



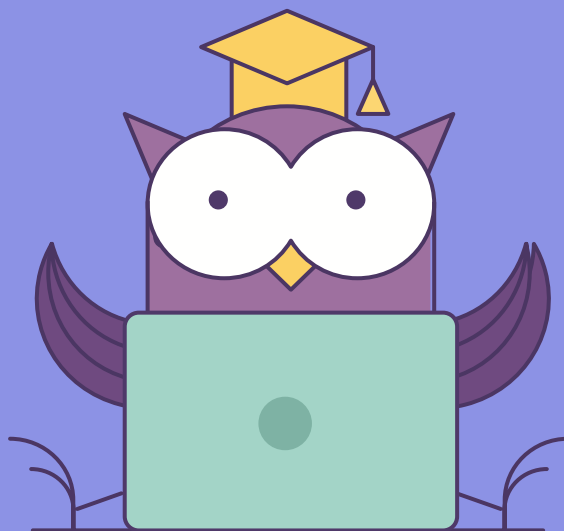
ОНЛАЙН-ОБРАЗОВАНИЕ

Оператор `SELECT` и простые фильтры

Курс “MS SQL Server разработчик”



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо
Или напишите, какие есть проблемы

- Какие СУБД бывают?
- Основные компоненты MS SQL Server
- Доп. возможности MS SQL Server (кроме хранения данных)
- Вопросы по домашнему заданию
- Кто какие проекты с СУБД нашел?

- вспомнить самое основное по реляционной теории

- научиться писать простые SELECT

Подзапросов, группировок, аналитических функций, оконных функций и тп сегодня не будет

1. Реляционная модель
2. SELECT из одной таблицы
3. WHERE
4. NULL
5. Строки, даты
6. JOIN
7. UNION
8. Порядок выполнения SELECT
9. Домашнее задание



01

Реляционная модель

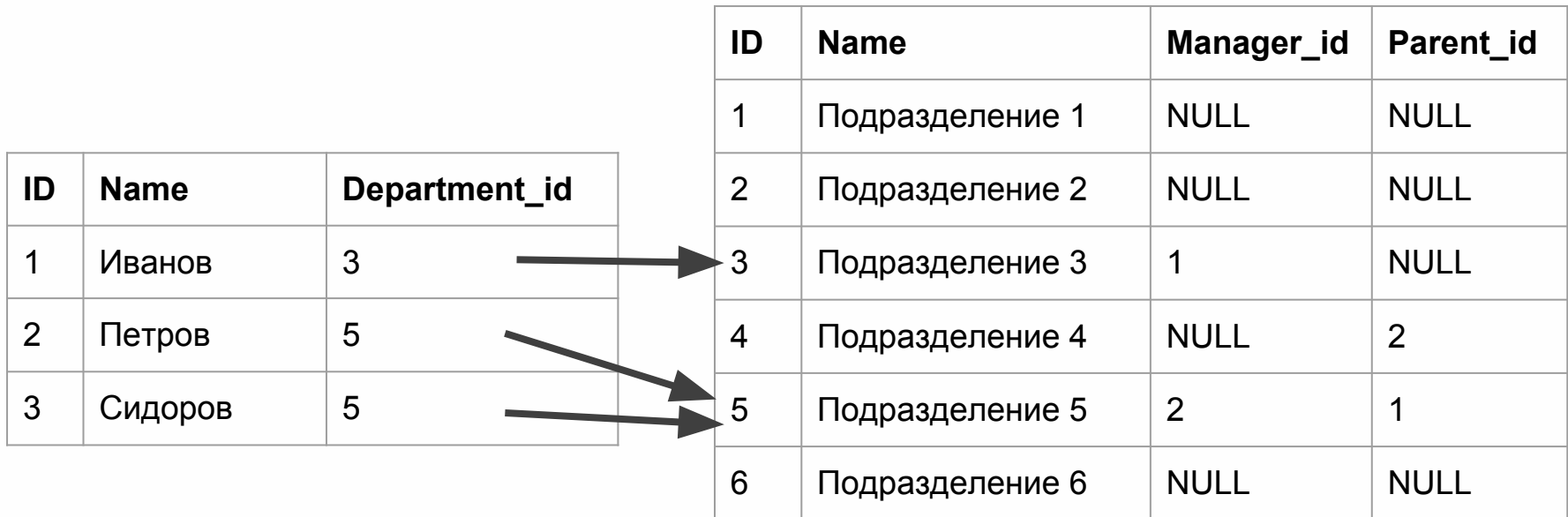
- Нормализация
- Работа с множествами
- Трехзначная логика, NULL

