

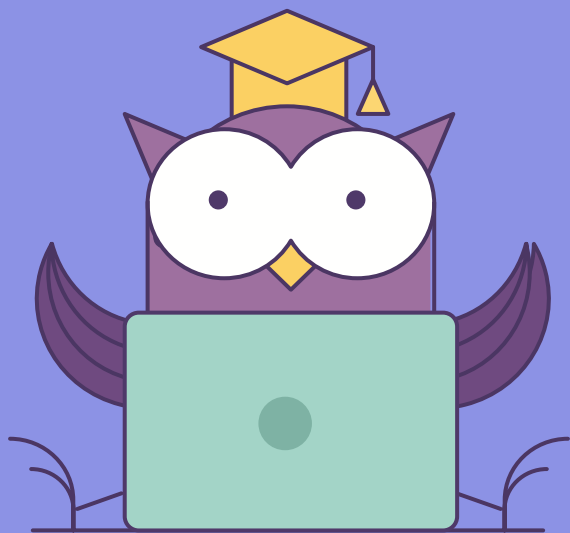


ОНЛАЙН-ОБРАЗОВАНИЕ


Проверить включена  
ли запись?



# Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте  если все хорошо  
Или напишите, какие есть проблемы

# Оператор SELECT и простые фильтры, JOIN

Курс “MS SQL Server Developer”  
Разработчик MS SQL Server

Группа 2023-01



- Что из не реляционных возможностей SQL Server запомнилось с прошлого занятия?



- Что из не реляционных возможностей SQL Server запомнилось с прошлого занятия?
- Сколько экземпляров SQL Server можно установить на одном сервере?

<https://docs.microsoft.com/ru-ru/sql/sql-server/maximum-capacity-specifications-for-sql-server>



- Что из не реляционных возможностей SQL Server запомнилось с прошлого занятия?
- Сколько экземпляров SQL Server можно установить на одном сервере?

<https://docs.microsoft.com/ru-ru/sql/sql-server/maximum-capacity-specifications-for-sql-server>

- На каких портах они будут?



- Что из не реляционных возможностей SQL Server запомнилось с прошлого занятия?
- Сколько экземпляров SQL Server можно установить на одном сервере?  
<https://docs.microsoft.com/ru-ru/sql/sql-server/maximum-capacity-specifications-for-sql-server>
- На каких портах они будут?
- Как обратиться к инстансу **DEV** на сервере **SQL-SRV** (как указать в строке подключения)?





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

**БД WideWorldImporters**

**Wide World Importers (WWI)** - оптовый импортер и дистрибьютор товаров (магазин).

StockItemID	StockItemName
1	USB missile launcher (Green)
2	USB rocket launcher (Gray)
3	Office cube periscope (Black)
4	USB food flash drive - sushi roll
5	USB food flash drive - hamburger
6	USB food flash drive - hot dog
7	USB food flash drive - pizza slice
8	USB food flash drive - dim sum 10 drive variety pack
9	USB food flash drive - banana
10	USB food flash drive - chocolate bar
11	USB food flash drive - cookie
12	USB food flash drive - donut
















- +  Application.Cities (System-Versioned)
- +  Application.Countries (System-Versioned)
- +  Application.DeliveryMethods (System-Versioned)
- +  Application.PaymentMethods (System-Versioned)
- +  Application.People (System-Versioned)
- +  Application.StateProvinces (System-Versioned)
- +  Application.SystemParameters
- +  Application.TransactionTypes (System-Versioned)

Схема **Application** — в основном справочники и сотрудники

- Города, страны, штаты
- Способы доставки
- Способы оплаты
- Типы транзакций






## Схема **Sales** — продажи

- Заказы (Order)
- Накладные (Invoice)
- и т.д.

- +  Sales.BuyingGroups (System-Versioned)
- +  Sales.CustomerCategories (System-Versioned)
- +  Sales.Customers (System-Versioned)
- +  Sales.CustomerTransactions
- +  Sales.InvoiceLines
- +  Sales.Invoices
- +  Sales.OrderLines
- +  Sales.Orders
- +  Sales.SpecialDeals

## Схема **Purchasing** — закупки

- Поставщики (Supplier)
- Заказы на закупки (Purchase Orders)
- и т.д.

- +  Purchasing.PurchaseOrderLines
- +  Purchasing.PurchaseOrders
- +  Purchasing.SupplierCategories (System-Versioned)
- +  Purchasing.Suppliers (System-Versioned)
- +  Purchasing.SupplierTransactions

## Схема Warehouse — склад

- Какие есть товары в наличии
- и т.д.

