



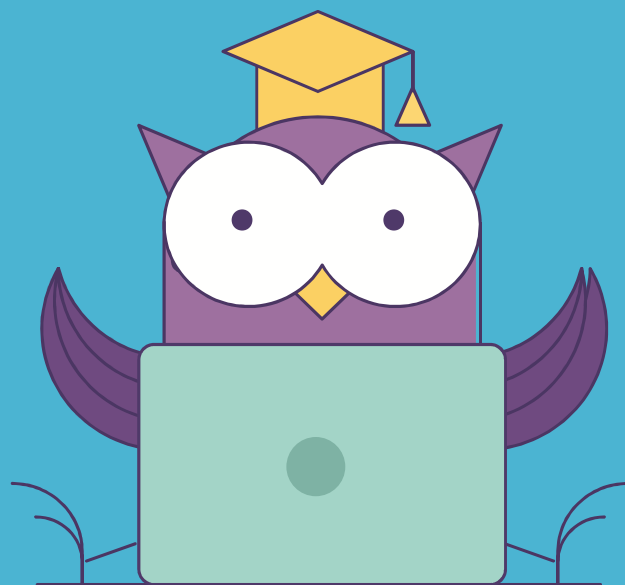
ОНЛАЙН-ОБРАЗОВАНИЕ

Оператор **SELECT** и простые фильтры

Курс «Разработчик MS SQL Server»



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте ☐ + если все хорошо
Ставьте ☐ - если есть проблемы

Цель вебинара

- Понять как работает оператор SELECT
- Использовать простые фильтры в WHERE
- Попробовать разные виды JOIN





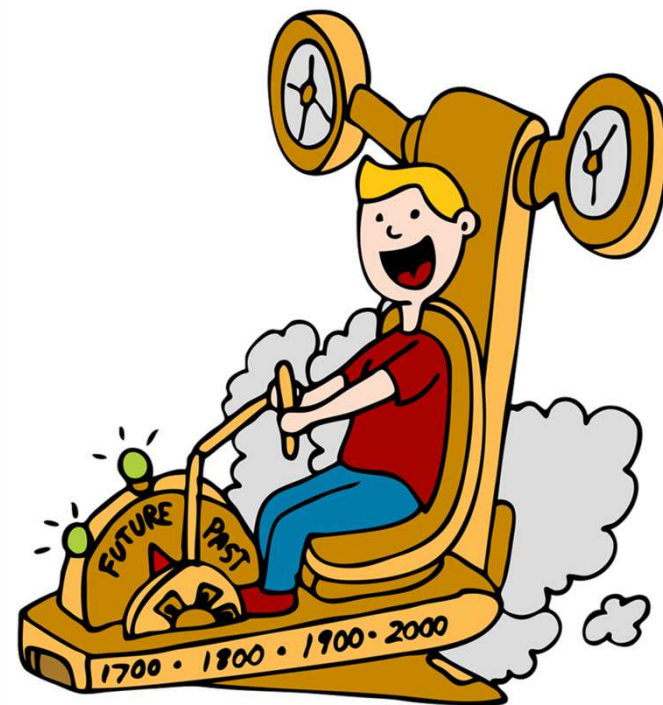
Изначально SEQUEL или Structured English Query Language, потом Structured Query Language

В SQL есть стандарт ANSI, но реализация немного отличается в каждой СУБД.

Transact-SQL (TSQL) - расширение SQL в MS SQL Server

В SQL есть

- DDL Data Definition Language
Create, Alter, Drop
- DML Data Manipulation Language
Insert, Update, Delete
- DCL Data Control Language
Grant, Revoke, Deny
- TCL Transaction Control Language
Begin Transaction, Commit



Select относится к DML

- | | |
|--------------------|--|
| 1. SELECT | какие поля выбираем |
| 2. FROM | откуда – таблица, представление, подзапрос |
| 3. WHERE | как фильтры накладываем |
| 4. GROUP BY | по каким полям строим группы |
| 5. HAVING | фильтры по группам |
| 6. ORDER BY | сортировка |




