### Определения (Лекция 3)

**Повторение пройденного:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **id1** | **id2** | | 1 | 10 | | 1 | 20 | | 2 | 30 | | SELECT \*  FROM T  ORDER BY id1, id2 desc | SELECT TOP 1 id2  FROM T  ORDER BY id1 |
| SELECT TOP 1 WITH TIES id2  FROM T  ORDER BY id1 | SELECT TOP 1 WITH TIES id1  FROM T  ORDER BY id2 | SELECT distinct id2  FROM T  ORDER BY id1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ндок** | **Дата** | **Покупатель\_ID** | **Сумма** |
| 1 | 10.09.16 | 3 | 200 |
| 2 | 12.09.16 | 2 | 90 |
| 3 | 15.09.16 | 2 | 20 |
| 4 | 16.09.16 | 3 | 50 |

**Совместное использование TOP и DISTINCT:**

|  |  |
| --- | --- |
| SELECT DISTINCT TOP 2 Покупатель\_ID  FROM Документы  ORDER BY Покупатель\_ID asc | SELECT TOP 2 Покупатель\_ID  FROM Документы  ORDER BY Покупатель\_ID asc |

**Полезная функция:**

|  |  |
| --- | --- |
| IIF: IIF(A = B,  Что выводить в случае равенства,  Что выводить иначе) | CASE  WHEN A = B THEN ЧтоВыводить  WHEN A = C THEN ….  ELSE ИНАЧЕ END |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT \*,  CASE ID1  WHEN 1 THEN 'Первый' WHEN 2 THEN 'Второй' ELSE 'Нет' END AS Коммент1,  CASE WHEN ID1 + ID2 < 20 THEN 'Сумма меньше 20 ' WHEN ID1 + ID2 > 10 THEN 'Сумма больше 10' ELSE 'Неизвестно' END Коммент2  FROM @T | |  |  |  |  | | --- | --- | --- | --- | | id1 | id2 | Коммент1 | Коммент2 | | 1 | 10 | Первый | Сумма меньше 20 | | 1 | 20 | Первый | Сумма больше 10 | | 2 | 30 | Второй | Сумма больше 10 |   iif(A=B; если Истина; если Ложь) |

**Работа с CAST (еще полезные: ROUND, FLOOR, CEILING)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Из** | **В** | **Поведение** | | Float | Int | Truncate | | Float | datetime | Round | | datetime | Int | Round | | SELECT CAST(4.5 as int) t1, CAST(4.2 as int) t2, CAST(4.7 as int) t3, 5/3 t4,  CAST(GETDATE() as int) t5, CAST(GETDATE() as float) t6, CAST(CAST(GETDATE() as float) as int) t7   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | t1 | t2 | t3 | t4 | t5 | t6 | t7 | | 4 | 4 | 4 | 1 | 42655 | 42654.63593 | 42654 | |

**Работа с датой (T-SQL):**

|  |  |
| --- | --- |
| CAST(GETDATE() as date) | CAST(FLOOR(CAST(GETDATE() as float)) as datetime) |
| DATEPART(DW, GETDATE()) | DATEPART(MONTH, GETDATE()), MONTH(GETDATE()) |
| DATEADD(*datepart***,** *number***,** *date*) | DATEDIFF(*datepart, startdate, enddate*) |
| Datepart: ss (second), n (minute), hh (hour), dd (day), mm (month), y (year), dw(day of week), ww (week) | |

**Access:** Cdbl(CDATE(Now()))

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **T-SQL:** | |  |  | | --- | --- | | id | data | | 1 | 06.10.16 | | 2 | 10.06.16 | | SELECT \*, MONTH(data), YEAR(data)  FROM T  WHERE data >= '20161001' |
| SET dateformat DMY  SELECT \*  FROM T  WHERE data >= '01.10.16' | | SET dateformat MDY  SELECT \*  FROM T  WHERE data >= '01.10.16' |

**Работа с параметрами**

|  |  |  |
| --- | --- | --- |
| DECLARE @id int  SET @id = 2  SELECT \* FROM Документы WHERE Покупатель\_ID = @id | set @q :=’название товара’;    SELECT \*  FROM Продажи  WHERE Товар = @q | \set param 1  SELECT \*  FROM Продажи  WHERE Товар  = :param |

**Операторы в WHERE (in и between):**

WHERE (П1>Значение1 or П1 < значение2) AND П2 <> значение3. («<>» == «!=»)

|  |  |
| --- | --- |
| SELECT \* FROM Документы  WHERE Покупатель\_ID in (1, 2) | SELECT \* FROM Товары  WHERE Масса BETWEEN 0.10 AND 0.12 |

SELECT \* FROM Покупатели WHERE Покупатель BETWEEN 'А' AND 'С'

Покупатели: Иванов, Андреев, Петров, Сидоров

**Оператор LIKE:** WHERE Наименование Like (примеры с одним шаблоном в начале слова, но применимо и в других вариантах)

|  |  |  |
| --- | --- | --- |
| **MS SQL** | **Access** | **Результат** |
| '\_пельсин' | “?пельсин” | Апельсин, Опельсин, но не Пельсин |
| '%пельсин' | “\*пельсин” | Апельсин, Опельсин, Пельсин |
| '[А-О]пельсин' | “[А-О]пельсин“ | Апельсин, Впельсин, Опельсин |
| '[АОЮ]пельсин' | “[АОЮ]пельсин“ | Апельсин, Опельсин, Юпельсин |
| '[^Б-О]пельсин' | “[!Б-О]пельсин“ | Апельсин, но не Впельсин и не Пельсин |
| '[\_]пельсин' | “[?]пельсин“ | \_пельсин (?пельсин), но не иные. |

