

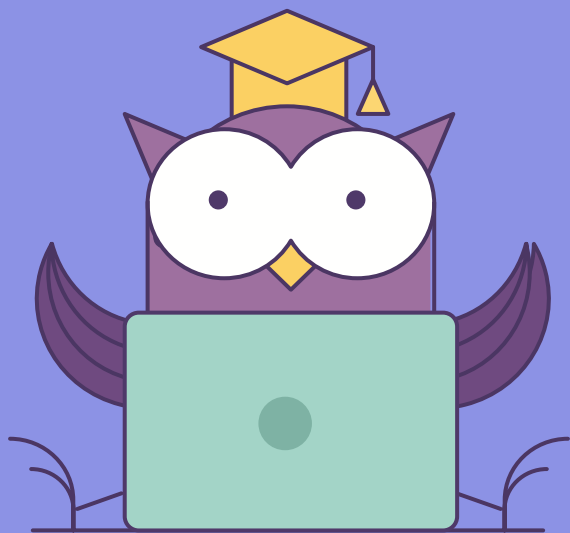


ОНЛАЙН-ОБРАЗОВАНИЕ


Проверить включена
ли запись?



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте  если все хорошо
Или напишите, какие есть проблемы

- Что из не реляционных возможностей SQL Server запомнилось с прошлого занятия?
- Сколько экземпляров SQL Server можно установить на одном сервере?
- На каких портах они будут?
- Как обратиться к инстансу **DEV** на сервере **SQL-SRV** (как указать в строке подключения)?



Оператор SELECT и простые фильтры, GROUP BY, HAVING

Курс “MS SQL Server Developer”
Разработчик MS SQL Server

Группа 2021-03



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

- Агрегатные функции
- GROUP BY, HAVING
- ROLLUP, CUBE
- GROUPING SETS

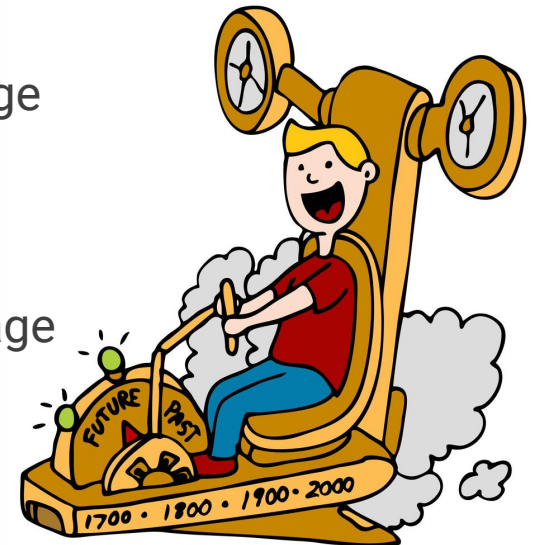
Подзапросов, аналитических функций, оконных функций и т.п. сегодня не будет — они будут дальше.

01

SELECT из одной таблицы



- Изначально SEQUEL или Structured English Query Language, потом Structured Query Language
- Декларативный
- На SQL есть стандарт ANSI, но реализация немного отличается в каждой СУБД.
- Transact-SQL (TSQL) — 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
 - 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 | какие фильтры накладываем |
| 4. GROUP BY | по каким полям строим группы |
| 5. HAVING | фильтры по группам |
| 6. ORDER BY | сортировка |

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>;
```


Выполняться 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]
```

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 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...

- >, <, =, !=, <>
- AND, OR, NOT
- BETWEEN, LIKE
- ...

- 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)

- 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)

ДЕМО WHERE



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>

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'

- 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()

Про 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 2-3 мин?

