# Ответы на вопросы

Студент: Болдинов Алексей ЭФМО-02-24

1. Для чего используются команды **LEAD**, **ROLLUP**, **HAVING**, **GROUPING**?

**LEAD** — это функция, которая помогает посмотреть на следующее значение в строках таблицы, не меняя сам запрос.

**ROLLUP** — используется с GROUP BY и помогает создать несколько уровней агрегации.

**HAVING** — используется после агрегации (после GROUP BY) для фильтрации данных.

**GROUPING** — эта функция помогает понять, на каком уровне иерархии мы находимся, когда используем ROLLUP или CUBE.

1. Перечислите агрегатные функции в PostgreSQL

COUNT() — считает количество строк.

SUM() — суммирует значения.

AVG() — вычисляет среднее значение.

MIN() — находит минимальное значение.

MAX() — находит максимальное значение.

### EXTRACT() — извлекает часть даты или времени (например, год, месяц, день) из типа данных DATE, TIME, TIMESTAMP.

### TO\_DATE() — преобразует строку в формат даты DATE, используя заданный формат.

### ARRAY\_LENGTH() — возвращает количество элементов в массиве.

### format() — форматирует строку, вставляя значения в заданный шаблон.

### COALESCE() — возвращает первый ненулевой (не NULL) аргумент из списка.

### jsonb\_build\_object() — создаёт объект JSON из пар "ключ-значение" для типа данных jsonb.

### TO\_JSONB() — преобразует данные в тип jsonb, представляющий объект JSON.

### CONCAT() — объединяет несколько строк в одну.

### RANK() — присваивает ранг строкам в пределах окна (группы данных) с учетом их порядка.

### XMLAGG() — агрегирует строки в одно XML-значение.

### XMLATTRIBUTES() — добавляет атрибуты к элементам XML.

### XMLELEMENT() — создаёт элемент XML с заданным именем и содержимым.

### UNNEST() — преобразует массив в строки, то есть разворачивает массив в несколько строк.

1. Как задать цикл в PostgreSQL?

В PostgreSQL можно использовать цикл, например, с помощью LOOP, FOR или WHILE.

Пример простого цикла с LOOP:

DECLARE

i INT := 1;

BEGIN

LOOP

EXIT WHEN i > 10; -- когда i больше 10, выходим из цикла

RAISE NOTICE 'i = %', i; -- выводим значение i

i := i + 1;

END LOOP;

END;

Пример цикла с FOR, который проходит по числам от 1 до 10:

BEGIN

FOR i IN 1..10 LOOP -- цикл от 1 до 10

RAISE NOTICE 'i = %', i; -- выводим значение i

END LOOP;

END;

Пример с WHILE, где цикл будет работать, пока i меньше или равно 10:

DECLARE

i INT := 1;

BEGIN

WHILE i <= 10 LOOP

RAISE NOTICE 'i = %', i; -- выводим значение i

i := i + 1;

END LOOP;

ENDж