В каком порядке оптимизатор выполняет инструкцию

1. **FROM**
2. **WHERE**
3. **GROUP BY**
4. **HAVING**
5. **SELECT**
6. **ORDER BY**

01

SELECT из одной таблицы

- Если в именах пробелы – обрамляем [] или " "
- "SELECT *" 
- **FROM** таблица, представление, функция, CTE, временная таблица, подзапрос
- Порядок возвращаемых строк не определен, если нет ORDER BY
- Комментарии `/* */`, `--`

- Часть Select где можно фильтровать строки и проверяя их по предикатам в условии.
- Строки, в которых предикат имеет значение "ложь" или возвращает неизвестное состояние, не возвращаются.

Желательно, чтобы на поля, по которым фильтры были бы индексы.

В Where могут быть условия по полям, подзапросам, функциям...

Можно ли в WHERE использовать алиасы для полей?

- **TRUE / FALSE / UNKNOWN**
- **NULL не имеет типа**
- **NULL может записываться в поля любого типа**
- **Любая операция с NULL дает в результате NULL**
- **Любое сравнение с NULL - дает значение UNKNOWN**
- **Функция IS NULL**
для определения NULL, возвращающая TRUE или FALSE

Используйте скобки, чтобы не запутаться с приоритетами !

Level	Операторы
1	~ (побитовое НЕ)
2	* (умножение), / (деление), % (остаток деления)
3	+ (положительное), - (отрицательное), + (сложение), +(объединение), - (вычитание), & (побитовое И), ^ (побитовое исключающее ИЛИ), (побитовое ИЛИ)
4	=, >, <, >=, <=, <>, !=, !>, !< (операторы сравнения)
5	NOT
6	AND
7	ALL, ANY, BETWEEN, IN, LIKE, OR, SOME
8	= (присваивание)

<https://docs.microsoft.com/ru-ru/sql/t-sql/language-elements/operator-precedence-transact-sql>

- Проверка попадают ли значения в диапазон
- **WHERE x BETWEEN x1 AND x2**
равносильно
x >= x1 AND x <= x2
- **WHERE x NOT BETWEEN x1 AND x2**
равносильно
NOT (x BETWEEN x1 AND x2)

LIKE определяет, совпадает ли указанная символьная строка с заданным шаблоном

Символ-шаблон	Описание
%	Любая строка, содержащая ноль или более символов.
_ (подчеркивание)	Любой одиночный символ.
[]	Любой одиночный символ, содержащийся в диапазоне ([a-f]) или наборе ([abcdef]).
[^]	Любой одиночный символ, не содержащийся в диапазоне ([^a-f]) или наборе ([^abcdef]).

<https://docs.microsoft.com/ru-ru/sql/t-sql/language-elements/like-transact-sql>

02

WHERE

Будет ли выбран null?

Если нет, то как нужно переписать запрос?

```
select *  
from Sales.Orders  
where PickedByPersonID in  
    (4, 3, null)  
order by PickedByPersonID
```

JOIN - Соединения

Suppliers (Поставщики)

SupplierID	SupplierName
1	A Datum Corporation
2	Contoso, Ltd.
3	Consolidated Messenger
9	Nod Publishers

SupplierTransaction (Сделки с поставщиками)

SupplierTransactionID	SupplierID	TransactionDate	TransactionAmount
293115	1	2016-01-04	2194.50
293124	1	2016-01-04	3762.00
293125	1	2016-01-04	-5956.50
293548	1	2016-01-05	7524.00
293973	1	2016-01-06	4639.80
294505	1	2016-01-07	9405.00
295785	1	2016-01-11	-21568.80
134	2	2013-01-02	360.53
1423	2	2013-01-07	-360.53

