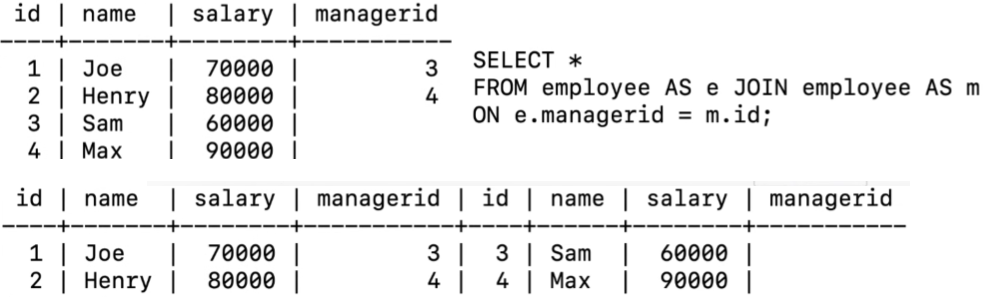
***Типы Join:***

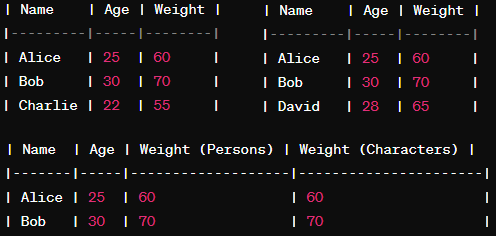
Cross join – декартово произведение соединенных таблиц (синтаксис без условия определения, Cross так же можно опускать).

Self join – Соединение таблицы со своей же копией, для соединения так же используются правила Inner и Outer join.

Можно использовать для анализа сложных структур в рамках одной таблицы.

Inner join – возвращает строки, которые имеют соответствующие значения в обеих таблицах для заданных аргументов. Используется по умолчанию, если написать join и указана секция сравнения.

Запрос по нескольким join выдаст совпадающие значения по двум этим аргументам.



SELECT \* FROM Persons p

JOIN Characters c ON p.Name = c.Name

AND p.Age = c.Age;

Natural join – Работает так же как Inner, автоматически соединяет аргументы, названия которых совпадают. Не рекомендуется к использованию по причине неясности при чтении (непонятно, какие аргументы заджоинены), также в случае, например, переименования столбцов появится новый джоин на основании того же запроса и результат вывода будет отличаться. Лучшей практикой будет явно прописывать все соединения аргументов.

Outer join – помимо строк, имеющих соответствующее значение, так же возвращает строки, которые не имеют соответствий.

