



[Анализ данных на языке SQL]

День 2

Вечер 3

www.specialist.ru

Программа курса

- День 1. Работа с одной таблицей
- **День 2. Работа с несколькими таблицами**
- День 3. Другие вопросы (более сложные)

День 2. Работа с несколькими таблицами

- Утренняя разминка
- Группировка
- Фильтрация сгруппированной таблицы
- Работа с несколькими таблицами (сценарии)
- Операции над множествами
- Подзапросы

Утренняя разминка

Какой продавец оформил самый последний заказ в Париж?

Решение

-- Какой продавец оформил самый последний заказ в Париж?

```
SELECT TOP (1) WITH TIES EmployeeID
FROM Orders
WHERE ShipCity = 'Paris'
ORDER BY OrderDate DESC
```

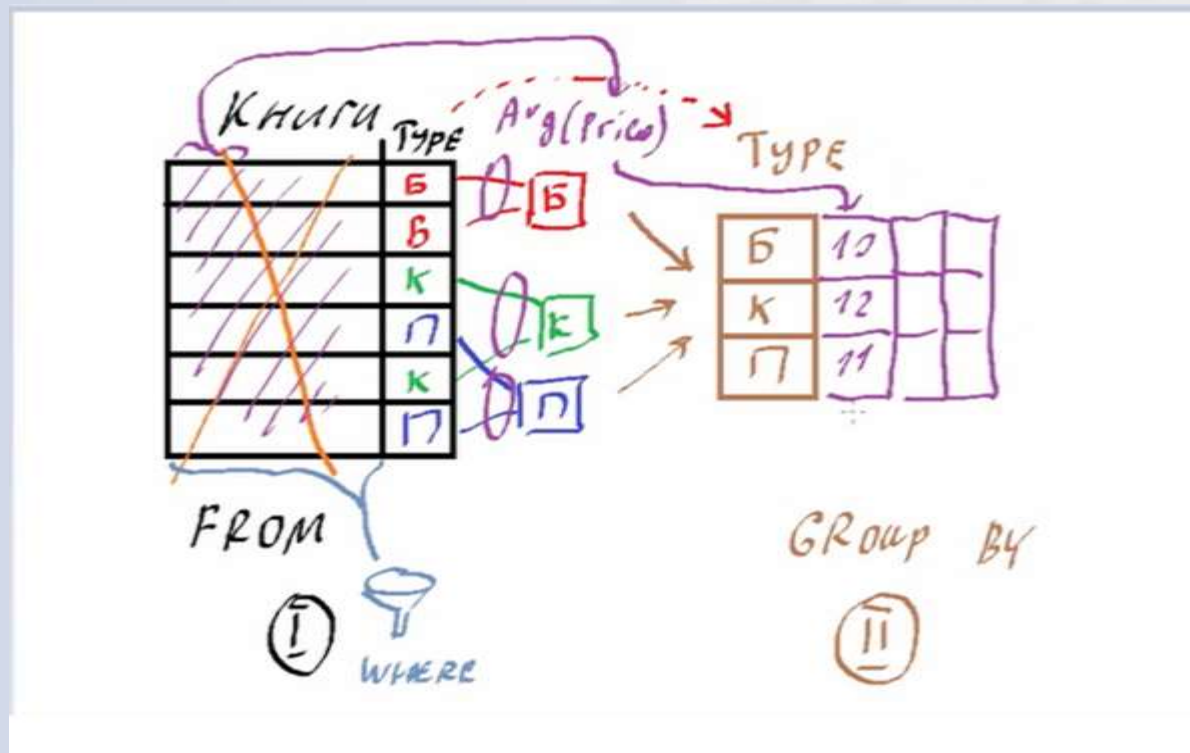
100 % <

Results Messages

	EmployeeID
1	5

Группировка

- Цель GROUP BY - не считать агрегаты
- Цель группировки – из одного списка получить другой (был список книг, а стал список категорий)
- Агрегатная функция применяется к первой таблице в момент группировки



Примеры группирования

-- В какой категории больше всего товаров?

```
SELECT TOP (1) WITH TIES CategoryID--, Count (*)  
FROM Products  
GROUP BY CategoryID  
ORDER BY Count (*) DESC
```