Напишите в чат:

- + — нужен перерыв
- - — перерыв не нужен



02

JOIN

Внешний ключ
Foreign Key
FK

Первичный ключ
Primary Key
PK

ID	Name	Department_id
1	Иванов	3
2	Петров	5
3	Сидоров	5

ID	Name	Manager_id	Parent_id
1	Подразделение 1	NULL	NULL
2	Подразделение 2	NULL	NULL
3	Подразделение 3	1	NULL
4	Подразделение 4	NULL	2
5	Подразделение 5	2	1
6	Подразделение 6	NULL	NULL

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**

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**

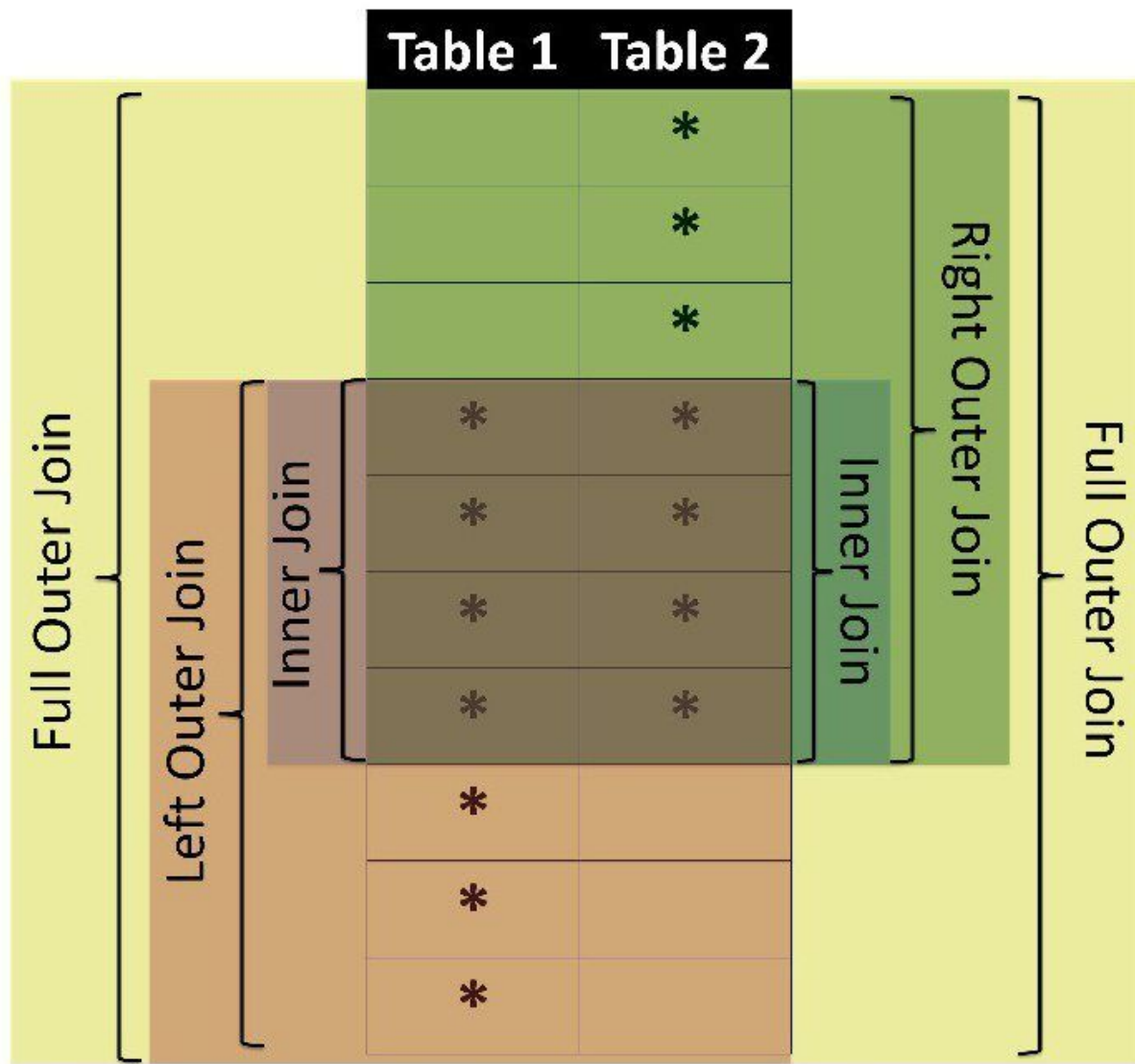
Декартово произведение + фильтрация + внешние строки

```
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
```



- + 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
- Накладные
- 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
- Заказы на закупки
- и т.д.