SELECT <...>

FROM Suppliers s

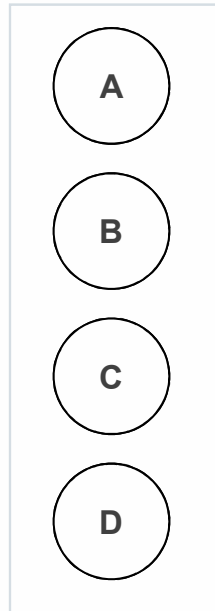
JOIN SupplierTransactions t

ON t.SupplierID = s.SupplierID

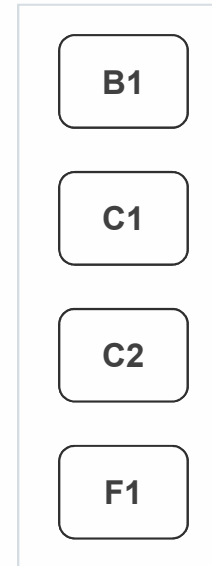
SupplierID	SupplierName	SupplierTransactionID	SupplierID	TransactionDate	TransactionAmount
1	A Datum Corporation	293124	1	2016-01-04	3762.00
1	A Datum Corporation	293125	1	2016-01-04	-5956.50
1	A Datum Corporation	293115	1	2016-01-04	2194.50
1	A Datum Corporation	293548	1	2016-01-05	7524.00
1	A Datum Corporation	293973	1	2016-01-06	4639.80
1	A Datum Corporation	294505	1	2016-01-07	9405.00
1	A Datum Corporation	295785	1	2016-01-11	-21568.80
2	Contoso, Ltd.	134	2	2013-01-02	360.53
2	Contoso, Ltd.	1423	2	2013-01-07	-360.53

