# Внутренние соединения

Соединения с помощью предиката JOIN бывают внутренние и внешние.

* Внутренние – INNER JOIN, JOIN, CROSS JOIN
* Внешние – OUTER FULL JOIN, RIGHT JOIN, LEFT JOIN

Начнем с внутренних операций JOIN

Пример:

|  |  |  |  |
| --- | --- | --- | --- |
| Auto | | Selling | |
| id | Name | Id | sum |
| 1 | BMW | 1 | 250 |
| 2 | Opel | 5 | 450 |
| 3 | KIA | 3 | 800 |
| 4 | Audi | 6 | 400 |

1. INNER JOIN  
   SELECT \*  
   FROM Auto  
   [INNER] Join Selling  
   ON Auto.id = Selling.id

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Name | Id | sum |
| 1 | BMW | 1 | 250 |
| 3 | KIA | 3 | 800 |

1. CROSS JOIN – выдает все виды комбинаций всех строк первой таблицы со строками второй таблицы.

SELECT Autoid AS id  
FROM Auto  
CROSS Join Selling  
ON Auto.id = Selling.id

# Внешние соединения

1. FULL JOIN – соединяет все (полное соединение всего (все строчки и первой и второй таблицы))

SELECT Auto.id, Auto.name, Auto.sum

FROM Auto

Full Join Selling

ON Auto.id= Selling.id

|  |  |  |
| --- | --- | --- |
| Auto.id | Name | Id |
| 1 | BMW | 250 |
| 2 | Opel | NULL |
| 3 | KIA | 300 |
| 4 | Audi | NULL |
| 5 | NULL | 450 |
| 6 | NULL | 400 |

1. LEFT JOIN – соединяет левую таблицу полностью данными из правой таблицы, если их нет, то заполнит поля NULL  
   SELECT Auto.id, Auto.name, Auto.sum

FROM Auto

LEFT Join Selling

ON Auto.id= Selling.id

|  |  |  |
| --- | --- | --- |
| Auto.id | Name | Id |
| 1 | BMW | 250 |
| 2 | Opel | NULL |
| 3 | KIA | 300 |
| 4 | Audi | NULL |

# Естественные соединения

Задание:   
Вывести имена всех покупателей вместе с именами продавцов, которые их обслуживают.

SELECT Customers.cname, Salespeople.sname

FROM Customers, Salespeople

WHERE Salespeople.snum = Customers.snum

При соединении таблиц СУБД сначала берет строку «Peel» из таблицы Salespeople и по очереди объединяет ее с каждой из строк таблицы Customers. Если значения полей полученных комбинаций придикатов true т.е. условие выполняется, то строка выводится. То же самое делается для продавцов.

21.12.2017

№1

Напишите запрос, который выводит номер каждого заказа вместе с именем покупателя, сделавшего этот заказ.

SELECT Orders.Onum, Customers.Cname

FROM Orders, Customers

WHERE Orders.cnum = Customers.cnum

№2

Составьте запрос, который показывает всех покупателей, обслуживаемых продавцом с комиссионными более 12%. Выведите имя покупателя, имя продавца, и комиссионные продавца

SELECT

FROM Orders, Customers

WHERE Orders.cnum = Customers.cnum

Работа с таблицами

|  |  |  |
| --- | --- | --- |
| Оператор INSERT  INSERT INTO имя табл. VALUES (‘Пётр’, 14,16,2017)  INSERT INTO Salespeople (snum,sname) VALUES (2012, (‘Пётр’)  Заполнение таблицы с помощью другой:  INSERT INTO STUDENTS\_Moscow SELECT \* FORM STUDENTS WHERE city = ‘Москва’ | Оператор delete  DELETE FROM Salespeople;  DROP TABLE Salespeople  DELETE From Salespeople Where city = ‘Москва’ | Оператор update  UPDATE Customers  SET Rating=500  WHERE city = ‘London’; |

Нормализация

Нормализация баз данных предназначена для приведения структуры базы данных к виду, обеспечивающему минимальную избыточность.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер заказа | Покупатель | Товар | цена | Кол-во | Сумма |
| 1001 | Григорьев С | iPhone | 80000 | 2 | 160000 |
| 1002 | Кириллов А | Samsung | 70000 | 1 | 70000 |
| 1003 | Розанов Ю | Xiaomi | 30000 | 3 | 90000 |
| 1004 | Розанов Ю | Nokia | 10000 | 2 | 20000 |

К первой нормальной форме приводятся базы данных, которые уже отвечают реляционным принципам: все должно быть неупорядоченно.

2) Все строки должны быть неупорядоченными

Для того, чтоб утверждать, что таблица соответствует 1-й нормальной форме, необходимо чтобы не было дублирующийся строк. Отсутствуют массивы и списки в любом виде.

Вторая нормальная форма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер курса | Учащийся | Тип обучения | Оценка | Кол-во часов |
| 177415 | Алексеев М. | дистанционн | 5 | 120 |
| 181812 | Барыбин С. | очная | 4 | 40 |
| 181812 | Степанов А. | заочная | 3 | 50 |

Третья нормальная форма

Приводятся в порядок не ключевые столбцы.

Первое правило: Отсутствие транзитивных зависимостей.

Не должно быть вычисляемых столбцов Далее дробление крайне редко необходимо.

11.01.2018

17. Создание коррелированных подзапросов.

Подзапросы делятся на два типа:

* Автономные
* Коррелированные

Автономный запрос – возможно выполнение определенных строк в SQL Server.

Коррелированный – невозможно выполнение определенных строк т.к. есть корреляция с внешними запросами, запрос не выполнится.

Пример автономного запроса:  
SELECT \* FORM Orders WHERE snum (SELECT snum FROM salespeople WHERE city=”Barcelona”)

Пример коррелированного запроса:

SELECT snum, sname FROM Salespeople main WHERE 1 < (SELECT COUNT (\*) FROM Customers WHERE snum=main.snum)

13.Использование предложения GROUP BY

GROUP BY позволяет перейти от одной таблицы с одними объектами к другой таблице с другими объектами.

Пример: SELECT cnum FROM Orders GROUP BY cnum, cname

Выбор клиента, у которого больше 10 заказов:

Пример: SELECT cnum, COUNT(\*) FROM Orders GROUP BY cnum, cname HAVING COUNT(\*)>10

Группирует выбранный набор строк для получения набора сводных строк по значениям одного или нескольких столбцов или выражений в SQL Server 2014. Возвращается одна строка для каждой группы. Агрегатные функции в списке <select> предложения SELECT предоставляют информацию о каждой группе, а не об отдельных строках.

Предложение GROUP BY имеет два синтаксиса: совместимый с ISO и несовместимый с ISO. В каждой отдельной инструкции SELECT может использоваться только один стиль синтаксиса. Во всех новых разработках используйте совместимый с ISO синтаксис. Синтаксис, несовместимый с ISO, служит для обеспечения обратной совместимости.

В этом разделе предложение GROUP BY можно описать как общее или простое.

* Общее предложение GROUP BY включает конструкции GROUPING SETS, CUBE, ROLLUP, WITH CUBE и WITH ROLLUP.
* Простое предложение GROUP BY не включает конструкции GROUPING SETS, CUBE, ROLLUP, WITH CUBE и WITH ROLLUP. Предложение GROUP BY (), предназначенное для определения общего итога, рассматривается как простое предложение GROUP BY.