DECLARE @Val TABLE (Товар\_ID int, Название nvarchar(30))

INSERT INTO @Val

VALUES (1, 'Тетрадь'), (2, 'Тетрадь белая'), (3, 'Тетрадь зеленая')

DECLARE @pat nvarchar(30)

SET @pat = 'Тетрадь%'

SELECT \* FROM @Val WHERE Название like @pat

SELECT \* FROM @Val WHERE Название = @pat

**Использование IF и WHILE**

|  |  |
| --- | --- |
| DECLARE @n int, @p int  SET @n = 3  IF @n > 3  BEGIN  SET @p = 1  END  ELSE  BEGIN  SET @p = 0  END  SELECT @p | DECLARE @n int, @p int  SET @n = 3  SET @p = 1  WHILE @n > 0  BEGIN  SET @p = @n\*@p  SET @n = @n - 1  END  SELECT @p |

**Запросы с группировкой:**

**Что такое агрегатные функции и группировки:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | id1 | id2 | val | | 1 | 1 | 10 | | 1 | 2 | 20 | | 2 | 1 | 30 | | SELECT MIN(val), MAX(val), COUNT(val), COUNT(\*), SUM(val), AVG(val)  FROM T | SELECT id1, MIN(val)  FROM T  GROUP BY id1 | SELECT id1, id2, MIN(val)  FROM T  GROUP BY id1, id2 |

**Примеры:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Ндок** | **…** | **Покупатель\_ID** | **Сумма** | | 1 |  | 1 | 1 | | 2 |  | 3 | 10 | | 3 |  | 2 | 14 | | 4 |  | 1 | 20 | | 5 |  | 2 | 21 | | SELECT COUNT(Покупатель\_ID), COUNT(distinct Покупатель\_ID),  FROM Документы  SUM, AVG, MIN, MAX… | SELECT Покупатель\_ID, MIN (Сумма), SUM(Сумма)  FROM Документы  GROUP BY Покупатель\_ID |

**Есть ли разница в результате?**

|  |  |
| --- | --- |
| SELECT Покупатель\_ID  FROM Документы  GROUP BY Покупатель\_ID | SELECT distinct Покупатель\_ID  FROM Документы |

**Что выдадут запросы:**

|  |  |  |
| --- | --- | --- |
| SELECT Покупатель\_ID,  SUM (Сумма) as S  FROM Документы  GROUP BY Покупатель\_ID | SELECT SUM (Сумма)  FROM Документы  GROUP BY Покупатель\_ID | SELECT SUM (Сумма)  FROM Документы |

SELECT Покупатель, ндок, SUM(Сумма) FROM Документы GROUP BY Покупатель\_ID, ндок

Будут ли работать следующие запросы?

|  |  |
| --- | --- |
| SELECT ндок, SUM(Сумма)  FROM Документы  GROUP BY Покупатель\_ID | SELECT ндок  FROM Документы  WHERE Сумма = MAX(Сумма) |

Следующий синтаксис допустим в MySQL, но выведется абсолютно произвольный покупатель, а не тот, у которого максимальная сумма.

SELECT Покупатель\_ID, MAX (Сумма)

FROM Документы

Вопрос:

Как без подзапросов вывести покупателей, которые закупились В СУММЕ на максимальную величину?

**Есть ли разница?**

|  |  |
| --- | --- |
| SELECT Покупатель\_ID, SUM (Сумма)  FROM Документы  WHERE Покупатель\_ID = 1  GROUP BY Покупатель\_ID | SELECT Покупатель\_ID, SUM (Сумма)  FROM Документы  GROUP BY Покупатель\_ID  HAVING Покупатель\_ID = 1 |

**Сравнить запросы:**

|  |  |  |
| --- | --- | --- |
| SELECT Покупатель\_ID,  SUM (Сумма)  FROM Документы  WHERE Сумма > 100  GROUP BY Покупатель\_ID | SELECT Покупатель\_ID,  SUM (Сумма)  FROM Документы  GROUP BY Покупатель\_ID  HAVING SUM (Сумма) > 100 | SELECT Покупатель\_ID, SUM (Сумма)  FROM Документы  GROUP BY Покупатель\_ID  HAVING MIN (Сумма) > 100 |

**Результат запроса?**

|  |  |
| --- | --- |
| DECLARE @T TABLE (id int, val float)  INSERT INTO @T VALUES (1, 10), (2, 15) | SELECT MIN(id)  FROM @T  HAVING SUM(val) > 10 |

**Дополнительная информация:**

|  |  |  |
| --- | --- | --- |
| **Подзапросы:**  SELECT \*  FROM  (  SELECT Поле  FROM Таблица  ) as T | **Как помещать результат запроса в переменную**  DECLARE @count int  SET @count =  (SELECT COUNT(\*) FROM Документы WHERE Покупатель\_ID = 1) | |
| SELECT Покупатель\_ID  FROM Документы  GROUP BY Покупатель\_ID  HAVING COUNT(\*) > @count | SELECT TOP (@count) Покупатель\_ID  FROM Документы  GROUP BY Покупатель\_ID  HAVING COUNT(\*) > @count |