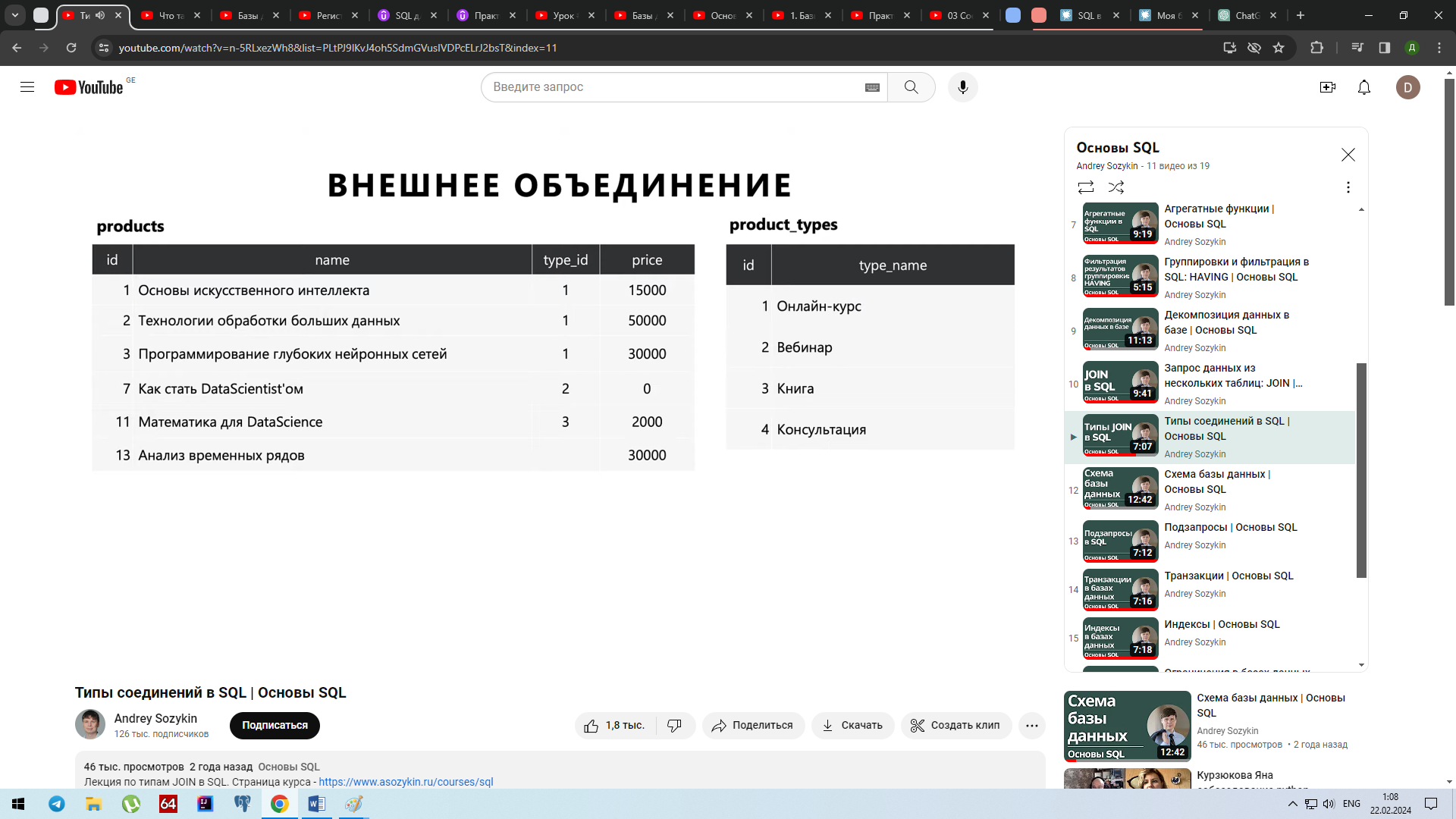
*Подразделяются на:*

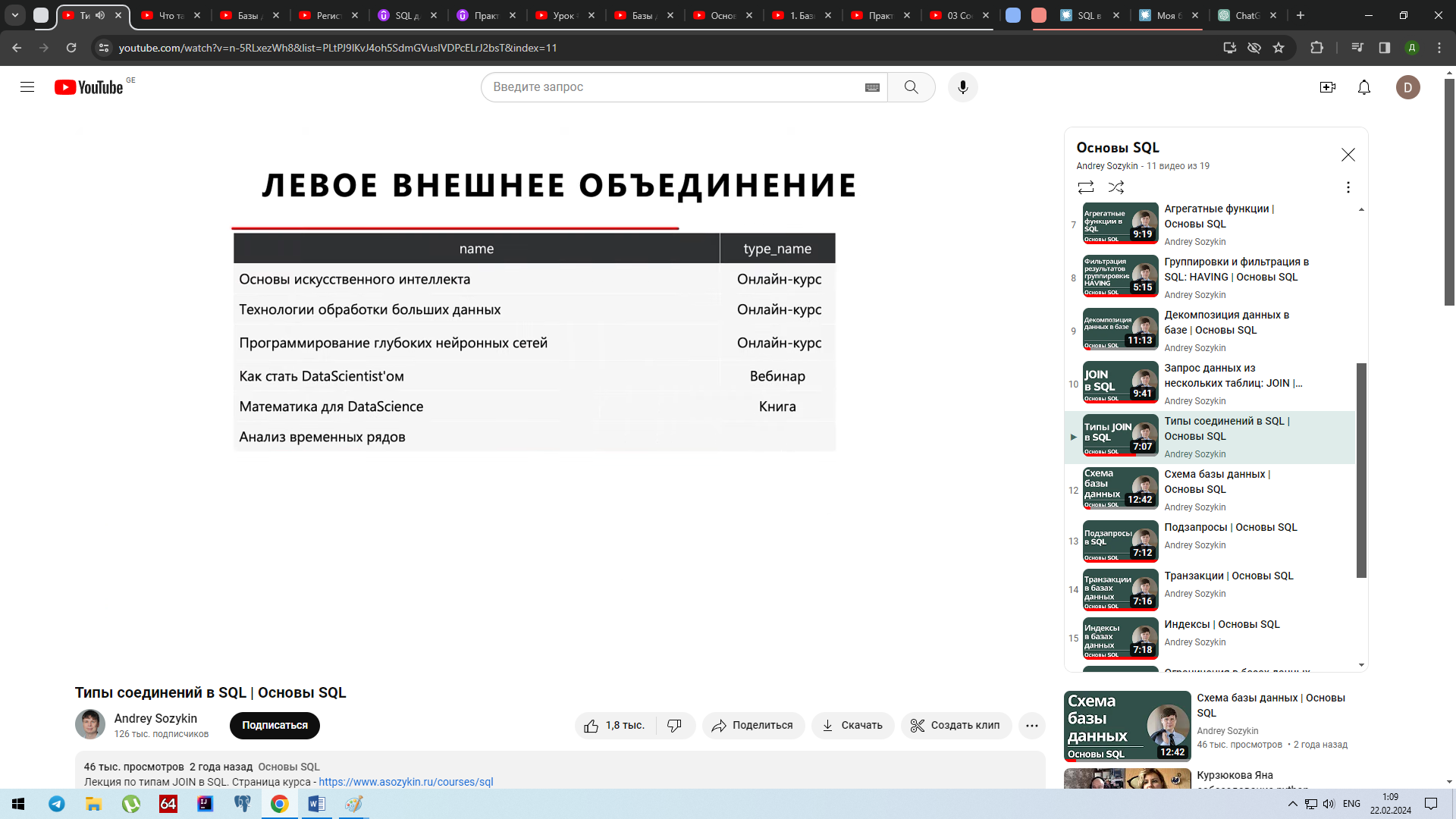
Left join – Возвращает все строки из левой таблицы, строки имеющие соответствия заполняются из правой таблицы, не имеющие заполняются null.

Right join – Возвращаются все строки из правой таблицы, если нет совпадений для них, то возвращаются null.

