**Базы данных**

**Отчет по лабораторной работе №4, Нечай-Ницевич Денис ПОИБМС 7-1**

**Цель работы**: Ознакомиться с соединением таблиц с помощью INNER, OUTER и CROSS JOIN и их отличиями, изучить синтаксис конструкции CASE…WHEN, работу функции ISNULL(), выражения is null / is not null.

Для организации многотабличных SELECT-запросов используется соединение таблиц или же JOIN. Они бывают нескольких типов, первым из которых рассмотрим INNER JOIN (внутренне соединение), общая схема представлена на рисунке 1.1.

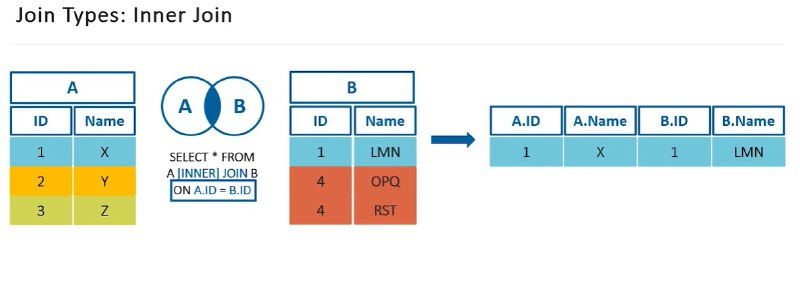


Рисунок 1.1 – Inner Join

Мы указываем две таблицы, которые мы хотим соединить по значению какого-либо столбца. В результирующем наборе мы увидим только те строки, в которых совпадают значения, по которым мы проверяли равенство, или же, пересечение таблиц.

Далее – OUTER JOIN (внешнее соединение). Оно может быть как LEFT – рисунок 1.2, так и RIGHT – рисунок 1.3. Принцип работы следующий: за основу у нас берется левая/правая таблицы и сравнивается со второй по определенному полю. В результирующий набор попадают как и строки с совпадающими значениями, так и строки из главной таблицы, для которых нету совпадений. В этом случае строкам из главной таблицы сопоставляются строки со всеми значениями равными NULL.

Ещё один вид OUTER JOIN’a – FULL OUTER JOIN – Рисунок 1.4. Он объединяет в себе результат LEFT и RIGHT JOIN.

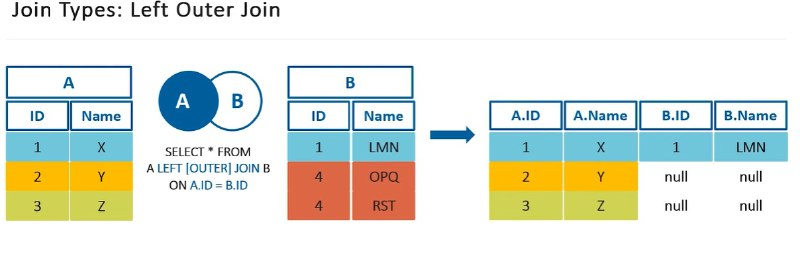
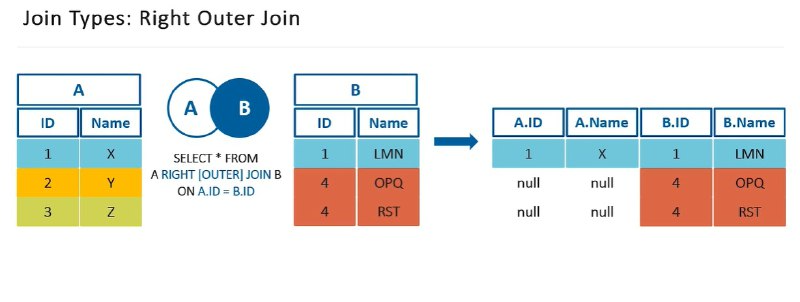