CROSS JOIN

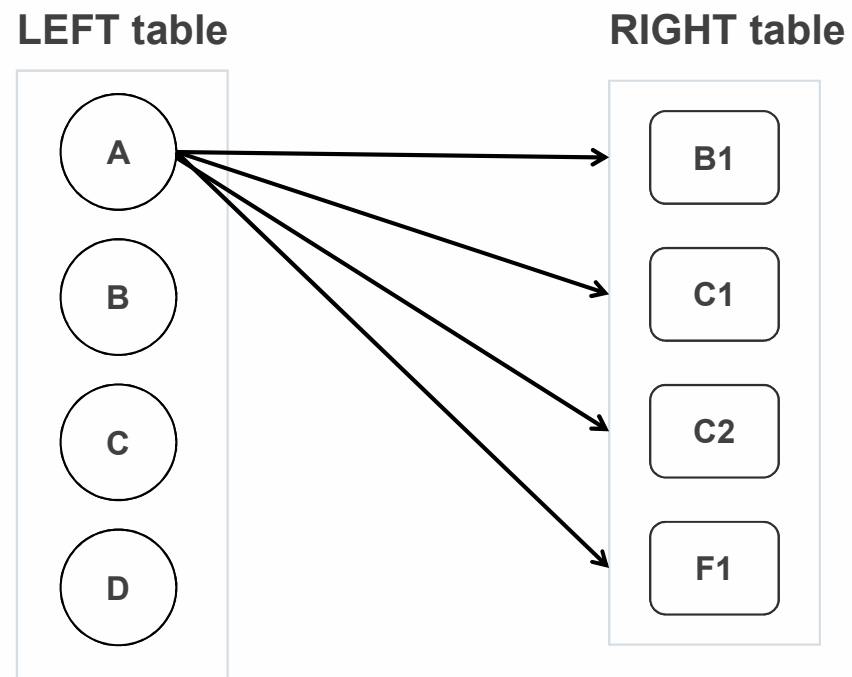
LEFT table



RIGHT table

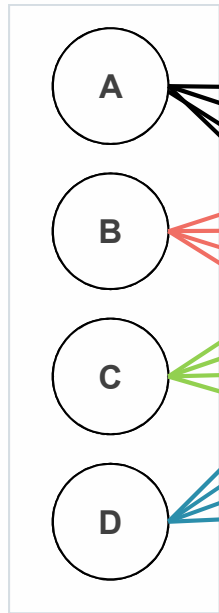


CROSS JOIN

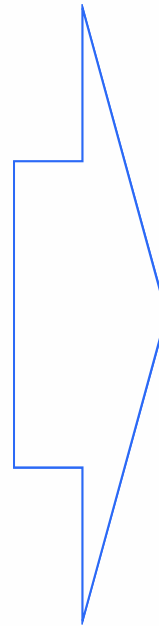
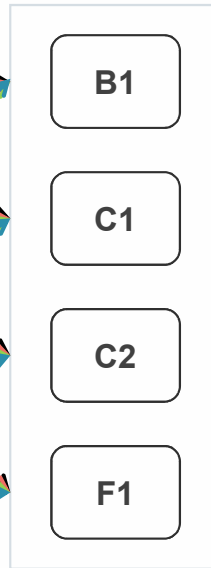


CROSS JOIN

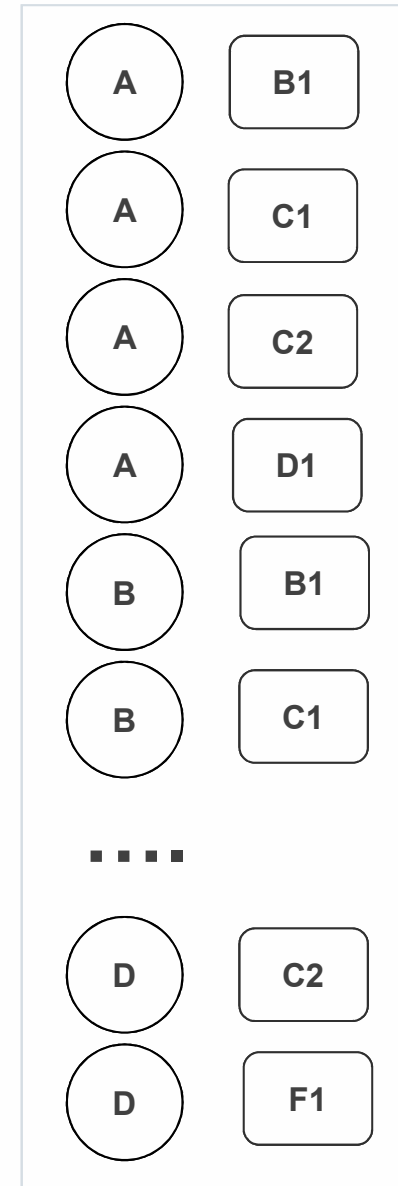
LEFT table



RIGHT table

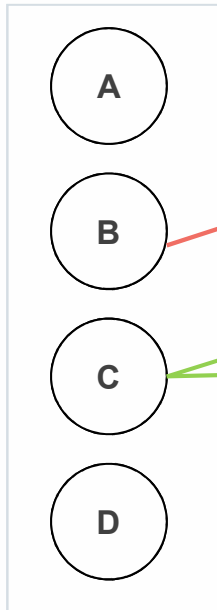


RESULT

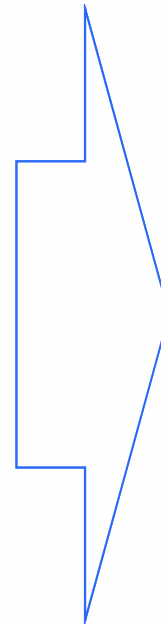
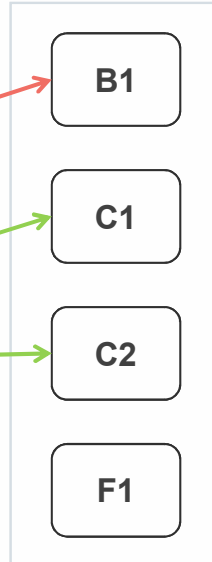


