1. Создайте функцию, которая делает бэкап таблицы customers (копирует все данные в другую таблицу), предварительно стирая таблицу для бэкапа, если такая уже существует (чтобы в случае многократного запуска таблица для бэкапа перезатиралась).

```
CREATE OR REPLACE FUNCTION back_up() RETURNS void AS $$

    DROP TABLE IF EXISTS customers_backup;

    SELECT * INTO customers_backup

    FROM customers;

$$ LANGUAGE SQL;
```

2. Создать функцию, которая возвращает средний фрахт (freight) по всем заказам

```
CREATE OR REPLACE FUNCTION freight_avg(OUT freight_avg double precision)

AS $$

    SELECT AVG(freight)

    FROM orders;

$$ LANGUAGE SQL;
```

3. Написать функцию, которая принимает два целочисленных параметра, используемых как нижняя и верхняя границы для генерации случайного числа в пределах этой границы (включая сами граничные значения).

Функция random генерирует вещественное число от 0 до 1.

Необходимо вычислить разницу между границами и прибавить единицу.

На полученное число умножить результат функции random() и прибавить к результату значение нижней границы.

Применить функцию floor() к конечному результату, чтобы не "уехать" за границу и получить целое число.

```
CREATE OR REPLACE FUNCTION rand_num(low int, high int) RETURNS int

AS $$

BEGIN

    RETURN FLOOR(random() * (high -low+1) + low);

END

$$ LANGUAGE plpgsql;
```

4. Создать функцию, которая возвращает самые низкую и высокую зарплаты среди сотрудников заданного города

```
CREATE OR REPLACE FUNCTION min_max_salary(ch_city varchar,
                                          OUT min_sal numeric,
                                          OUT max_sal numeric)
AS $$
   SELECT MIN(salary), MAX(salary)
        FROM employees
        WHERE city = ch_city
$$ LANGUAGE SQL;


SELECT * FROM min_max_salary('London')
```

5. Создать функцию, которая корректирует зарплату на заданный процент, но не корректирует зарплату, если её уровень превышает заданный уровень при этом верхний уровень зарплаты по умолчанию равен 70, а процент коррекции равен 15%.

```
CREATE OR REPLACE FUNCTION salary_correct() RETURNS SETOF employees
AS $$
DECLARE
employee record;
BEGIN
        FOR employee IN SELECT * FROM employees
        LOOP
                IF employee.salary < 70 THEN
                        employee.salary = employee.salary*1.15;
                END IF;
        RETURN NEXT employee;
        END LOOP;
```

```
END

$$ LANGUAGE plpgsql;
```

6. Модифицировать функцию, корректирующую зарплату таким образом, чтобы в результате коррекции, она так же выводила бы изменённые записи.

7. Модифицировать предыдущую функцию так, чтобы она возвращала только колонки last_name, first_name, title, salary

```
 CREATE OR REPLACE FUNCTION salary_cor()

RETURNS TABLE(last_name varchar,first_name varchar,title varchar, salary numeric)
AS

$$

        UPDATE employees

        SET salary = salary * 1.15

        WHERE salary < 70

        RETURNING last_name, first_name, title, salary

$$ LANGUAGE SQL;



SELECT * FROM salary_cor();
```

8. Написать функцию, которая принимает метод доставки и возвращает записи из таблицы orders в которых freight меньше значения, определяемого по следующему алгоритму:

- ищем максимум фрахта (freight) среди заказов по заданному методу доставки

- корректируем найденный максимум на 30% в сторону понижения

- вычисляем среднее значение фрахта среди заказов по заданному методому доставки

- вычисляем среднее значение между средним найденным на предыдущем шаге и скорректированным максимумом

- возвращаем все заказы в которых значение фрахта меньше найденного на предыдущем шаге среднего

```
CREATE OR REPLACE FUNCTION get_orders(ship_method int)

RETURNS SETOF orders AS $$

DECLARE
```

```
    average numeric;
    maximum numeric;
    middle numeric;
BEGIN
    SELECT MAX(freight) INTO maximum
    FROM orders
    WHERE ship_via = ship_method;

    maximum = maximum * 0.7;

    SELECT AVG(freight) INTO average
    FROM orders
    WHERE ship_via = ship_method;

    middle = (maximum + average) / 2;

    RETURN QUERY
    SELECT *
    FROM orders
    WHERE freight < middle;
END
$$ LANGUAGE plpgsql
```

9. Написать функцию, которая принимает:

уровень зарплаты, максимальную зарплату (по умолчанию 80) минимальную зарплату (по умолчанию 30), коээфициет роста зарплаты (по умолчанию 20%)

Если зарплата выше минимальной, то возвращает false

Если зарплата ниже минимальной, то увеличивает зарплату на коэффициент роста и проверяет не станет ли зарплата после повышения превышать максимальную.

Если превысит - возвращает false, в противном случае true.

Проверить реализацию, передавая следующие параметры

(где c - уровень з/п, max - макс. уровень з/п, min - минимальный уровень з/п, r - коэффициент):

c = 40, max = 80, min = 30, r = 0.2 - должна вернуть false

c = 79, max = 81, min = 80, r = 0.2 - должна вернуть false

c = 79, max = 95, min = 80, r = 0.2 - должна вернуть true

```
CREATE FUNCTION work_with_salary(c int, max int DEFAULT 80,
                                        min int DEFAULT 30,
                                         r float8  DEFAULT 0.2)
                                        RETURNS bool
AS $$
BEGIN
        IF c > min THEN
                RETURN'false';
        ELSEIF c < min THEN
                c = c + (c*r);
                IF c > max THEN
                        RETURN 'false';
                ELSE
                        RETURN 'true';
                END IF;
        END IF;
END
$$ LANGUAGE plpgsql;
```