

# Введение в язык SQL

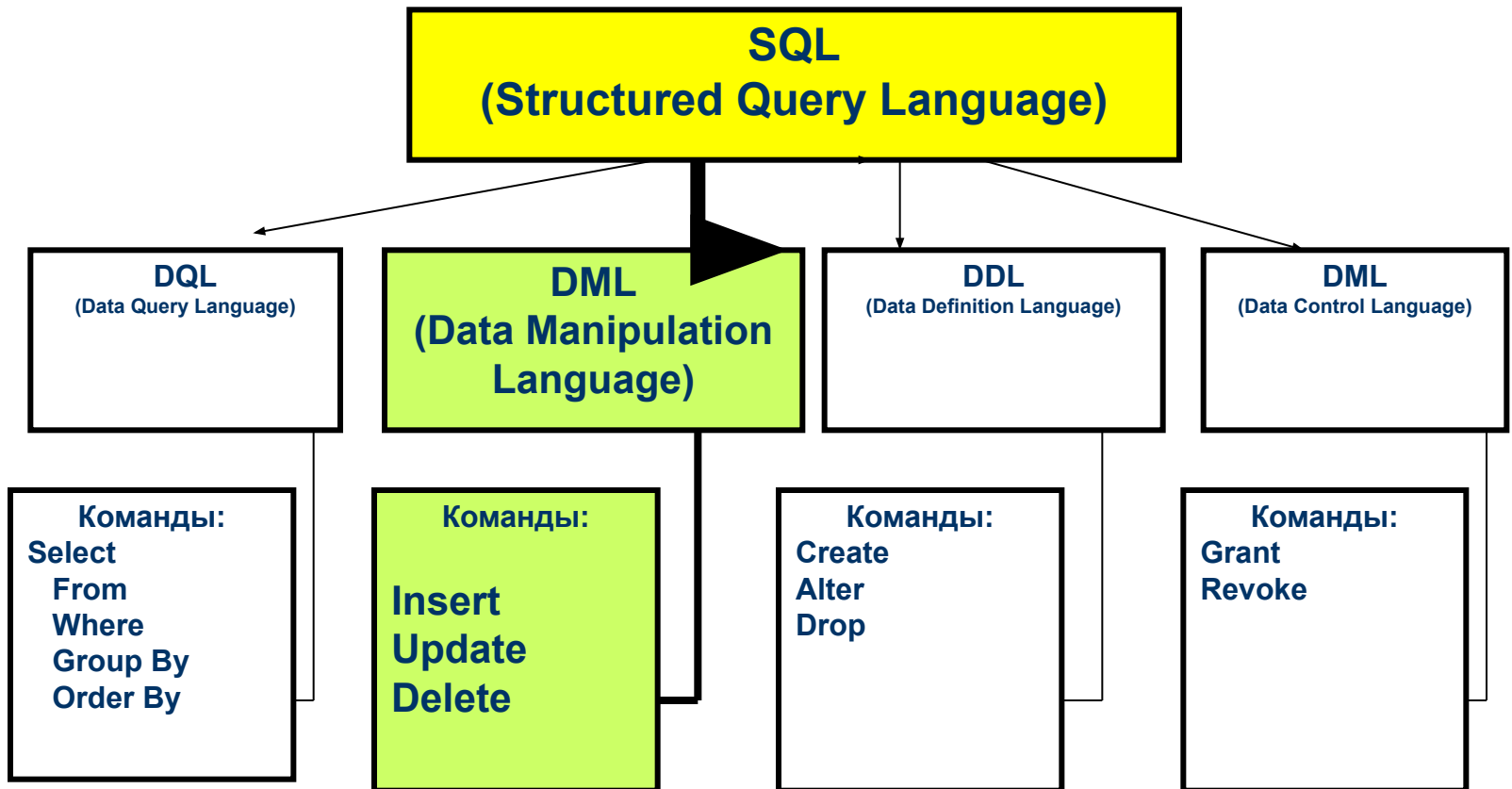
## 4. Запросы на ввод и корректировку данных в таблицах базы.



