**Базы данных**

**Отчет по лабораторной работе №6, Лешук Дмитрий ПОИБМС 7-2**

**Цель**: изучить возможность группировки с помощью секции GROUP BY, её совместное использование с агрегирующими функциями AVG(), MAX(), COUNT(), MIN(), SUM(), а также изучить секцию HAVING фильтрации сортировки. Изучить такие операции над множествами, как UNION, INTERSECT, UNION ALL, EXCEPT.

GROUP BY – разбиение множества строк, сформированных секциями FROM и WHERE, на группы в соответствии со значениями в заданных столбцах, а также выполнение вычислений над группами строк с помощью наиболее часто используемых функций. При использовании секции GROUP BY в SELECT-списке допускается указывать только те столбцы, по которым осуществляется группировка.

В данном запросе, представленным на рисунке 1.1, необходимо было разработать запрос, вычисляющий для каждого типа аудиторий максимальную, минимальную, среднюю вместимость аудиторий, суммарную вместимость всех аудиторий и общее количество аудиторий данного типа.

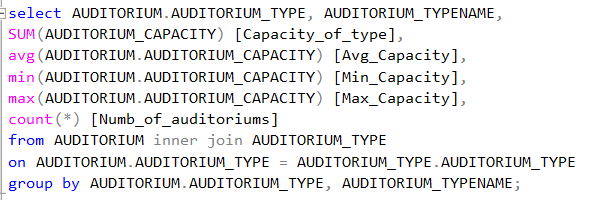


Рисунок 1.1 – Запрос с группировкой данных и агрегирующими функциями.

В данном запросе мы производим группировку по двум столбцам: AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME. Также в выбираемых столбцах мы указываем агрегирующие функции, которые будут выполняться для каждого типа аудиторий, и проводить вычисления с количеством строк и вместимостью аудиторий.

В следующем задании мы используем конструкцию CASE…WHEN в секции группировки – рисунок 1.2, группировку мы осуществляем по промежутку оценок. На этом же рисунке представлен результирующий набор запроса.

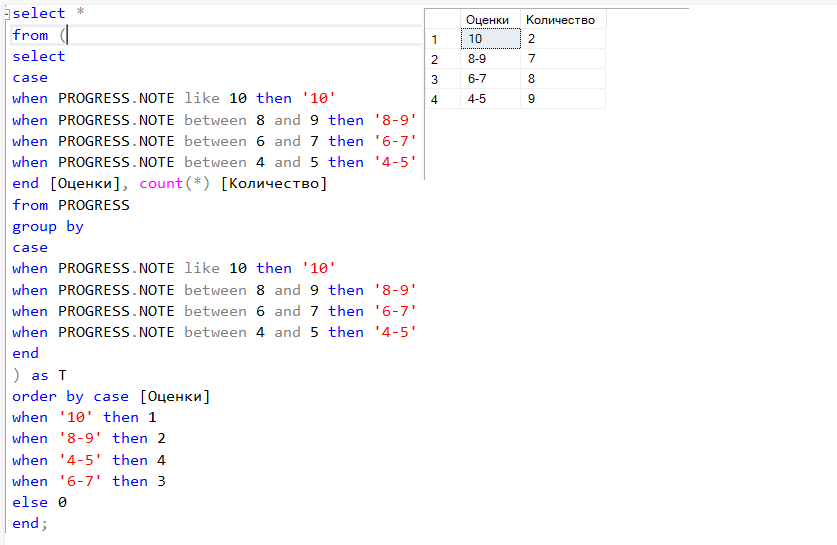


Рисунок 1.2 – Использование CASE при группировке данных

Далее мы используем функцию CAST, которая выполняет привидение типов для нашего запроса, функция ROUND обеспечивает расчет значений с точностью до указанного числа знаков после запятой.

Далее – ROLLUP и CUBE в конструкции GROUP BY. Оператор ROLLUP добавляет суммирующую строку в результирующий набор. При группировке по нескольким критериям ROLLUP будет создавать суммирующую строку для каждой из подгрупп. CUBE же добавляет суммирующие строки для каждой комбинации групп. Всевозможные варианты представлены на рисунке 1.3.

