



Запросы модификации данных

Виды запросов действия:


- `INSERT INTO` – запрос добавления;
- `DELETE` – запрос удаления;
- `UPDATE`–запрос обновления.

Запрос добавления

Оператор **INSERT** применяется для добавления записей в таблицу.

Формат оператора:

***INSERT INTO имя_таблицы
[(имя_столбца [...n])
{VALUES (значение
[,...n])|SELECT_оператор}***



Список значений должен следующим образом соответствовать списку столбцов:

- количество элементов в обоих списках должно быть одинаковым;
- должно существовать прямое соответствие между позицией одного и того же элемента в обоих списках, поэтому первый элемент списка значений должен относиться к первому столбцу в списке столбцов, второй – ко второму столбцу и т.д.
- типы данных элементов в списке значений должны быть совместимы с типами данных соответствующих столбцов таблицы.

Пример. Добавить в таблицу
ТОВАР новую запись.

***INSERT INTO Товар (Название, Тип,
Цена)VALUES(" Славянский ", " шоколад ", 12)***

Если столбцы таблицы *ТОВАР* указаны в полном составе и в том порядке, в котором они перечислены при создании таблицы *ТОВАР*, оператор можно упростить.

INSERT INTO Товар VALUES (" Славянский ", " шоколад ", 12)

Пример. Добавить в итоговую таблицу сведения об общей сумме ежемесячных продаж каждого наименования товара.

```
INSERT INTO Итог (Название, Месяц, Стоимость )  
SELECT Товар.Название, Month(Сделка.Дата) AS  
Месяц, Sum(Товар.Цена*Сделка.Количество) AS  
Стоимость  
FROM Товар INNER JOIN Сделка ON Товар.  
КодТовара = Сделка.КодТовара  
GROUP BY Товар.Название, Month(Сделка.Дата)
```

Запрос удаления

Оператор **DELETE** предназначен для удаления группы записей из таблицы.

Формат оператора:

DELETE

FROM имя_таблицы [***WHERE***
<условие_отбора>]

*Пример. Удалить все
прошлогодние сделки.*

```
DELETE  
FROM Сделка  
WHERE Year(Сделка.  
Дата)=Year(GETDATE())-1
```

В данном примере условие отбора формируется с учетом года (функция Year) от текущей даты (функция GETDATE()).

Запрос обновления

Оператор **UPDATE** применяется для изменения значений в группе записей или в одной записи указанной таблицы.

Формат оператора:

UPDATE имя_таблицы

SET имя_столбца= [,...n]

[WHERE <условие_отбора>]

*Пример. Для товаров первого сорта
установить цену в значение 140 и остаток
– в значение 20 единиц.*

UPDATE Товар

SET Товар.Цена = 140, Товар.Остаток = 20


WHERE Товар.Сорт="Первый"

*Пример. Увеличить цену товаров
первого сорта на 25%.*

```
UPDATE Товар  
SET Товар.Цена = Товар.Цена*1.25  
WHERE Товар.Сорт="Первый"
```

Пример. В сделке с максимальным количеством товара увеличить число товаров на 10%.

```
UPDATE Сделка SET Сделка.Количество =  
Сделка.Количество*1.1 WHERE Сделка.  
Количество= (SELECT Max(Сделка.  
Количество) FROM Сделка)
```



Выполнение операторов модификации данных в таблицах базы данных INSERT, DELETE и UPDATE может привести к нарушению целостности данных и их корректности, т.е. к потере их достоверности и непротиворечивости. Ограничения целостности позволяют свести к минимуму ошибки, возникающие при обновлении и обработке данных.

Типы ограничений целостности данных:

- обязательные данные;
- ограничения для доменов полей;
- корпоративные ограничения;
- целостность сущностей;
- ссылочная целостность

Обязательные данные

Некоторые поля всегда должны содержать одно из допустимых значений, другими словами, эти поля не могут иметь пустого значения.

Ограничения для доменов полей

Каждое поле имеет свой домен, представляющий собой набор его допустимых значений.

Корпоративные ограничения целостности

- Это дополнительные правила поддержки целостности данных, определяемые пользователями, принятые на предприятии или администраторами баз данных. Ограничения предприятия называются бизнес–правилами.

Целостность сущностей

Это ограничение целостности касается первичных ключей базовых таблиц.

Целостность сущностей определяет, что в базовой таблице ни одно поле первичного ключа не может содержать отсутствующих значений, обозначенных NULL.

Ссылочная целостность

касается внешних ключей.

Внешний ключ – это поле (или множество полей) одной таблицы, являющееся ключом другой (или той же самой) таблицы.

Внешние ключи используются для установления логических связей между таблицами. Связь устанавливается путем присвоения значений внешнего ключа одной таблицы значениям ключа другой