## 4.1. Запросы на ввод, корректировку и удаление данных в таблицах

- Все команды манипулирования данными объединены в разделе языка **DML (Data Manipulation Language)**, который включает следующие команды:  
INSERT (вставить),  
UPDATE(изменить) и  
DELETE (удалить).
- **Замечание:** Язык SQL не приспособлен для организации удобного ввода данных в таблицы, хотя в его состав включены языковые конструкции, позволяющие выполнить все необходимые операции по манипуляции данными в таблицах.

# Раздел DML (Data Manipulation language)



# Команда INSERT (вставить данные)

- Структура команды для вставки данных в таблицы:

**INSERT INTO** <имя таблицы> [<список столбцов>]

**VALUES** (<значение 1>, <значение 2>, ..., <значение N>);

Вставляет строку с данными в таблицу

# Пример использования INSERT

- Пример вставки новой записи с данными. Например, надо добавить нового продавца в таблицу Salespeople со следующими параметрами:
  - код продавца - 1010,
  - фамилия - Бояринов,
  - адрес - Москва,
  - комиссионные - 0.12,
  - телефона - нет,
  - почта - bojar @yandex.ru.

Это можно сделать следующим запросом на вставку:

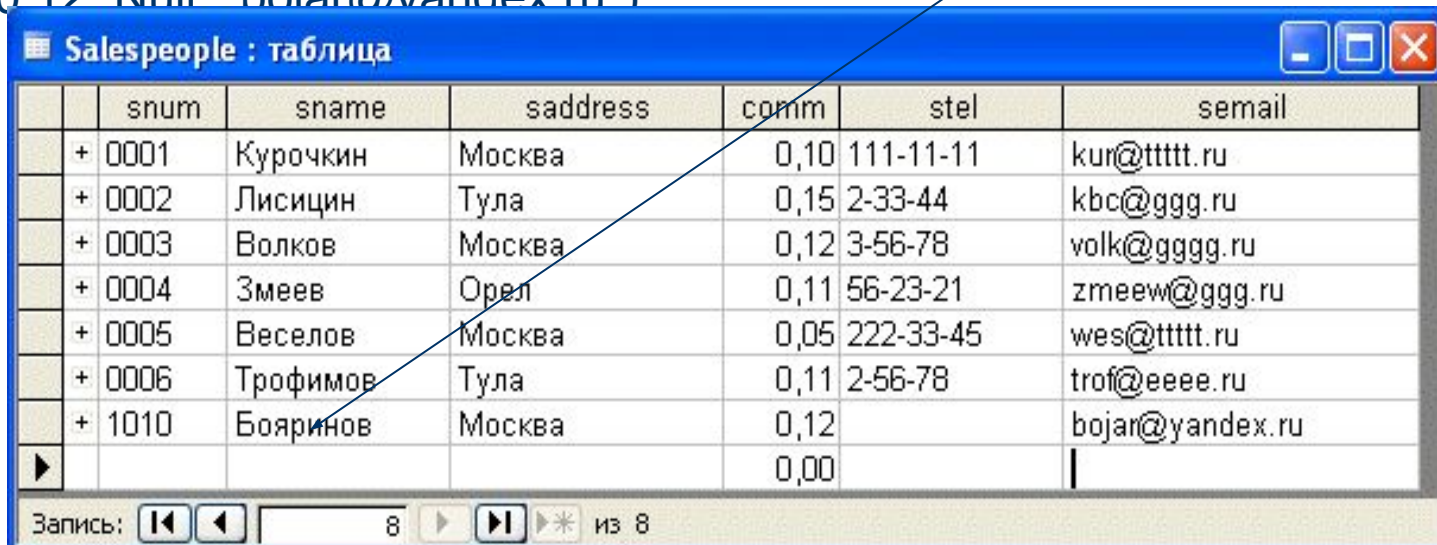
```
INSERT INTO Salespeople
```

```
VALUES ("1010", "Бояринов", "Москва", 0.12, Null,  
"bojar@yandex.ru");
```

# Результат

- Пример ввода строки данных в таблицу с помощью команды **INSERT INTO Salespeople VALUES ("1010", "Бояринов", "Москва", 0,12, Null, "boiar@yandex.ru"):**