100 %

Results Messages

CategoryID
1

-- Сколько заказов оформлено в каждый город?

```
SELECT ShipCity, Count (*)  
FROM Orders  
GROUP BY ShipCity
```

100 %

Results Messages

ShipCity	(No column name)
1 Aachen	6
2 Albuquerque	18
3 Anchorage	10
4 Athens	11
5 Barcelona	5
6 Barquisimeto	14
7 Bergamo	10
8 Berlin	6
9 Bern	8
10 Boise	31
11 Bracke	19

Query executed successfully.

(local) (12.0 RTM) | CKO-ALM4N08U701\Федор ... | Northwind | 00:00:00 | 70 rows

Примеры группирования

- Можно группировать по вычисляемой колонке. Например, по функции Year()

```
SELECT Year (BirthDate), Count (*)  
FROM Employees  
GROUP BY Year (BirthDate)
```

100 %

Results Messages

	(No column name)	(No column name)
1	1937	1
2	1948	1
3	1952	1
4	1955	1
5	1958	1
6	1960	1
7	1963	2
8	1966	1

Примеры группирования

- Можно группировать по нескольким полям

```
SELECT ShipCountry, Year (OrderDate), Count (*)
FROM Orders
GROUP BY ShipCountry, Year (OrderDate)
```

100 %

Results Messages

	ShipCountry	(No column name)	(No column name)
1	Austria	1996	8
2	Belgium	1996	2
3	Brazil	1996	13
4	Canada	1996	4
5	Denmark	1996	3
6	Finland	1996	4
7	France	1996	15
8	Germany	1996	24
9	Ireland	1996	5
10	Italy	1996	3
11	Mexico	1996	9
12	Norway	1996	1
13	Poland	1996	1
14	Portugal	1996	4
15	Spain	1996	6
16	Sweden	1996	6
17	Switzerland	1996	3

Группировка – это сжатие или растяжение?

- Для программиста: группирование – это сжатие информации (была детализация – стала группа)
- Для аналитика: группирование – это растяжение таблицы (интересуют продажи не только по странам, но и по городам, годам, месяцам, клиентам и тд)
- **SELECT Count(*) FROM Orders** – формально группировки нет, а на самом деле вся таблица Orders представлена одной группой.
- Всего 830 заказов по таблице Orders. Но теперь интересно, как эти 830 заказов разложились по годам, странам, заказчикам
- **GROUP BY Year(orderdata), ShipCountry** и т.п.

Фильтрация сгруппированной таблицы

- Выражение HAVING
- Иногда говорят, что HAVING - это фильтрация по агрегатной функции

-- Какие товары принесли выручку больше 10000?

```
SELECT ProductID, sum (UnitPrice * Quantity * (1 - Discount))
FROM [Order Details]
GROUP BY ProductID
HAVING sum (UnitPrice * Quantity * (1 - Discount)) > 10000
```

	ProductID
1	69
2	29
3	72
4	26
5	12
6	35
7	63
8	55
9	43
10	27
11	64

Другие примеры HAVING

-- Какие продавцы в 1997 году сумели обслужить
-- больше пяти городов в одной стране?

```
SELECT DISTINCT EmployeeID
FROM Orders
WHERE Year (OrderDate) = 1997
GROUP BY EmployeeID, ShipCountry
HAVING Count (DISTINCT ShipCity) > 5
```

100 %
Results Messages

	EmployeeID
1	1
2	2
3	3
4	4
5	8

-- В каких категориях максимальная цена товара превышает 50?

```
SELECT CategoryID
FROM Products
GROUP BY CategoryID
HAVING Max (UnitPrice) > 50
```

