**Определения 7.**

**NULL-значения**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Артикул | Наименование | Масса | | 1 | Мышка | 0.02 | | 2 | Монитор | 5 | | 3 | Системный блок | 3 | | 4 | Клавиатура | NULL | | 5 | Видеокарта | 1 | | SELECT \*  FROM Номенклатура  WHERE Масса = 1 | | SELECT \*  FROM Номенклатура  WHERE Масса <> 1 |
| SELECT \*  FROM Номенклатура  WHERE Масса is NULL | | SELECT \*  FROM Номенклатура  WHERE Масса is NOT NULL |
| |  |  |  | | --- | --- | --- | | X | IS NULL | IS NOT NULL | | TRUE | FALSE | TRUE | | FALSE | FALSE | TRUE | | NULL | TRUE | FALSE | | | **Сравнение с NULL-значением**  **(**ISO standard SET ANSI\_NULLS ON):  WHERE column\_name = NULL /\* 0 строк\*/  WHERE column\_name != NULL /\* 0 строк\*/ | |

**Таблицы истинности**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | X | **Not** X | | True | False | | False | True | | NULL | NULL | | |  |  |  |  | | --- | --- | --- | --- | | **AND** | TRUE | FALSE | NULL | | TRUE | TRUE | FALSE | NULL | | FALSE | FALSE | FALSE | FALSE | | NULL | NULL | FALSE | NULL | | |  |  |  |  | | --- | --- | --- | --- | | **OR** | TRUE | FALSE | NULL | | TRUE | TRUE | TRUE | TRUE | | FALSE | TRUE | FALSE | NULL | | NULL | TRUE | NULL | NULL | |

Пример:

|  |  |
| --- | --- |
| DECLARE @T TABLE (id int, val int)  INSERT INTO @T  VALUES (1, NULL), (2, 1), (3, 2) | SELECT \*  FROM @T  WHERE NOT val is NULL |

**Работа с NULL-значениями в запросах:**

|  |  |  |  |
| --- | --- | --- | --- |
| SELECT SUM(Масса)  FROM Номенклатура | SELECT Артикул,  SUM(Масса) as Масса  FROM Номенклатура  GROUP BY Артикул | SELECT  COUNT(Артикул)  FROM Номенклатура | SELECT  COUNT(Масса),  COUNT(\*)  FROM Номенклатура |
| SELECT Масса, COUNT(Артикул)  FROM Номенклатура  GROUP BY Масса | SELECT isNULL(Масса, 0)  FROM Номенклатура | SELECT \*  FROM Номенклатура  WHERE Масса = NULL  (или > NULL) | SELECT \*  FROM Номенклатура  WHERE Масса NOT IN (1 , 2 , NULL) |

Еще примеры:

|  |  |  |  |
| --- | --- | --- | --- |
| DECLARE @T TABLE  (id int, val int)  INSERT INTO @T VALUES  (1, NULL), (2, 1), (3, 2), (4, NULL) | SELECT \*  FROM @T T1  INNER JOIN @T T2 ON T1.val = T2.val | SELECT distinct val  FROM @T | SELECT id, COUNT(val)  FROM @T  GROUP BY id |

**Полезные функции для работы с NULL-значениями:**

|  |  |
| --- | --- |
| IsNULL() – T-SQL  Coalesce() – T-SQL  SELECT isNULL(Масса, 0)  FROM Номенклатура | DECLARE @T TABLE (t1 int, t2 int, t3 int, t4 int)  INSERT INTO @T VALUES (NULL, NULL, 1, 2)  SELECT COALESCE(t1, t2, t3, t4)  FROM @T |

**LEFT JOIN, RIGHT JOIN, FULL JOIN**

SELECT A.\*, B.\* (указывать атрибуты для вывода)

FROM A LEFT JOIN B ON A.поле = B.поле

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ндок** | **Пок\_ID** |  | **ID** | **Пок** | | 1 | 10 |  | 10 | Иванов | | 2 | 11 |  | 15 | Сидоров | | SELECT \*  FROM Документы LEFT JOIN Покупатели ON  Документы.Пок\_ID = Покупатели.ID |

Что будет, если добавить в запрос: WHERE Покупатель.ID > 5?

SELECT \* FROM Документы RIGHT JOIN Покупатели ON Документы.Пок\_ID = Покупатели.ID

SELECT \* FROM Документы FULL JOIN Покупатели ON Документы.Пок\_ID = Покупатели.ID

**Примеры:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **A** |  | **B** | | **id** |  | **id** | | 1 |  | 1 | | 2 |  | 3 | | SELECT \*  FROM A LEFT JOIN B ON  A.id = B.id | SELECT \*  FROM A LEFT JOIN B ON  A.id = B.id AND  A.id = 2 | SELECT \*  FROM A LEFT JOIN B ON  A.id = B.id  WHERE A.id = 2 |

Вопрос: как получить разницу между таблицами А и B?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **A** |  | **B** | | **id** |  | **id** | | 1 |  | 1 | | 1 |  | 1 | | 2 |  | 3 | | SELECT \*  FROM A LEFT JOIN B ON  A.id = B.id |

**Объединение таблиц (UNION и UNION ALL)**

SELECT поле1, поле2…, полеN

FROM А

WHERE …

UNION

SELECT поле1, поле2…, полеN

FROM B

WHERE …

ORDER BY ….

**Задача:** подсчет выручки в магазине на любой день

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | ндок | Дата | Пок | Сумма | Анн | аДата | | 1 | 20.10.16 | Иванов | 5000 | True | 21.10.16 | | 2 | 21.10.16 | Иванов | 2000 | False |  | | |  |  | | --- | --- | | Дата | Оборот, руб | | 20.10.16 | 5000 | | 21.10.16 | -3000 | |

**Пример замены UNION на LEFT**

|  |  |
| --- | --- |
| SELECT Покупатель, SUM(Отпущено), SUM(Брак)  FROM  (  SELECT Покупатель, SUM(Колво) as Отпущено, 0 as Брак  FROM Продажи  GROUP BY Покупатель  UNION  SELECT Покупатель, 0, SUM(Колво)  FROM Брак  GROUP BY Покупатель  ) T  GROUP BY Покупатель | SELECT Покупатели.Покупатель,  isNULL(Отпущено, 0) Продажи, isNULL(Брак, 0) Брак  FROM Покупатели  LEFT JOIN  (  SELECT Покупатель, SUM(Колво) as Отпущено  FROM Продажи  GROUP BY Покупатель  ) as Продажи ON  Покупатели.Покупатель = Продажи.Покупатель  LEFT JOIN  (  SELECT Покупатель, SUM(Колво) as  Брак  FROM Брак  GROUP BY Покупатель  ) as Брак ON  Покупатели.Покупатель = Брак.Покупатель |