- + Warehouse.ColdRoomTemperatures (System-Versioned)
- + Warehouse.Colors (System-Versioned)
- + Warehouse.PackageTypes (System-Versioned)
- + Warehouse.StockGroups (System-Versioned)
- + Warehouse.StockItemHoldings
- + Warehouse.StockItems (System-Versioned)
- + Warehouse.StockItemStockGroups
- + Warehouse.StockItemTransactions
- + Warehouse.VehicleTemperatures

StockItemID	StockItemName	SupplierID	ColorID	UnitPackageID	OuterPackageID	Brand	Size	LeadTimeDays	QuantityPerOuter
1	USB missile launcher (Green)	12	NULL	7	7	NULL	NULL	14	1
2	USB rocket launcher (Gray)	12	12	7	7	NULL	NULL	14	1
3	Office cube periscope (Black)	12	3	7	6	NULL	NULL	14	10
4	USB food flash drive - sushi roll	12	NULL	7	7	NULL	NULL	14	1
5	USB food flash drive - hamburger	12	NULL	7	7	NULL	NULL	14	1
6	USB food flash drive - hot dog	12	NULL	7	7	NULL	NULL	14	1
7	USB food flash drive - pizza slice	12	NULL	7	7	NULL	NULL	14	1

# 01

## SELECT из одной таблицы

- Изначально SEQUEL или Structured English Query Language, потом Structured Query Language (язык структурированных запросов)
- Декларативный
- На SQL есть стандарт ANSI, но реализация немного отличается в каждой СУБД.
- Transact-SQL (T-SQL) — расширение SQL в MS SQL Server
  - Хранимые процедуры, циклы (курсоры) и т.д.
- В SQL есть подмножества
  - DDL — Data Definition Language  
*Create, Alter, Drop*
  - DML — Data Manipulation Language  
*Insert, Update, Delete, Select*
  - DCL — Data Control Language  
*Grant, Revoke, Deny*
  - TCL — Transaction Control Language  
*Begin Transaction, Commit*



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

какие поля выбираем

откуда – таблица, представление, подзапрос

какие фильтры накладываем

по каким полям строим группы

фильтры по группам

сортировка

```
SELECT OrderDateKey, SUM(SalesAmount) AS TotalSales  
FROM FactInternetSales  
GROUP BY OrderDateKey  
HAVING OrderDateKey > 20010000  
ORDER BY OrderDateKey;
```

Основа  
основ



Напишите в чат, как вы думаете в  
каком порядке выполняется  
SELECT?



# В каком порядке выполняется оператор SELECT

В каком порядке оптимизатор выполняет инструкцию

Правильно:

**FROM**  
**WHERE**  
**GROUP BY**  
**HAVING**  
**SELECT**  
**ORDER BY**

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

```
(5) SELECT (5-2) DISTINCT (5-3) TOP(<top_specification>) (5-1) <select_list>
(1) FROM (1-J) <left_table> <join_type> JOIN <right_table> ON <on_predicate>
    | (1-A) <left_table> <apply_type> APPLY <right_table_expression> AS <alias>
    | (1-P) <left_table> PIVOT(<pivot_specification>) AS <alias>
    | (1-U) <left_table> UNPIVOT(<unpivot_specification>) AS <alias>
(2) WHERE <where_predicate>
(3) GROUP BY <group_by_specification>
(4) HAVING <having_predicate>
(6) ORDER BY <order_by_list>;
```

## Выполняются ли запросы?


```
SELECT OrderLineID as [Order Line ID],  
       Quantity,  
       UnitPrice,  
       (Quantity * UnitPrice) AS [TotalCost]  
FROM Sales.OrderLines  
WHERE [TotalCost] > 1000
```

```
SELECT OrderLineID as [Order Line ID],  
       Quantity,  
       UnitPrice,  
       (Quantity * UnitPrice) AS [TotalCost]  
FROM Sales.OrderLines  
ORDER BY [TotalCost]
```

**FROM  
WHERE  
GROUP BY  
HAVING  
SELECT  
ORDER BY**

# ДЕМО SELECT



- Если в именах пробелы, то можно использовать [ ] или " "
- "SELECT \*" 
- FROM — таблица, представление, функция, CTE, временная таблица, подзапрос
- Порядок возвращаемых строк не определен, если нет ORDER BY

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

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

- $>$ ,  $<$ ,  $=$ ,  $\neq$ ,  $<>$
- AND, OR, NOT
- BETWEEN, LIKE
- ...