Рисунок 1.2 – Left Outer Join

Рисунок 1.3 – Right Outer Join

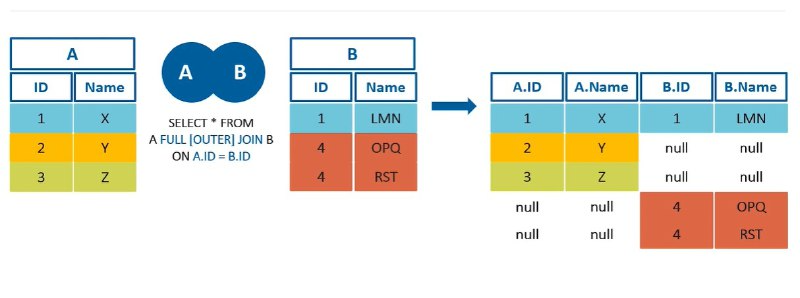


Рисунок 1.4 – FULL OUTER JOIN

Следующий тип соединения – CROSS JOIN – рисунок 1.5. При таком типе соединения каждый столбец из левой таблицы перемножается с каждым столбцом из правой таблицы. Результирующий набор содержит множество строк, но стоит учесть, что если мы добавим поле, по которому будем проводить сравнение, то у нас получится результат, если бы мы использовали INNER JOIN. Таким способом перекрестное соединение использовать не рекомендуется.

Стоит отметить, что с помощью JOIN’ов в одном запросе можно объединить несколько таблиц, если они связаны между собой по определенным столбцам.

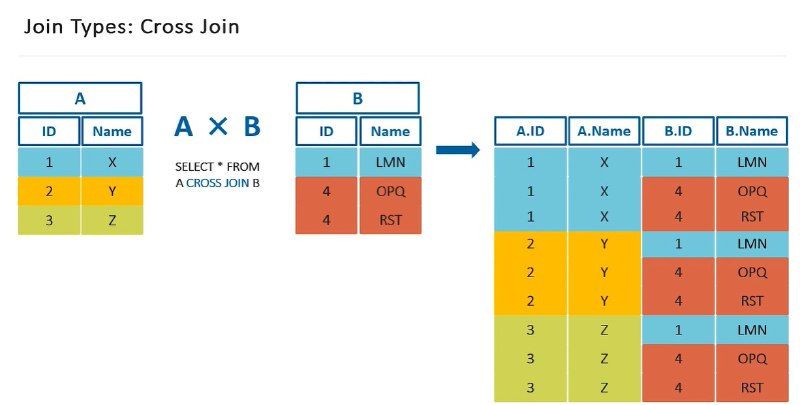


Рисунок 1.5 – Cross Join

Следующий пункт лабораторной работы – использование конструкции CASE…WHEN – рисунок 1.6. После SELECT мы указываем ключевое слово CASE, которое является началом конструкции, далее в WHEN – мы записываем какое-либо условие для нужного нам столбца, и после условия – то, что будет записано в этом столбце при «истинности» этого условия. Если результат – «ложь», выполняется действие из поля ELSE. Ключевое слово END является обязательным, сообщает об окончании конструкции CASE…WHEN. Данную конструкцию можно использовать как при выборке, так и при фильтрации.

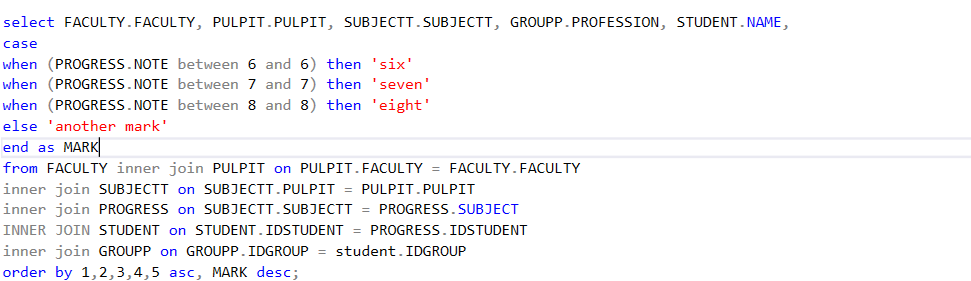


Рисунок 1.6 – CASE…WHEN

Далее мы изучили функцию isnull, которая проверяет, содержится ли в ячейке значение null, и, если содержится, устанавливает в эту ячейку заданное пользователем значение. После функции указывается название столбца.

Следующее – работа с is null и is not null, которые используются при фильтрации. В результирующий набор попадают только те значения, которые удовлетворяют условию.

**Вывод.**

В данной лабораторной работе был получен практические навыки и теоретические знания о соединениях таблиц, а также полезных «инструментах" при написании SELECT-запросов: конструкция CASE…WHEN, функция isnull(), выражения is null и is not null.