100 %
Results Messages

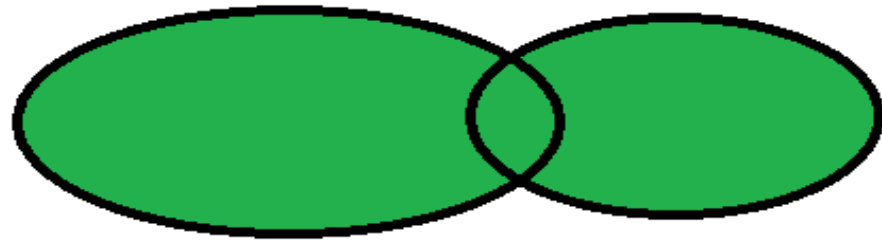
	CategoryID
1	1
2	3
3	4
4	6
5	7
6	8

Работа с несколькими таблицами (стратегии)

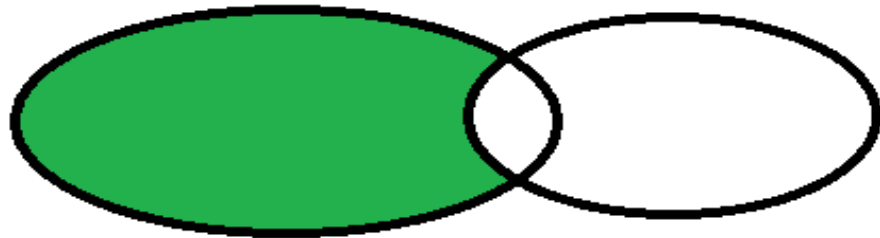
- Подзапрос
 - Простой
 - Связанный (коррелированный)
- JOIN (соединение)
 - INNER (внутреннее)
 - OUTER (внешнее) -> LEFT, RIGHT, FULL
 - CROSS (декартовое произведение)
- Операции над множествами
 - UNION (объединение) -> ALL
 - EXCEPT (исключение)
 - INTERSECT (пересечение)
 - APPLY -> CROSS, OUTER

Операции над множествами

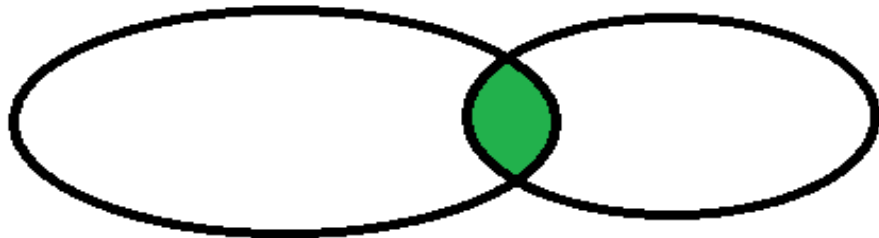
- **UNION**



- **EXCEPT**



- **INTERSECT**



UNION, EXCEPT, INTERSECT

-- UNION (объединить записи)

```
select City, Country from Customers  
union all
```

```
select City, Country from Employees
```

UNION, EXCEPT, INTERSECT

-- EXCEPT (вычесть из первого набора второй)

(Важен порядок таблиц)

```
select City, Country from Customers
```

except --Минус города с сотрудниками

```
select City, Country from Employees
```


UNION, EXCEPT, INTERSECT

-- INTERSECT (найти пересечение)