- 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
- Существует специальная функция для определения NULL, возвращающая TRUE или FALSE (IS NULL)

- 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)**



# ДЕМО WHERE



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]).
ESCAPE	Экранирование спец символов  LIKE ' %100!%% ' ESCAPE ' ! ' LIKE ' DownA_ % ' ESCAPE ' A '

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)

- WHERE ... IS NULL, WHERE ... IS NOT NULL
- Функции ISNULL(), COALESCE()
- Если на столбец можно сделать ограничение NOT NULL, то лучше его сделать
- SET ANSI\_NULLS { ON | OFF }
- SET CONCAT\_NULL\_YIELDS\_NULL { ON | OFF }

- GETDATE, GETUTCDATE
- DAY, MONTH, YEAR, DATEPART
- DATEDIFF
- DATEADD, EOMONTH

- Функции CONVERT(), FORMAT()

Отображение дат с заданием формата отображения

<https://docs.microsoft.com/ru-RU/sql/t-sql/functions/cast-and-convert-transact-sql>

<https://docs.microsoft.com/ru-ru/sql/t-sql/functions/format-transact-sql>

'20170504' будет четвертым мая 2017 года при любых настройках языка

'2017-05-04T00:00:00.000' будет полночью четвертого мая 2017 года при любых настройках языка

2014-05-02 - однозначен только для **date**, для **datetime** только без разделителей

[https://docs.microsoft.com/ru-ru/previous-versions/sql/sql-server-2008/ms180878\(v=sql.100\)#UnseparatedStringFormat](https://docs.microsoft.com/ru-ru/previous-versions/sql/sql-server-2008/ms180878(v=sql.100)#UnseparatedStringFormat)

- Collation (параметры сортировки)

**Делаем перерыв 5 мин?**

Напишите в чат:

- + — нужен перерыв
- - — перерыв не нужен



**02**

**JOIN**



# JOIN - Соединения

**Suppliers (Поставщики)**

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

**Первичный ключ**  
Primary Key  
PK

**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	2016-01-07	9405.00
134	2	2013-01-01	
1423	2	2013-01-01	

**Внешний ключ**  
Foreign Key  
FK

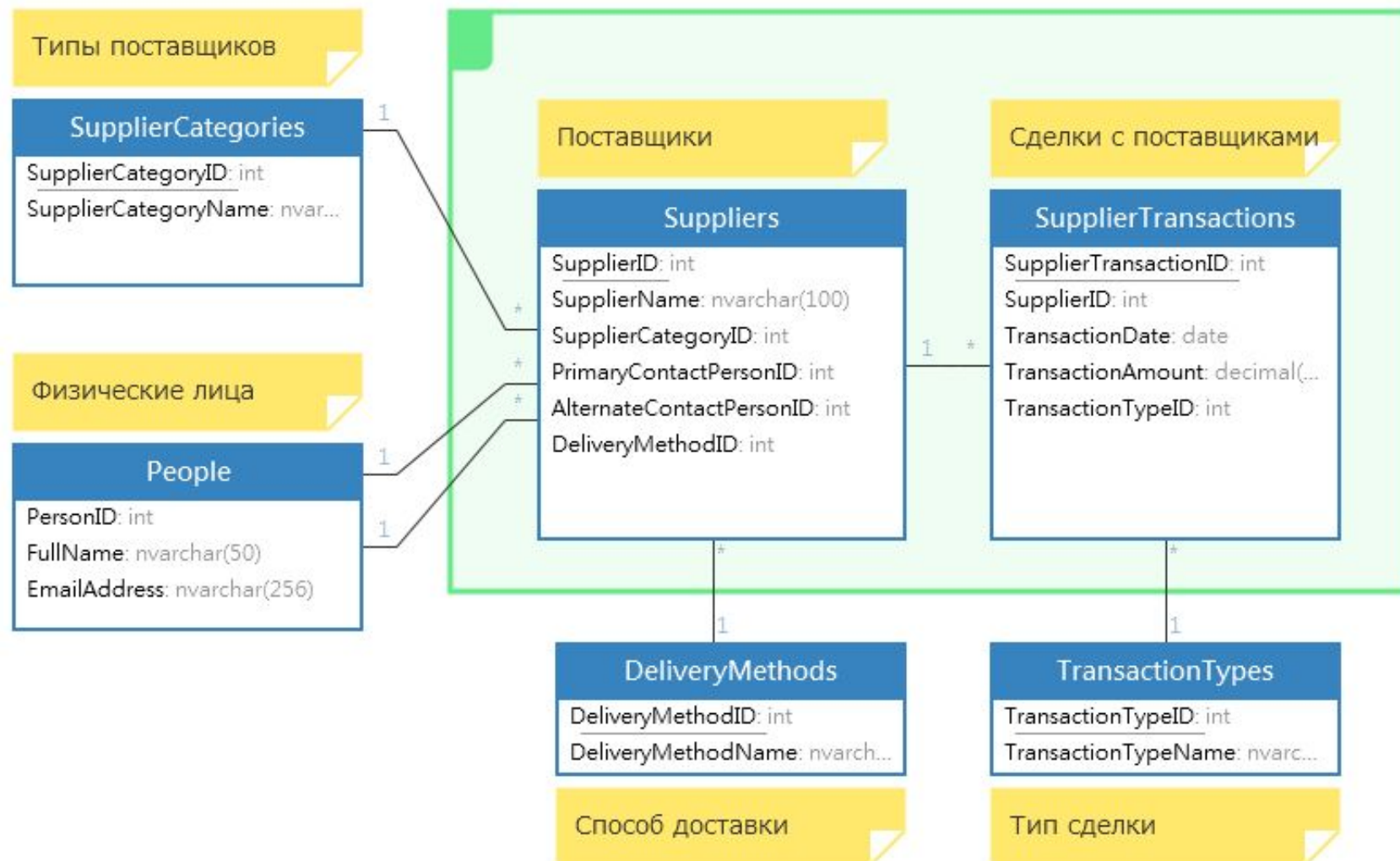
SupplierID	SupplierName	SupplierTransactionID	SupplierID	TransactionDate	TransactionAmount
1	A Datum Corporation	293124	1	2016-01-04	3762.00
1	A Datum Corporation	293125	1	2016-01-04	-5956.50
1	A Datum Corporation	293115	1	2016-01-04	2194.50
1	A Datum Corporation	293548	1	2016-01-05	7524.00
1	A Datum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