- Реляционная модель данных для больших совместно используемых банков данных
- 12 правил Кодда

Термин реляционной модели	Термин РСУБД
Отношение	Таблица
Кортеж	Строка
Атрибут	Колонка (столбец)

Операции:

- Объединение (UNION)
- Пересечение (INTERSECT)
- Вычитание (DIFFERENCE, MINUS)
- Декартово произведение (Cartesian)
- Выборка (SELECT)
- Проекция (PROJECT)
- Соединение (JOIN)
- Деление (DEVIDE BY)



Натуральные ключи vs искусственные

- Декларативный
- Transact-SQL (TSQL) - расширение SQL в MS SQL Server
- DDL - Data Definition Language
CREATE, ALTER, DROP
- DML - Data Manipulation Language
SELECT, INSERT, UPDATE, DELETE
- DCL - Data Control Language
GRANT, REVOKE, DENY
- TCL - Transaction Control Language
COMMIT, ROLLBACK, SAVEPOINT

02

SELECT из одной таблицы

```
<SELECT statement> ::=
    [ WITH { [ XMLNAMESPACES ,] [ <common_table_expression> [,...n] ] } ]
    <query_expression>
    [ ORDER BY { order_by_expression | column_position [ ASC | DESC ] }
    [ ,...n ] ]
    [ <FOR Clause> ]
    [ OPTION ( <query_hint> [ ,...n ] ) ]
<query_expression> ::=
    { <query_specification> | ( <query_expression> ) }
    [ { UNION [ ALL ] | EXCEPT | INTERSECT }
    <query_specification> | ( <query_expression> ) [...n] ]
<query_specification> ::=
SELECT [ ALL | DISTINCT ]
    [TOP ( expression ) [PERCENT] [ WITH TIES ] ]
    <select_list >
    [ INTO new_table ]
    [ FROM { <table_source> } [ ,...n ] ]
    [ WHERE <search_condition> ]
    [ <GROUP BY> ]
    [ HAVING < search_condition > ]
```

SELECT

```
<SELECT statement> ::=
```

```
    [ ORDER BY { order_by_expression | column_position [ ASC | DESC ] }  
    [ { UNION [ ALL ] | EXCEPT | INTERSECT }
```

```
SELECT [ ALL | DISTINCT ]  
    [TOP ( expression ) [PERCENT] [ WITH TIES ] ]  
< select_list >  
  
    [ FROM { <table_source> } [ ,...n ] ]  
    [ WHERE <search_condition> ]
```

ДЕМО SELECT



- Если в именах пробелы, то можно использовать [] или " "
- Можно указать, какие колонки возвращать. Лучше не делать "SELECT *"
- Источником может быть таблица, представление, функция, CTE, временная таблица, подзапрос
- Можно указывать псевдонимы таблиц и колонок. Можно с AS, можно без AS
- Если порядок возвращаемых строк не определен, если его не задать
- Комментарии `/* */`, `--``

03

WHERE



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

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)

- 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>

ДЕМО WHERE



04

NULL



- 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

- 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

ДЕМО NULL



Будет ли выбран null?

Если нет, то как нужно переписать запрос?

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

- WHERE ... IS NULL, WHERE ... IS NOT NULL
- Функции ISNULL(), COALESCE()
- Если на столбец можно сделать ограничение NOT NULL, то лучше его сделать
- NULL в OLTP vs OLAP
- SET ANSI_NULLS { ON | OFF }
- SET CONCAT_NULL_YIELDS_NULL { ON | OFF }

05

Даты, строки

Какая дата указана в запросе?

```
SELECT *  
FROM [Sales].[Orders]  
WHERE OrderDate > '01.05.2016'  
ORDER BY OrderDate
```

- 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>

- Collation (параметры сортировки)

ДЕМО

Даты, Строки



06

JOIN



- **CROSS JOIN**

Декартово произведение

- **INNER JOIN**

Декартово произведение + фильтрация

- **LEFT/RIGHT JOIN**

Декартово произведение + фильтрация + внешние строки

- **FULL JOIN**

LEFT JOIN + RIGHT JOIN

CROSS JOIN

Таблица А (Person)

Name	CityId
Андрей	1
Леонид	2
Сергей	1
Григорий	4

Таблица В (City)