INNER JOIN

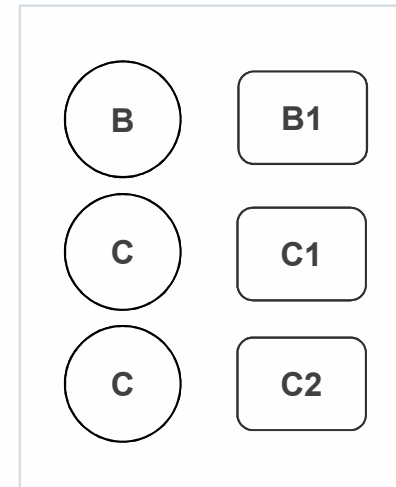
LEFT table



RIGHT table

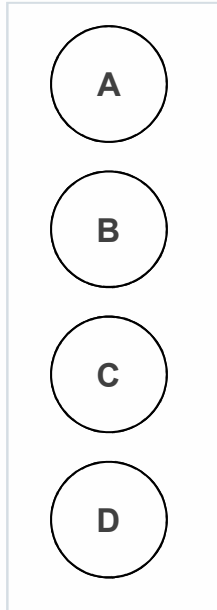


RESULT

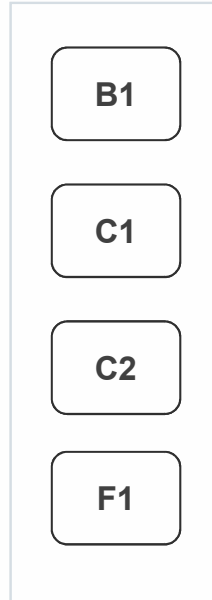


LEFT JOIN

LEFT table

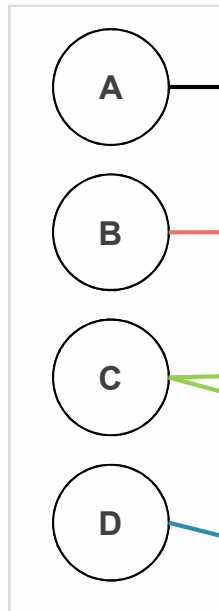


RIGHT table

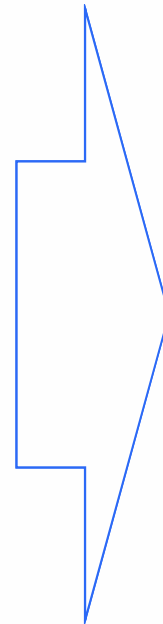
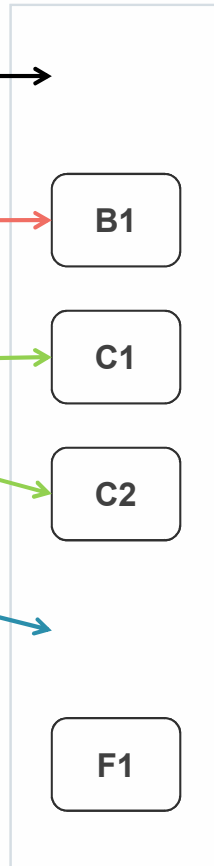