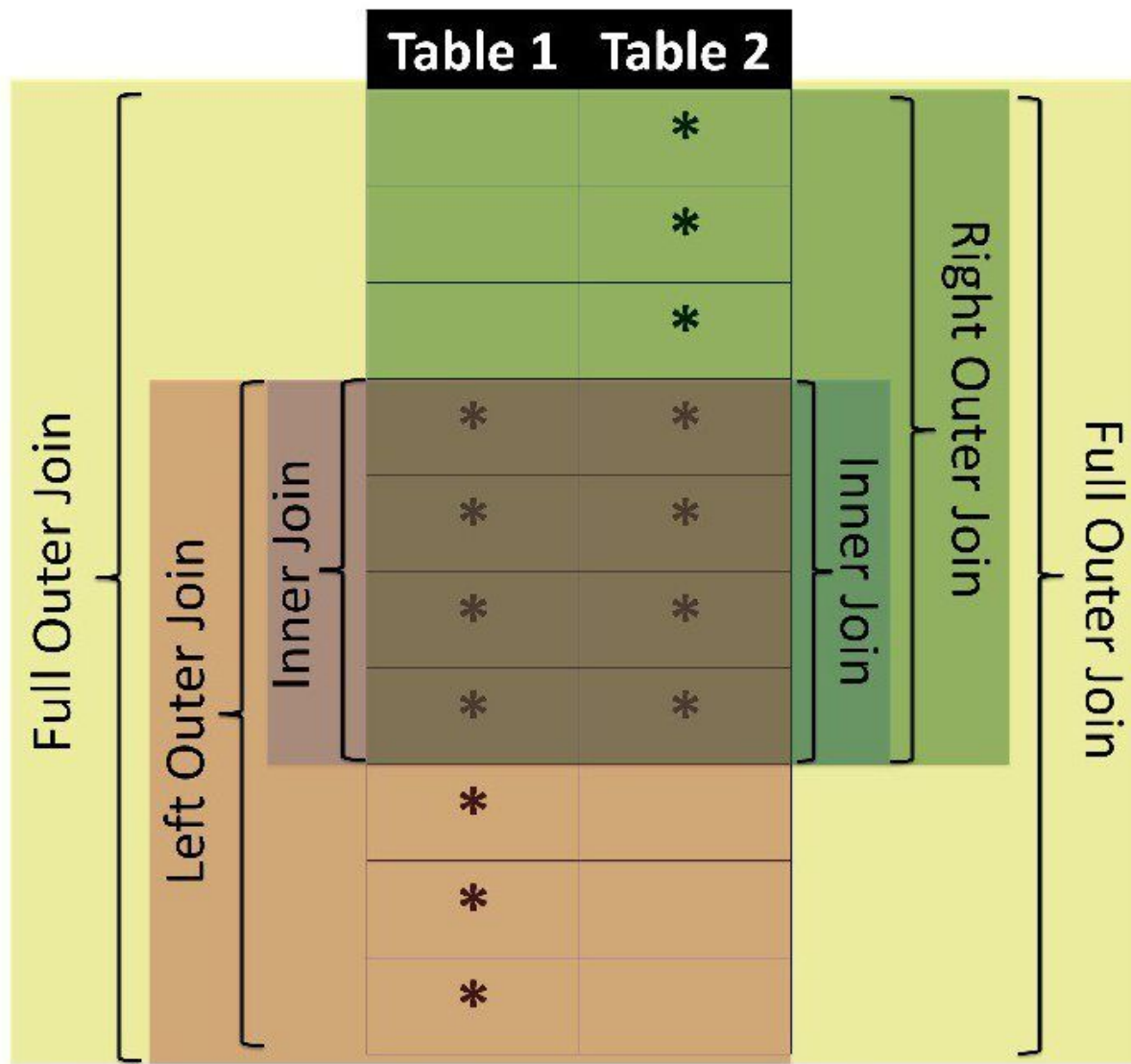
```
SELECT ...  
FROM Suppliers s  
JOIN SupplierTransactions t  
ON t.SupplierID = s.SupplierID
```