Id	Name
1	Москва
2	Санкт-Петербург
3	Казань



CROSS JOIN

CROSS JOIN

Name	CityId	B.Id	B.Name
Андрей	1	1	Москва
Андрей	1	2	Санкт-Петербург
Андрей	1	3	Казань
Леонид	2	1	Москва
Леонид	2	2	Санкт-Петербург
Леонид	2	3	Казань
Сергей	1	1	Москва
Сергей	1	2	Санкт-Петербург
Сергей	1	3	Казань
Григорий	4	1	Москва
Григорий	4	2	Санкт-Петербург
Григорий	4	3	Казань

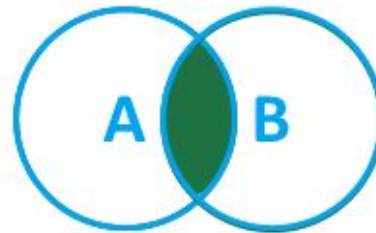
INNER JOIN

Таблица А (Person)

Name	CityId
Андрей	1
Леонид	2
Сергей	1
Григорий	4

Таблица В (City)

Id	Name
1	Москва
2	Санкт-Петербург
3	Казань



INNER JOIN

INNER JOIN

Name	CityId	B.Id	B.Name
Андрей	1	1	Москва
Сергей	1	1	Москва
Леонид	2	2	Санкт-Петербург

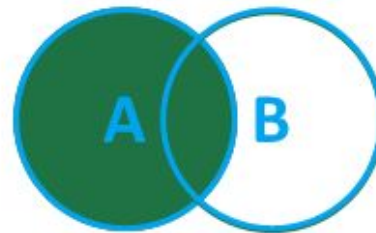
LEFT JOIN

Таблица A (Person)

Name	CityId
Андрей	1
Леонид	2
Сергей	1
Григорий	4

Таблица B (City)

Id	Name
1	Москва
2	Санкт-Петербург
3	Казань



LEFT JOIN

LEFT JOIN

Name	CityId	B.Id	B.Name
Андрей	1	1	Москва
Сергей	1	1	Москва
Леонид	2	2	Санкт-Петербург
Григорий	4		

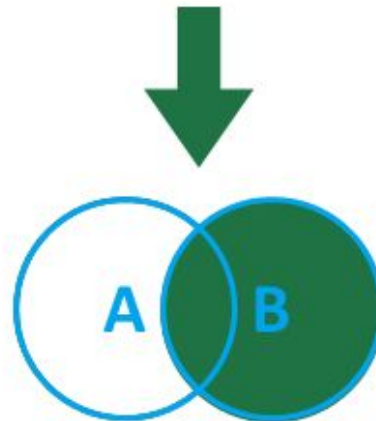
RIGHT JOIN

Таблица А (Person)

Name	CityId
Андрей	1
Леонид	2
Сергей	1
Григорий	4

Таблица В (City)

Id	Name
1	Москва
2	Санкт-Петербург
3	Казань



RIGHT JOIN

RIGHT JOIN

Name	CityId	B.Id	B.Name
Андрей	1	1	Москва
Сергей	1	1	Москва
Леонид	2	2	Санкт-Петербург
		3	Казань

ДЕМО



По каким видам колонок можно делать JOIN?

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

07

UNION



UNION

“вертикальное соединение”

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

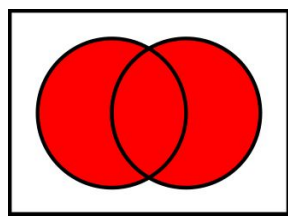
Табл. 1

ID	Name
1	User1
2	User2

Табл. 2

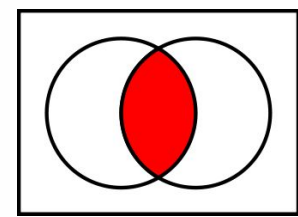
ID	Name
2	User2
3	User3
4	User4

UNION, UNION ALL



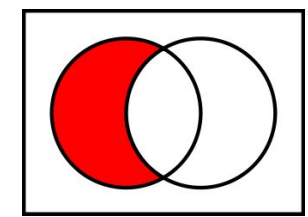
ID	Name
1	User1
2	User2
3	User3
4	User4

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

08

Порядок выполнения SELECT

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]
```

```
(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>;
```

9

Домашнее задание

- SQL-скрипты в github
- Для каждого ДЗ своя папка

HW01

`query.sql`

HW02

`abc.sql`

...

- В чат в личном кабинете ссылку на github
- Если github совсем сложно, то можно и в ЛК отправить файл

Текст ДЗ в личном кабинете на otus.ru

The End



- Простые SELECT
- WHERE
- Некоторые особенности NULL, строк, дат



- Что запомнилось?
- Узнали ли что-то новое?



Заполните,
пожалуйста,
опрос в ЛК о занятии
<https://otus.ru/polls/2647/>

