1. **Что такое расширенные группировки?**

**Расширенные группировки** — это возможность в SQL, позволяющая выполнять более сложные операции агрегации, чем простая группировка по одной колонке.

1. **Приведите примеры использования расширенных группировок.**
   * **ROLLUP**: Создает иерархическую агрегацию, добавляя промежуточные итоги на разных уровнях. Например, если у вас есть данные о продажах по регионам и месяцам, ROLLUP может вернуть итоги по месяцам, затем по регионам и общий итог по всем продажам.
   * **CUBE**: Генерирует агрегаты для всех возможных комбинаций указанных столбцов. Например, если вы анализируете продажи по продуктам и регионам, CUBE вернет итоги для каждой комбинации продукта и региона, а также общие итоги по каждому из них.
   * **GROUPING SETS**: Позволяет явно определить, какие комбинации группировок должны быть возвращены. Это дает возможность комбинировать разные наборы группировок в одном запросе, что делает его более гибким.
2. **Для чего предназначена функция GROUPING ()?**

Функция GROUPING() помогает понять, был ли определенный столбец объединен при использовании расширенной группировки, такой как ROLLUP или CUBE. Она возвращает 1, если столбец объединен, и 0, если нет. Это полезно, чтобы знать, какие данные суммированы.

1. **Для чего предназначена функция GROUP\_ID ()?**

Функция GROUP\_ID() возвращает уникальный идентификатор для каждой группы в результате расширенной группировки. Это полезно для определения, к какой группе относится конкретная строка в результате запроса.

1. **Для чего предназначена функция GROUPING\_ID ()?**

Функция GROUPING\_ID() показывает, какие столбцы были объединены в группе, возвращая битовую маску. Это помогает быстро понять, к какой группе относятся данные в результате запроса.

1. **Что такое составные столбцы?**

Составные столбцы — это столбцы, которые образуются путем объединения нескольких отдельных столбцов. Например, вы можете создать составной ключ из столбцов first\_name и last\_name.

1. **Что такое GROUPING SETS?**

GROUPING SETS — это возможность в SQL, которая позволяет определять, какие комбинации группировок должны быть возвращены. Вы можете указать несколько наборов группировок в одном запросе, что позволяет получать агрегацию по различным критериям.

1. **Поясните синтаксис аналитических функций.**

**function\_name (arguments) OVER (**

**[PARTITION BY partition\_expression]**

**[ORDER BY order\_expression]**

**[ROWS | RANGE frame\_specification]**

**)**

* function\_name: название функции (например, SUM, AVG).
* PARTITION BY: делит набор данных на группы.
* ORDER BY: задает порядок обработки строк.
* ROWS или RANGE: определяет рамки для вычисления функции.

1. **Перечислите виды аналитических функций.**

* **Агрегатные функции**
* **Функции ранжирования**
* **Функции для анализа**

1. **Перечислите известные вам аналитические функции каждого вида.**

* **Агрегатные функции**: SUM, AVG, COUNT, MIN, MAX.
* **Функции ранжирования**: ROW\_NUMBER(), RANK(), DENSE\_RANK(), NTILE().
* **Функции для анализа**: LEAD(), LAG(), FIRST\_VALUE(), LAST\_VALUE().

1. **Поясните секцию ORDER BY аналитических функций.**

Секция ORDER BY задает порядок строк, когда функция выполняется. Это важно для функций, которые зависят от порядка, таких как ROW\_NUMBER(), RANK(), и для получения значений, относящихся к предыдущим или следующим строкам, как в случае с LEAD() и LAG().

1. **Поясните секцию PARTITION BY аналитических функций.**

Секция PARTITION BY разбивает набор данных на отдельные группы, к которым применяются аналитические функции. Например, если вы хотите вычислить среднюю зарплату по каждому отделу, вы можете использовать PARTITION BY department.

1. **Поясните ключевые слова ROWS и RANGE аналитических функций.**

* **ROWS**: Определяет физические строки в пределах текущего окна. Например, ROWS BETWEEN 1 PRECEDING AND CURRENT ROW будет включать текущую строку и одну строку перед ней.
* **RANGE**: Определяет диапазон значений в пределах текущего окна. Например, RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW включает все строки до текущей строки, которые имеют те же значения в столбце ORDER BY.

1. **Опишите, как работает вложенность аналитических функций.**

Вложенность аналитических функций позволяет использовать одну функцию внутри другой. Например, вы можете сначала вычислить среднее значение с помощью AVG(), а затем использовать RANK() для ранжирования значений по этому среднему. Однако следует помнить, что вложенные функции могут увеличивать сложность запросов и должны использоваться осторожно для обеспечения читаемости и производительности.