LEFT JOIN

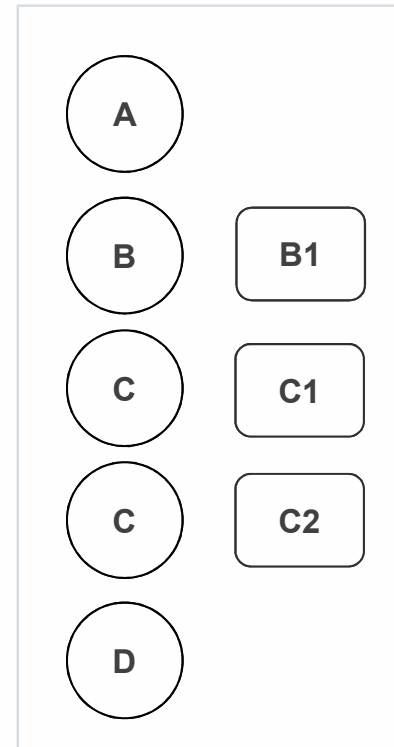
LEFT table



RIGHT table

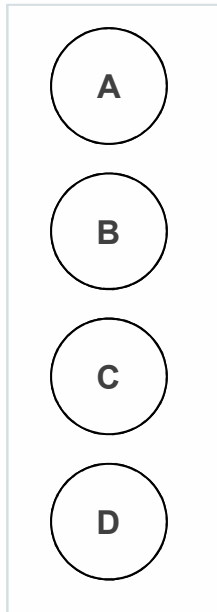


RESULT

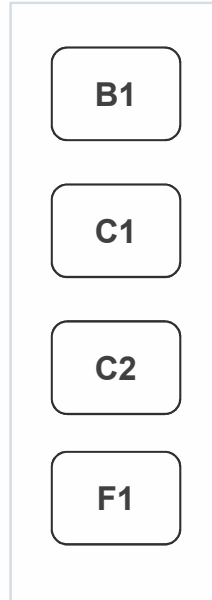


RIGHT JOIN

LEFT table

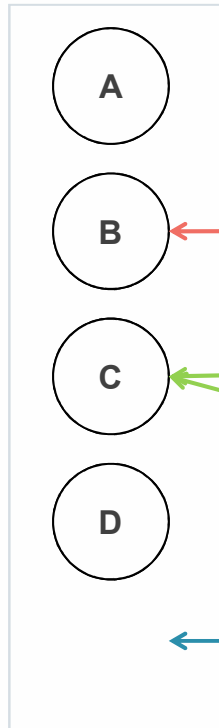


RIGHT table

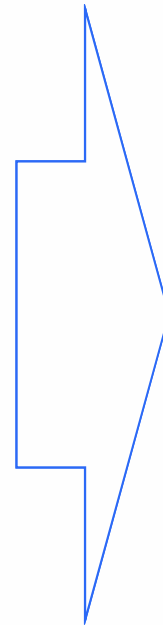
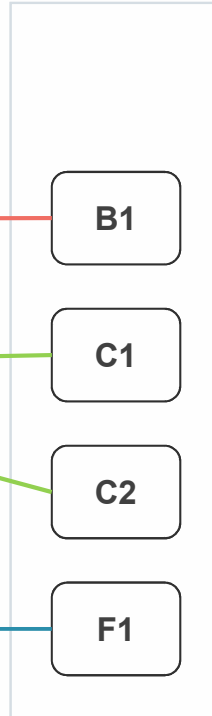