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

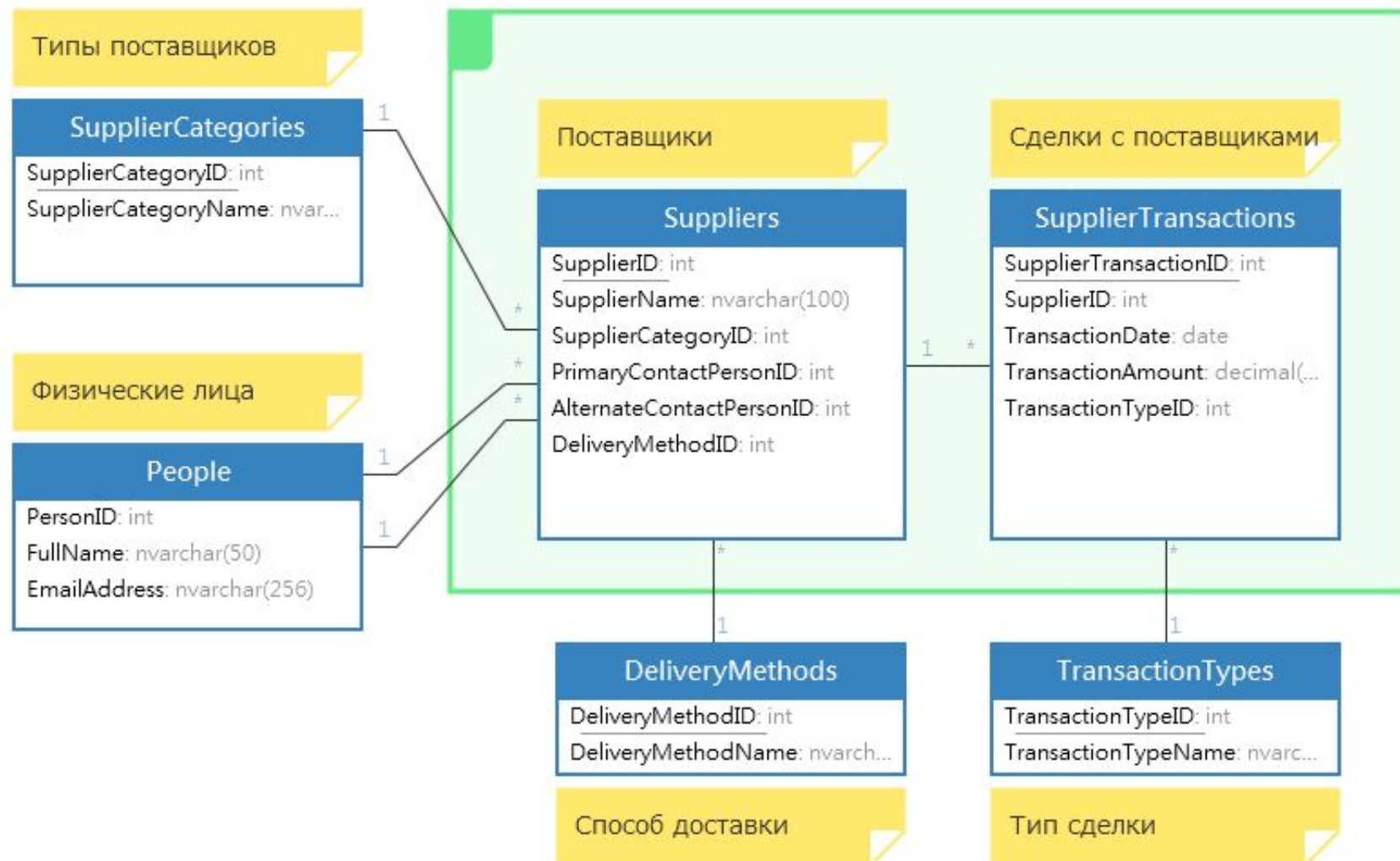
Схема Warehouse — склад

- 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

Таблицы БД WWI, используемые в примерах



ДЕМО JOIN



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 ...]
```

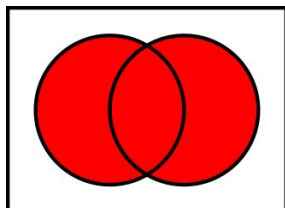
Табл. 1

ID	Name
1	User1
2	User2

Табл. 2

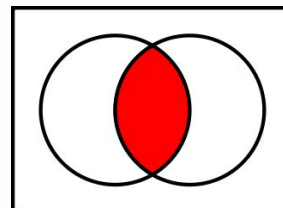
ID	Name
2	User2
3	User3
4	User4

UNION, UNION ALL



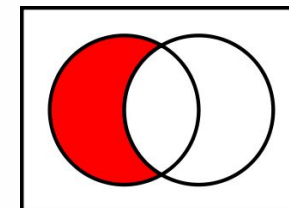
ID	Name
1	User1
2	User2
3	User3
4	User4

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

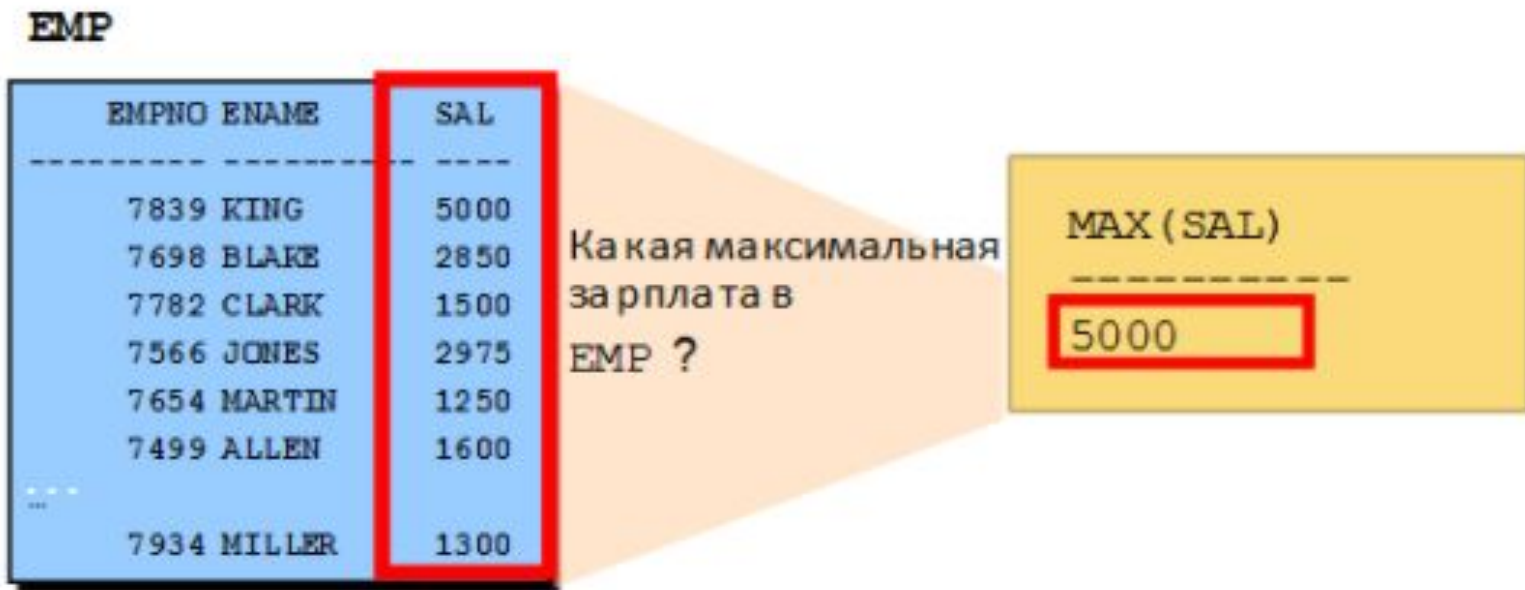
04

Агрегатные функции

- Что это такое?
- Примеры?
- Кто какие знает?



Агрегатные функции принимают множество строк и возвращают одно значение



- Стандартные - COUNT, SUM, AVG, MIN, MAX
- MS SQL - COUNT_BIG, APPROX_COUNT_DISTINCT (с 2019), CHECKSUM_AGG, STDEV, STDEVP, VAR, VARP, STRING_AGG