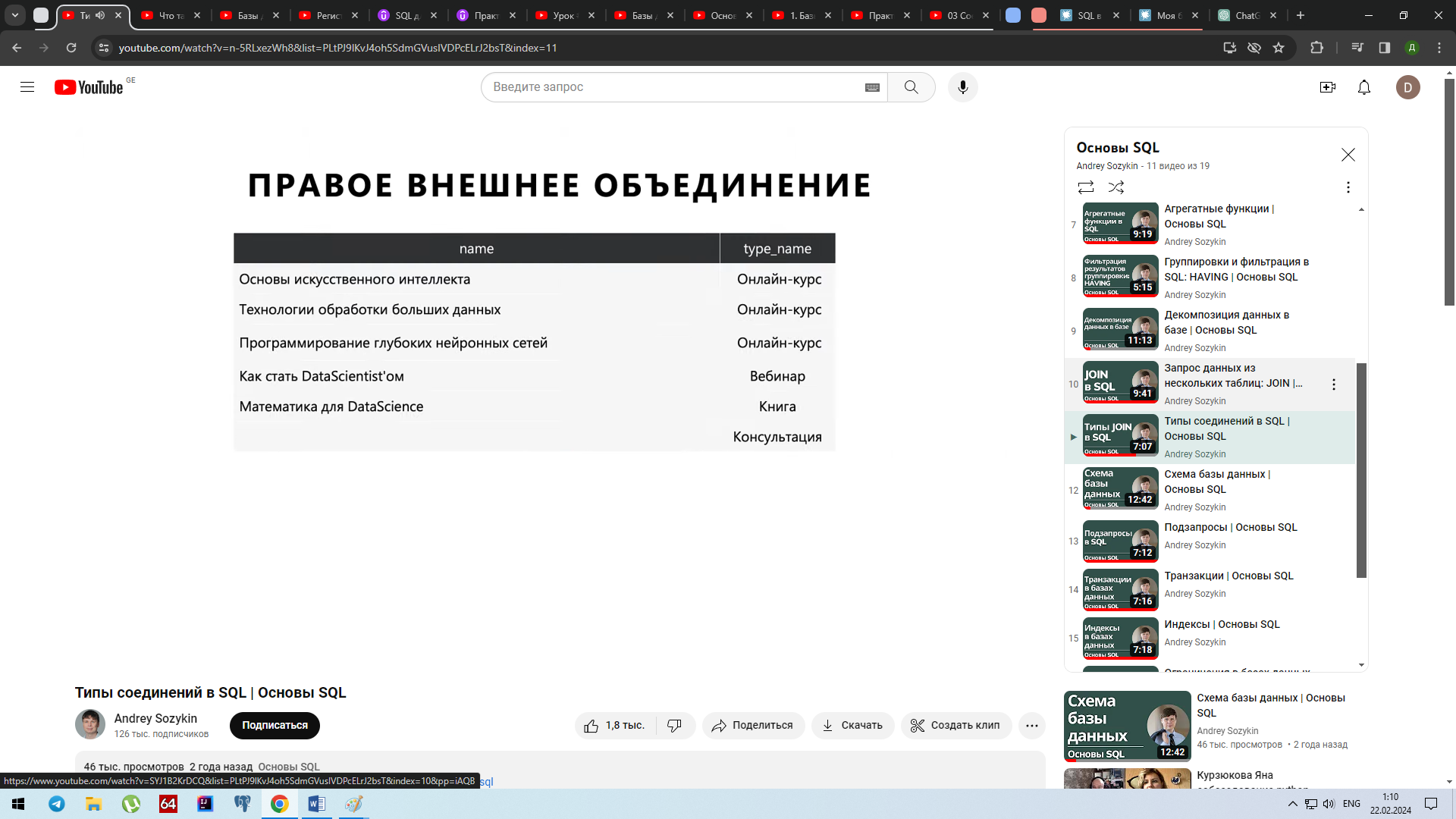
Взаимозаменяемы, по сути можно всегда использовать Left или Right Join, переставляя табилцы в запросе местами.

Пример:

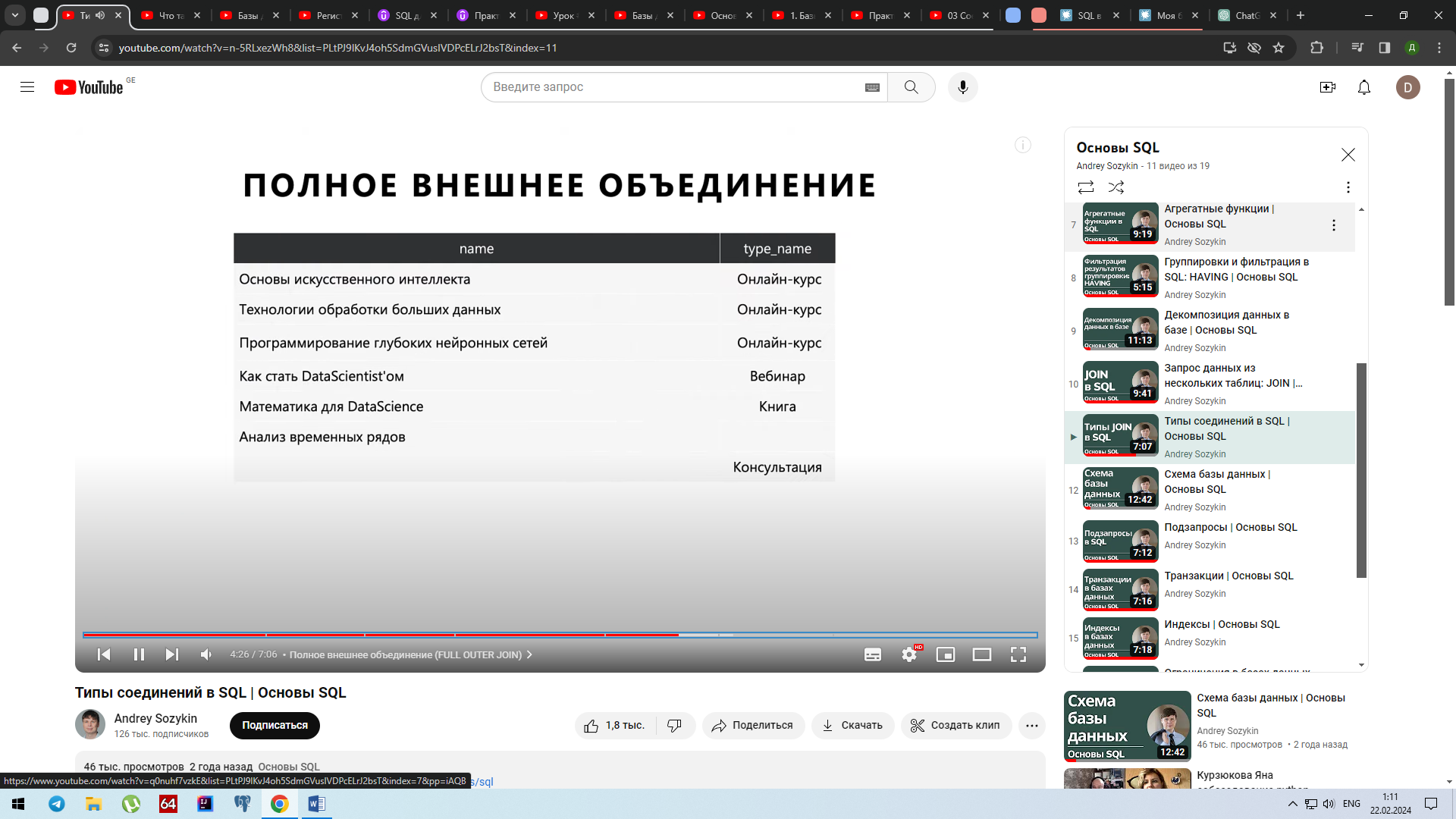
  
Left Join



Right Join



Full join – объединяет в себе Left и Right join;



**Подзапросы –** запрос, вложенный внутри другого запроса.

*Как скалярный операнд:* Результатом запроса будет одно значение подзапроса, с ним используются операторы сравнения ‘>’ ‘<’ ‘=’

*Как нескалярный операнд*: Подзапрос возвращает набор значений, с ним используются in, all, any

Некоррелированный подзапрос – может быть выполнен независимо от основного запроса и результат не зависит от строк, возвращаемых из внешнего запроса. Выполняется первым и только один раз.

Пример: Select \* from table where salary > (select avg(salary) from table) – выбирает строки, где зарплата выше средней. Сначала выполняется подзапрос и это значение сравнивается с каждой строкой внешнего запроса.

Коррелированный подзапрос – зависит от внешнего запроса. Выполняется для каждой строки основного запроса, используя значения из этой строки в качестве параметров для подзапроса.

Пример: Select \*, (select avg(salary) from workers where workers.department\_id = departments.id) from departments – выводит все из таблицы departments и добавляет дополнительный столбец, который выводит среднюю зп для каждого отдела.

Подзапросы можно использовать в секциях from и join как источники данных.

Нельзя изменять данные в таблице, если эти данные используются в подзапросе.

Подзапросы чаще всего можно заменить другими конструкциями (join, group by). В большинстве случаев следует избегать использование подзапросов, т.к. они являются менее производительными (особенно вложенные подзапросы), чем join. Выбирая между join и подзапросом следует обратить внимание на производительность обоих вариантов и читабельность запроса.

В некоторых ситуациях планировщик запросов оптимизирует подзапросы до join, чтобы точнее узнать, каким образом будет исполняться запрос нужно использовать EXPLAIN.