RIGHT JOIN

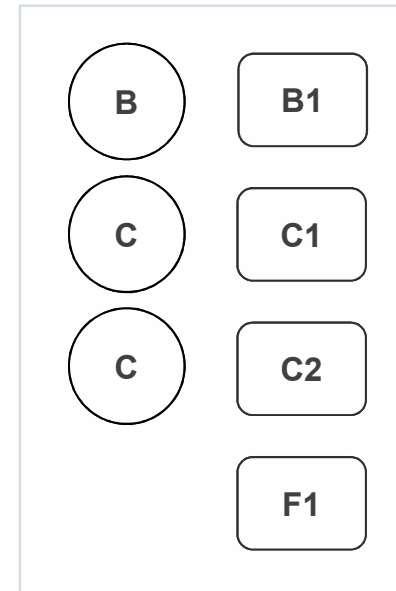
LEFT table

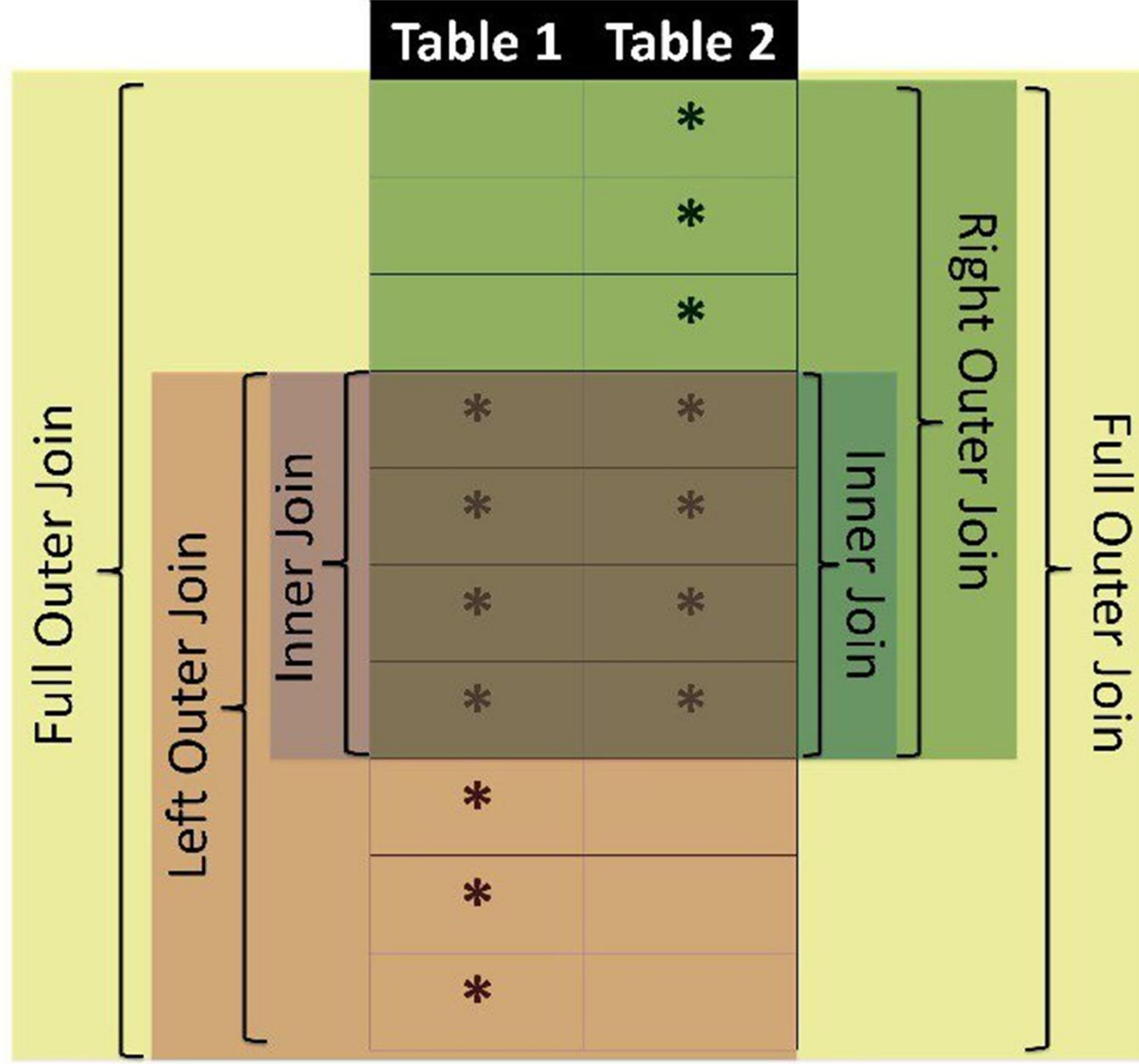


RIGHT table



RESULT





- **CROSS JOIN**

Декартово произведение

```
FROM t1 CROSS JOIN t2
```

- **INNER JOIN**

Декартово произведение + фильтрация

```
FROM t1 [INNER] JOIN t2 on t1.id = t2.id
```