- Можно писать свои на CLR
- В качестве выражения в агрегатных функциях может использоваться константа, функция, столбец
- NULL игнорируется (кроме COUNT(*))
- GROUPING, GROUPING_ID

ДЕМО

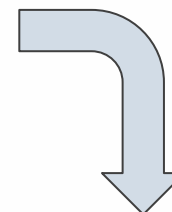
Агрегатные функции



05

GROUP BY, HAVING

SupplierID	SupplierName	SupplierCategoryID	SupplierCategoryName
1	A Datum Corporation	2	Novelty Goods Supplier
2	Contoso, Ltd.	2	Novelty Goods Supplier
3	Consolidated Messenger	6	Courier Services Supplier
4	Fabrikam, Inc.	4	Clothing Supplier
5	Graphic Design Institute	2	Novelty Goods Supplier
6	Humongous Insurance	9	Insurance Services Supplier
7	Litware, Inc.	5	Packaging Supplier
8	Lucerne Publishing	2	Novelty Goods Supplier
9	Nod Publishers	2	Novelty Goods Supplier
10	Northwind Electric Cars	3	Toy Supplier
11	Trey Research	8	Marketing Services Supplier
12	The Phone Company	2	Novelty Goods Supplier
13	Woodgrove Bank	7	Financial Services Supplier



Количество поставщиков по категориям:

```
SELECT
    c.SupplierCategoryName [Category],
    count(*) as [Suppliers Count]
FROM Purchasing.Suppliers s
JOIN Purchasing.SupplierCategories c
    ON c.SupplierCategoryID = s.SupplierCategoryID
GROUP BY c.SupplierCategoryName
```

Category	Suppliers Count
Novelty Goods Supplier	6
Toy Supplier	1
Clothing Supplier	1
Packaging Supplier	1
Courier Services Supplier	1
Financial Services Supplier	1
Marketing Services Supplier	1
Insurance Services Supplier	1

DEMO GROUP BY



```
SELECT ...  
FROM ...  
[WHERE ... ]  
[GROUP BY [ROLLUP | CUBE] ...]  
[GROUPING SETS ...]  
[HAVING ...]  
[ORDER BY ...]
```


GROUPING SETS	Эквивалент GROUP BY
GROUP BY GROUPING SETS (a, b, c)	GROUP BY a UNION ALL GROUP BY b UNION ALL GROUP BY c
GROUP BY GROUPING SETS(a, b, (b, c))	GROUP BY a UNION ALL GROUP BY b UNION ALL GROUP BY b, c
GROUP BY GROUPING SETS((a, b, c))	GROUP BY a, b, c
GROUP BY GROUPING SETS (a, ROLLUP(b,c))	GROUP BY a UNION ALL GROUP BY ROLLUP(b,c)

- **GROUP BY**
Агрегация по одному или нескольким полям
- **ROLLUP**
Формирует промежуточные итоги для каждого указанного элемента и общий итог
- **CUBE**
Формирует результаты для всех возможных перекрестных вычислений
- **GROUPING**
Равно 1, если строка является результатом промежуточного итога
- **GROUPING_ID**
Нумерация уровней группировки

07

Домашнее задание

Материалы



hw_select_group.sql

[Скачать](#)



02 - Оператор SELECT и простые фильтры, GROUP BY, HAVING.pdf
Файл, 07.04.2021



hw_select_group.sql
Файл, 07.04.2021



SELECT (Transact-SQL) - SQL Server
Ссылка, 14.01.2021



Почему SQL Server не гарантирует сортировку результатов без ORDER BY
Ссылка, 14.01.2021



Ошибки при работе с датой и временем в SQL Server
Ссылка, 14.01.2021

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

TODO: 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.
*/
```

TODO: напишите здесь свое решение

```
/*
```

The End

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



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

