```
      ***COMMIT and ROLLBACK*** -> if we do not want to disallow auto-commit

      ***
      CREATE SEQUENCE person_id_by_2_sequence
      START 6
      INCREMENT 2
      OWNED BY person.id
      -- owned by table person, column id
      -- if we delete the table person or column id, this sequence will be
automatically deleted


      ALTER TABLE person
      ALTER COLUMN id SET DEFAULT nextval('person_id_by_2_sequence');

      ALTER TABLE person
      DROP CONSTRAINT person_pkey

      ALTER TABLE person
      ADD COLUMN new_id SERIAL PERIMARY KEY
      ***

1. Lexical Structure in PostgreSQL:
      - Keywords and unquoted table/column names are case insensitive
      CREATE TABLE customer();

      - String literals are enclosed in 'single quotes'
      'bogomila'

      - Table and column names containing special symbols use "double quotes"(e.g.
if we use capital letters or keywords)
      /*
      - Below is how you add comment
      -- comment
      */

      - Adding a comment on multiple lines or anywhere is the SQL statement
      CREATE TABLE /* comment */ new_table;

2. Retrieving Data: Using SQL Select:

      SELECT * FROM clients;
      SELECT first_name, last_name FROM clients;

      - Aliases rename a table or a column heading
      SELECT p.first_name AS 'First Name'
      FROM clients AS p;

      - Concatenate operator: you can concatenate column records using the ||
operator
      SELECT first_name || ' ' || last_name AS "Full Name"
      FROM clients;

      ***
      INSTEAD USE BELOW:
      SELECT concat('Karlson',' ','ot','Pokriva')

      SELECT
            p.first_name AS "First Name",
```

```
      p.last_name AS "Last Name",
      a.birthday AS "Age",
FROM
      person AS p,
      age AS a;
***

- Limiting the selected rows - could be returned fewer
SELECT id, first_name FROM clients LIMIT 2;

***
SELECT * from employees
LIMIT 3;
***

-Sorting with ORDER BY
SELECT last_name, salary
FROM employees
ORDER BY salary;

SELECT last_name, salary
FROM employees
ORDER BY salary DESC;
```

3. Filtering Selected Rows: Using SQL WHERE Clause:

```
- Eliminate duplicate results:
SELECT DISTINCT first_name FROM employees;
***
SELECT
      DISTINCT ON (first_name) first_name, --> distinct is only for the
first_name, but not last_name
      last_name
FROM
      employees
***

- Eliminate duplicate results based on the combination of values
SELECT DISTINCT first_name, last_name FROM employees;

- Eliminate duplicate results based on the first column
SELECT DISTINCT ON (first_name) first_name, last_name
FROM employees;

- Filtering the selected rows
SELECT id, first_name, last_name
FROM employees
WHERE id <= 2;


SELECT id,
      first_name || + ' ' || last_name
            AS full_name,
      job_title, salary
FROM employees
WHERE salary > 1000.00
ORDER BY id;

- Logical operators: AND, NOT, OR
```

```
SELECT last_name FROM employees
WHERE salary = 900 and first_name = 'John'

SELECT last_name FROM employees
WHERE NOT salary = 900;

SELECT last_name FROM employees
WHERE salary = 900 or salary = 1100;

SELECT last_name, salary FROM employees
WHERE salary >= 900 AND salary <=2100;

SELECT last_name, salary FROM employees
WHERE salary BETWEEN 900 AND 2100;

SELECT first_name, last_name
FROM employees
WHERE salary IN(2100, 1100, 900); --> Use IN to specify a set of values

SELECT first_name, last_name
FROM employees
WHERE salary NOT IN (2100, 1100, 900);


SELECT * FROM employees
WHERE salary >= 1000.00
      AND departnment_id = 4
ORDER BY id;

- Comparing with NULL - null is a special value that means missing value
- Not the same as 0 or blank space
***
USE:
      WHERE last_name IS NULL; // WHERE last_name IS NOT NULL;
NOT:
      WHERE last_name = NULL; > it always returns False
**

SELECT first_name, room_id FROM clients
WHERE last_name IS NULL;

SELECT first_name, room_id FROM clients
WHERE last_name IS NOT NULL;
```

4. Data Manipulation: Using SQL INSERT, UPDATE, DELETE

```
- Inserting Data

INSERT INTO towns
VALUES (33, 'Paris');

INSERT INTO towns(name)
VALUES ('Sofia');

-Bulk data can be recorded in a single query
INSERT INTO towns(name)
VALUES ('London'),
         ('Rome');
```

```
      - You can use existing records to create a new table
      CREATE TABLE customer_data
      AS SELECT id, last_name, room_id
      FROM clients;

      -Or into an existing table
      INSERT INTO projects(name, start_date)
      SELECT name || 'Restructuring', '2023-01-01'
      FROM departments;

      - Updating Data

      UPDATE employees
      SET last_name = 'Brown'
      WHERE id=1;

      UPDATE employees
      SET salary = salary * 1.1l,
            job_titile = 'Senior' || job_title
      WHERE department_id = 3; --> if you skip the WHERE clause all rows in the
table will be updated

      - Deleting Data

      DELETE FROM employees
      WHERE id=1; --> if you skip the WHERE clause all rows in the table will be
deleted

      TRUNCATE TABLE employees; --> Truncate works faster than DELETE --> delete
all rows from a table


5. Views: CREATE VIEW .. AS:
      - Views are virtual tables made from other tables, views, or joins between
them
      - When you create a view, you basically create a query and assign a name to
the query

      CREATE VIEW hr_result_set AS
      SELECT employees.first_name || ' ' || employees.last_name
      AS "Full Name",
      employees.salary
      FROM employees
      ORDER BY department_id;


      CREATE VIEW top_paid_employee AS
      SELECT * FROM employees
      ORDER BY salary SESC LIMIT 1;

      ***
      HOW TO PLACE A ROW IN A SPECIFIC PLACE IN THE TABLE:

      CREATE TABLE person(
      id SERIAL PRIMARY KEY,
      first_name VARCHAR(20),
      last_name VARCHAR(20)
      );
```

```
INSERT INTO person(first_name, last_name)
VALUES
      ('Mila', 'M'),
      ('Mina', 'L');

ALTER TABLE person
RENAME TO person_old;


CREATE TABLE person(
      id SERIAL PRIMARY KEY,
      age INT,
      first_name VARCHAR(20),
      last_name VARCHAR(20)
);

INSERT INTO person(first_name, last_name)
SELECT first_name, last_name
FROM person_old;

DROP RABLE person_old;
***

***mockaroo.com - for generating data***
```