- **LEFT/RIGHT JOIN**

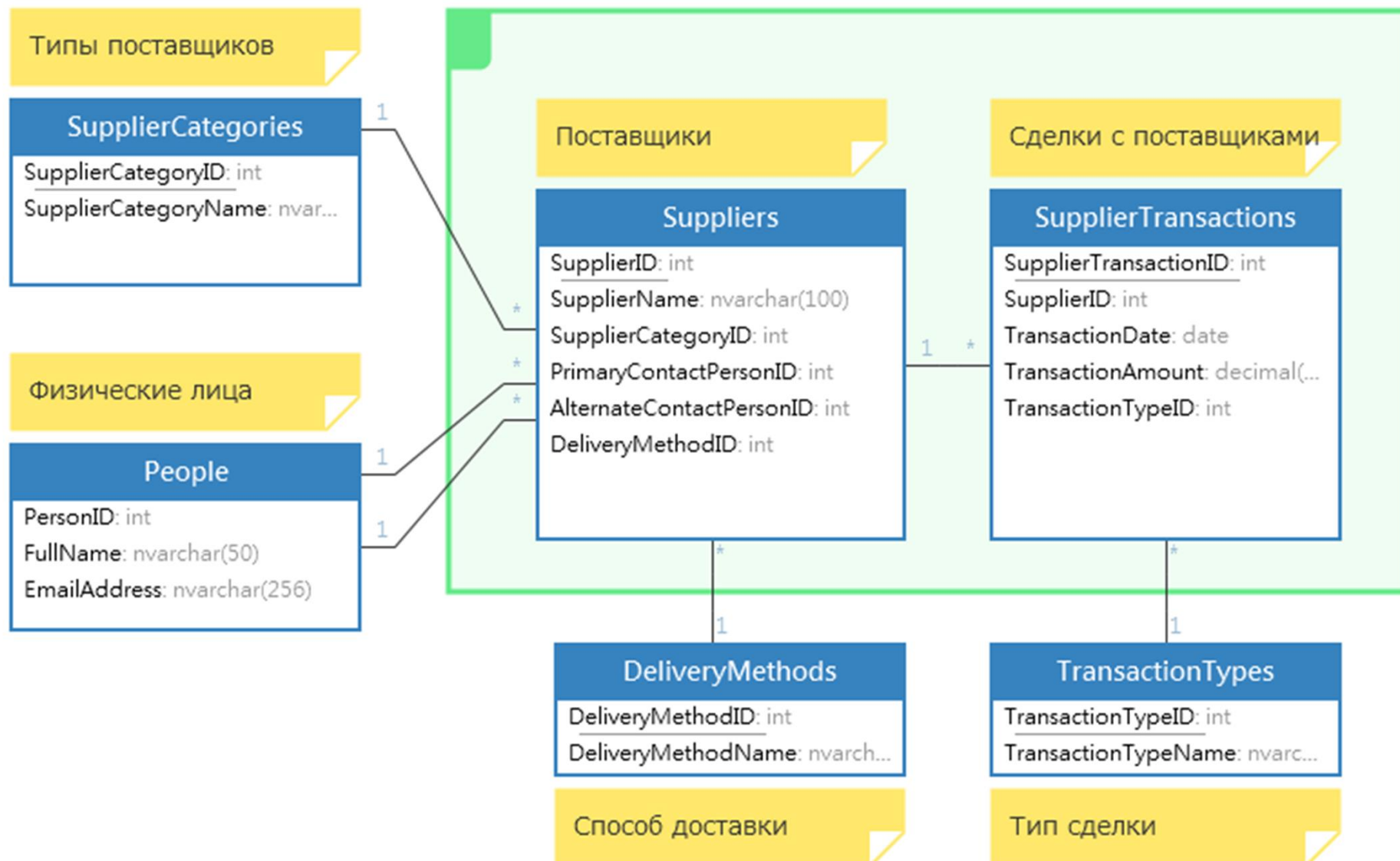
Декартово произведение + фильтрация + внешние строки

```
FROM t1 LEFT [OUTER] JOIN t2 on t1.id = t2.id
```

- **FULL JOIN**

LEFT JOIN + RIGHT JOIN

```
FROM t1 FULL [OUTER] JOIN t2 on t1.id = t2.id
```



03

JOIN

О чем мы говорили сегодня?

- Какую часть SELECT выполняет оптимизатор первой?
- Null и '' одинаковы?
- Какие JOIN операторы вы помните?



Спасибо
за внимание!

До встречи в **Slack** и на вебинаре