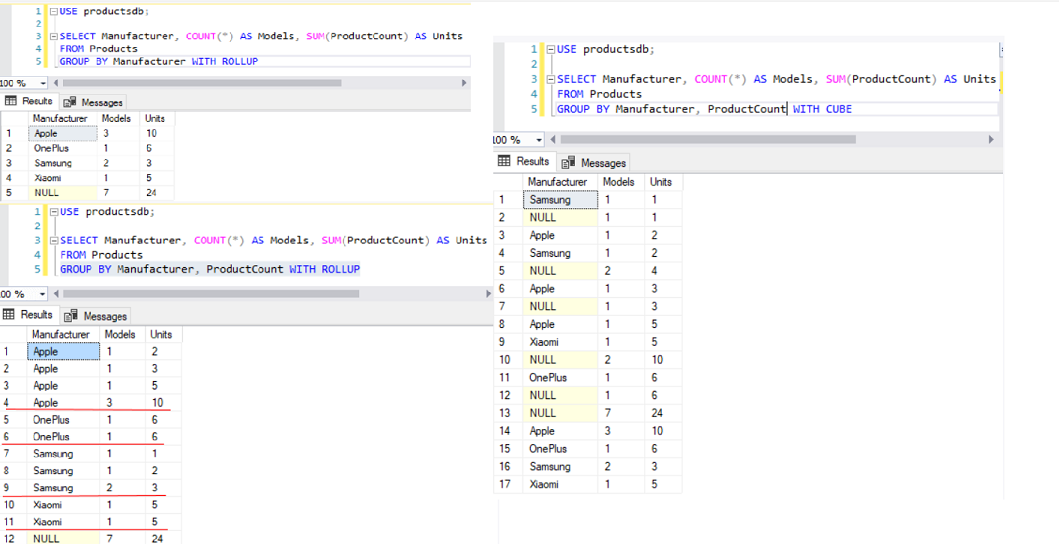


Рисунок 1.3 – ROLLUP, CUBE в секции GROUP BY

Следующая тема, которую затрагивает лабораторная работа – операции над множествами. Представим, что у нас есть 2 SELECT-запроса, и результирующий набор каждого – множество. При указании оператора INTERSECT наш результирующий запрос будет содержать только те строки, которые содержатся в обоих множествах – рисунок 1.4. SQL оператор EXCEPT используется для возврата всех строк в первом операторе SELECT, которые не возвращаются вторым оператором SELECT – рисунок 1.4.

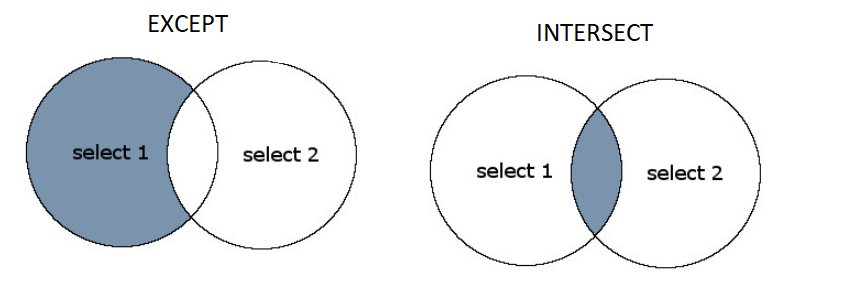


Рисунок 1.4 – Оператор Intersect и Except

UNION и UNION ALL производят объединение результирующей выборки 2-х наборов, пример представлен на рисунке 1.5. Но стоит отметить, что при использовании UNION не может быть повторяющихся строк (они отображаются, как одна), а при UNION ALL могут быть повторения.

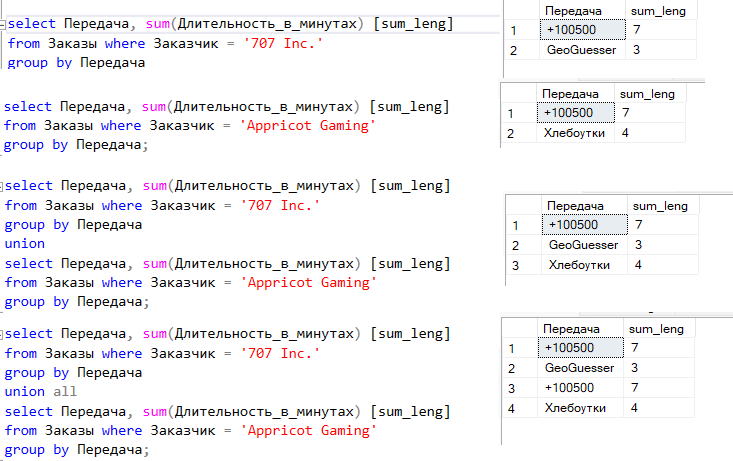


Рисунок 1.5 – UNION и UNION ALL

**Вывод:** В результате выполнения лабораторной работы были получены знания, связанные с группировкой в SELECT-запросах, способы использования а также модификаторы ROLLUP и CUBE. Изучены операции над множествами, исполняемые с помощью операторов UNION, UNION ALL, INTERSECT и EXCEPT.