# ДЕМО JOIN



# Таблицы БД WWI, используемые в примерах





- **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
```

По каким видам колонок можно делать JOIN?

*Есть ли какие-то ограничения? Только FK, PK, только int?*

**03**

**UNION**

# UNION

ОТЧЕТ

“вертикальное соединение”

```
SELECT ...  
UNION [ALL] | INTERSECT | EXCEPT  
SELECT ...  
[ORDER BY ...]
```

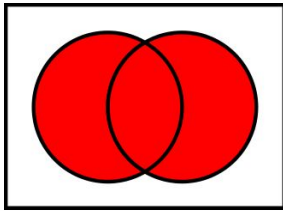
Табл. 1

ID	Name
1	User1
2	User2

Табл. 2

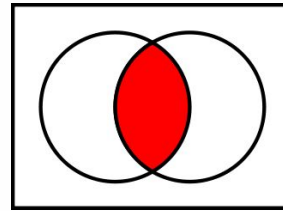
ID	Name
2	User2
3	User3
4	User4

UNION, UNION ALL



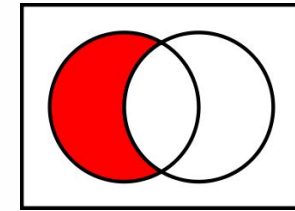
ID	Name
1	User1
2	User2
3	User3
4	User4

INTERSECT



ID	Name
2	User2

EXCEPT



ID	Name
3	User3
4	User4



Написать запрос, чтобы получить

	Col1
1	a
2	b
3	c

# ДЕМО UNION



- Одинаковое количество столбцов
- Совместимость столбцов по типам
- Имена столбцов берутся из первого запроса
- Значения NULL являются тождественными
- Когда можно использовать UNION ALL, то используйте его вместо UNION
- Приоритет - INTERSECT выполняется первым, UNION и EXCEPT запускаются в порядке своего размещения внутри запроса

# 04

## Домашнее задание

## Материалы



**homework\_select.sql**

[Скачать](#)



**Примеры GitHub - OTUS-MS-SQL-Server/ms-sql-server-2022-09:**  
Репозиторий со скриптами для курса MS SQL Server 2022 09  
Ссылка, 24.09.2022



**homework\_select.sql**  
Файл, 24.09.2022



**Ошибки при работе с датой и временем в SQL Server**  
Ссылка, 25.05.2022

```
/*
1. Все товары, в названии которых есть "urgent" или название начинается с "Animal".
Вывести: ИД товара (StockItemID), наименование товара (StockItemName).
Таблицы: Warehouse.StockItems.
*/

напишите здесь свое решение

/*
2. Поставщиков (Suppliers), у которых не было сделано ни одного заказа (PurchaseOrders).
Сделать через JOIN, с подзапросом задание принято не будет.
Вывести: ИД поставщика (SupplierID), наименование поставщика (SupplierName).
Таблицы: Purchasing.Suppliers, Purchasing.PurchaseOrders.
По каким колонкам делать JOIN подумайте самостоятельно.
*/

напишите здесь свое решение
```

Или здесь

<https://github.com/OTUS-MS-SQL-Server/ms-sql-server-2023-01/tree/main/02-select/homework>

**The End**

- Какую часть SELECT выполняет оптимизатор первой?
- Идентичны ли null и " (пустая строка)?
- Какие виды JOIN вы помните?
- Что запомнилось?
- Узнали ли что-то новое?



**Заполните,  
пожалуйста,  
опрос в ЛК о занятии**

