW() - INTERVAL '1 YEAR';
SELECT NOW() + INTERVAL '10 DAYS';
SELECT EXTRACT(MONTH FROM NOW());
SELECT EXTRACT(YEAR FROM NOW());
SELECT first_name, last_name, gender, country_of_birth, date_of_birth, AGE(NOW(),
date_of_birth) AS age FROM person;
Cannot duplicate primary key because of unique constraint
ALTER TABLE person DROP CONSTRAINT person_pkey;
ALTER TABLE person ADD CONSTRAINT unique_email_address UNIQUE (email);
ALTER TABLE person ADD CONSTRAINT gender_contraint CHECK (gender = 'Female' OR gender
= 'MALE');
SELECT DISTINCT gender FROM person;
UPDATE
DELETE
ON CONFLICT () DO NOTHING;
ADDING RELATIONSHIP BETWEEN TABLES
UPDATING FOREIGN KEY COLUMNS
```

```
UPDATE person SET car_id = 2 WHERE id = 1;
INNER JOINS
SELECT * FROM person
JOIN car ON person.car_id = car.id;
LEFT JOINS
SELECT * FROM person
LEFT JOIN car ON person.car_id = car.id;
\copy will copy queries so you can save output
BIGINT BIGSERIAL
SELECT * FROM pg_available_extensions;
UUID Universally Unique Identifier is
CREATE EXTENSION IF NOT EXISTS "UUID-OSSP";
SELECT uuid_generate_v4();
         uuid_generate_v4
--------------------------------------
 dccda1c2-2227-427c-8cf0-8ebe0b1eb137
UUID as primary keys
INDEXES
FUNCTIONS
CTE COMMON TABLE EXPRESSIONS
TRIGGERS
VIEWS
```