### Определения (Лекция 3)

**Повторение пройденного материала:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **id** | **val** | | 1 | 1 | | 2 | 10 | | 3 | 11 | | 4 | 1 | | SELECT id, val,  'Результат сложения val и id равен: ' + CAST(val + id as nvarchar(10))  FROM T | | SELECT distinct val  FROM T | | SELECT distinct id, val  FROM T | |
| SELECT id  FROM T  ORDER BY id | SELECT ID  FROM T  ORDER BY val | | SELECT distinct val  FROM T  ORDER BY val | | SELECT distinct val  FROM T  ORDER BY id |

Еще больше **TOP (теперь и с ORDER BY):**

|  |  |  |  |
| --- | --- | --- | --- |
| **ндок** | **Дата** | **Покупатель\_ID** | **Сумма** |
| 1 | 10.09.21 | 3 | 200 |
| 2 | 12.09.21 | 2 | 90 |
| 3 | 15.09.21 | 2 | 20 |
| 4 | 16.09.21 | 3 | 50 |

|  |  |  |
| --- | --- | --- |
| SELECT TOP 3 Покупатель\_ID  FROM Документы | SELECT TOP 3 Покупатель\_ID  FROM Документы  ORDER BY Покупатель\_ID asc | SELECT TOP 3 with ties Покупатель\_ID  FROM Документы  ORDER BY Покупатель\_ID asc |
| SELECT TOP 3 with ties Покупатель\_ID  FROM Документы  ORDER BY Дата desc | SELECT TOP 3 with ties Дата  FROM Документы  ORDER BY Покупатель\_ID asc | **-- PostGre, MySQL:**  SELECT Покупатель\_ID  FROM Документы  ORDER BY Покупатель\_ID desc  LIMIT 1 |

**Сортировка по нескольким полям:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица Т: | |  |  | | --- | --- | | id1 | id2 | | 1 | 2 | | 2 | 1 | | SELECT id1, id2  FROM T  ORDER BY id1, id2 desc | SELECT id1, id2  FROM T  ORDER BY 1, 2 desc |

**Более сложный пример:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица Т: | |  |  | | --- | --- | | **id1** | **id2** | | 1 | 1 | | 1 | 2 | | 2 | 3 | | SELECT id1, id2  FROM T  ORDER BY id1, id2 desc | SELECT id2, id1  FROM T  ORDER BY 1, 2 desc |

**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 |

**Выражение CASE:**

|  |  |  |
| --- | --- | --- |
| 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 датой (T-SQL) (Другие СУБД: current\_date, extract, date\_part, date\_trunc, +, -, date\_add):**

|  |  |
| --- | --- |
| 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) | |

**Тонкости работы с CAST (кстати, полезные для работы с числами: 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 | 43376 | 43375.5584849537 | 43375 | |

**Как в целом работать с датой (T-SQL, но общий смысл для всех СУБД одинаков):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **id** | **DT** | | 1 | 2021-10-04 15:12:39 | | 2 | 2021-06-04 15:12:39 | | SELECT \*, MONTH(DT), YEAR(DT)  FROM T  WHERE DT >= '20211004' |
| SELECT \*  FROM T  WHERE DT > '2021-10-04T00:00:00' | SELECT \*  FROM T  WHERE DT > CONVERT(datetime, '04.10.2021', 103) |
| SET dateformat DMY  SELECT \*  FROM T  WHERE DT >= '04.10.21' | SET dateformat MDY  SELECT \*  FROM T  WHERE DT >= '10.04.21' |

**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). Почитайте про WildCards.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **MS SQL** | **Результат** | | '\_пельсин' | Апельсин, Опельсин, но не Пельсин | | '%пельсин' | например: Апельсин, Опельсин, Пельсин | | '[А-О]пельсин' | Например: Апельсин, Впельсин, Опельсин | | '[АОЮ]пельсин' | Только Апельсин, Опельсин, Юпельсин | | '[^Б-О]пельсин' | Апельсин, но не Впельсин и не 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 |

*Доп. материал: В тексте лекции или в MSDN можно прочитать доп материал про сравнение строк с замыкающим пробелом, а так же, как работает like в случае, если строки разного типа (ansi и unicode) и в том числе с замыкающими пробелами*