-- Города с покупателями и сотрудниками

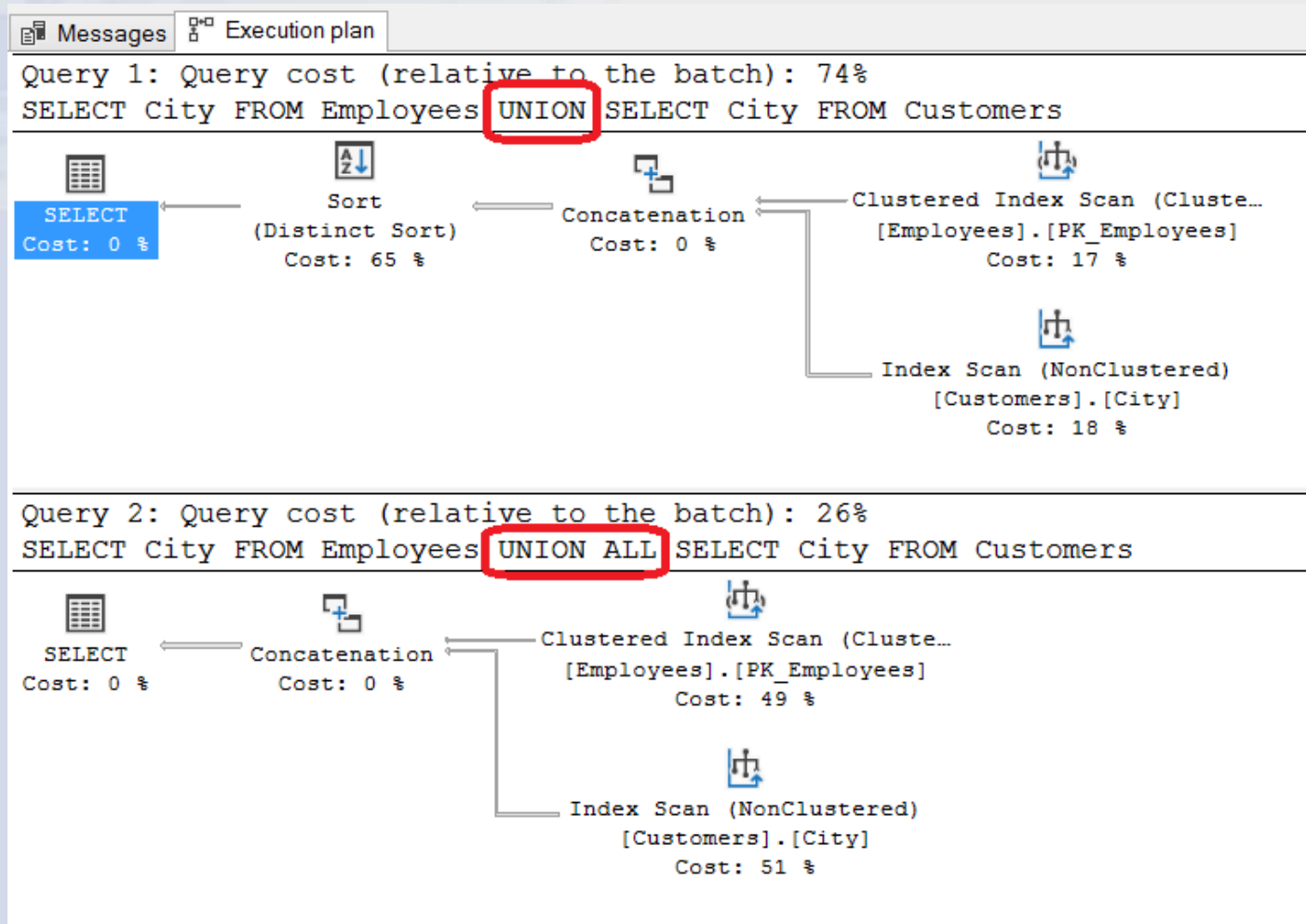
select City, Country from Customers

intersect

select City, Country from Employees

Вопросы быстродействия UNION

UNION ALL -
быстрее,
чем просто
UNION



Подзапросы

- Подзапросы – это вложенные запросы
- Считается, что JOINS – для профессионалов, а подзапросы – для начинающих
- Но это не совсем так, иначе бы 2-го механизма не было, а они есть в любой СУБД
- Подзапросы проще, и как правило более безошибочны
- Недостатки: более громоздки и менее поддаются оптимизации
- JOINS – более сложны и содержат незаметные ошибки (очень коварны)

Пример вложенного запроса

Коррелированный подзапрос

```
-- Сколько заказов оформил каждый продавец (ФИО)?  
  
SELECT EmployeeID, FirstName, LastName,  
    (  
        SELECT Count (*)  
        FROM Orders  
        WHERE EmployeeID = Employees.EmployeeID  
    )  
FROM Employees
```

Results Messages

EmployeeID	FirstName	LastName	(No column name)
1	Nancy	Devolio	123
2	Andrew	Fuller	95
3	Janet	Leverling	127
4	Margaret	Peacock	156
5	Steven	Buchanan	42
6	Michael	Suyama	67
7	Robert	King	72
8	Laura	Callahan	104
9	Anne	Dodsworth	43

Вопросы?