Новая строка



	snum	sname	saddress	comm	stel	semail
+ 0001	0001	Курочкин	Москва	0,10	111-11-11	kur@tttt.ru
+ 0002	0002	Лисицин	Тула	0,15	2-33-44	kbc@ggg.ru
+ 0003	0003	Волков	Москва	0,12	3-56-78	volk@gggg.ru
+ 0004	0004	Змеев	Орел	0,11	56-23-21	zmeew@ggg.ru
+ 0005	0005	Веселов	Москва	0,05	222-33-45	wes@tttt.ru
+ 0006	0006	Трофимов	Тула	0,11	2-56-78	trof@eeee.ru
+ 1010	1010	Бояринов	Москва	0,12		boiar@yandex.ru
▶				0,00		

Запись: 8 из 8

# Вставка по выборочным столбцам

- В некоторых случаях можно использовать имена столбцов для ввода данных в конкретные поля. Например, если нам надо ввести нового покупателя, о котором известна пока только его фамилия. Тогда для этого можно использовать следующую команду:

```
INSERT INTO Customers (cname, cnum)  
VALUES ("Синичкин", "2007");
```

В результате выполнения запроса в таблицу покупателей добавится новая строка с кодом «2007» (сnum ="2007"), у которой будет заполнен только столбец cname (cname="Синичкин").

# Ввод данных по запросу

- Можно осуществить вставку данных в текущую таблицу, путем выбора данных по запросу из другой таблицы.

Например, имеется отдельная таблица «Жители Москвы» (“CitizenOfMoscow”), в которую мы хотим скопировать всех покупателей, которые живут в городе «Москва».

```
SELECT * FROM Customers WHERE City="Москва"
```

- Для переноса данных в эту таблицу можно записать и выполнить запрос следующего вида



# Запрос на перенос данных

- **INSERT INTO** CitizenOfMoscow  
SELECT \* FROM Customers WHERE city="Москва";

**Замечание:** Корректное выполнение этого запроса предполагает, что таблица CitizenOfMoscow создана до начала его выполнения и структура таблицы CitizenOfMoscow совпадает со структурой таблицы Customers

# Пример добавления столбца в таблицу

- Такой режим использования команды INSERT часто используется при обновлении структуры таблицы, когда требуется добавить новый столбец к уже существующей таблице. Обычно, для этого можно использовать команду, которая обеспечивает изменение структуры таблицы:

ALTER TABLE < имя таблицы > ADD <тип данных><размер>.

# Аналог команды ALTER

- Но, команда ALTER не является стандартной для ANSI, поэтому на практике в некоторых СУБД (например, в Access) ее приходится заменять набором команд. Так, например, набор команд на обновление структуры таблицы может состоять из следующих команд:
  - 1) создание новой таблицы (CREATE),
  - 2) перенос данных в новую таблицу (INSERT совместно с SELECT) и
  - 3) удаление старой таблицы (DROP).

## 4.2. Запросы на изменение данных в таблицах

- Изменение значений данных в полях производится с помощью команды UPDATE, которая имеет следующую конструкцию:

```
UPDATE <имя таблицы>  
    SET   <имя столбца 1>=<значение>,  
          <имя столбца 2>=<значение>,  
    ...,  
          <имя столбца N>=<значение>,  
WHERE <условие>
```

# Запрос на изменение (UPDATE)

- Запрос на изменение в Access удобно строить с помощью конструктора запросов в два этапа:  
Сначала построить простой запрос с условием для отбора строк. Посмотреть результаты отбора и, если, отобранные строки полностью удовлетворяют вашим условиям, трансформировать исходный запрос на запрос в команду UPDATE (через режим Запрос→Обновление).
- Пример: Предположим, мы хотим всем покупателям, которые живут в городе «Москва», изменить рейтинг на 125. Выполним запрос на отбор строк с продавцами, которые живут в городе «Москва»:

```
SELECT * FROM Customers WHERE caddress = "Москва";
```

# Трансформация запроса SELECT в UPDATE

- В результате отберутся только две строчки. Теперь выполним команду:  
**Запрос**→**Обновление**

Теперь снова войдем в режим просмотра SQL-запросов.

Запрос на обновление будет выглядеть, как показано ниже:

```
UPDATE Customers SET  
WHERE (((Customers.caddress) = "Москва"));
```

- В оператор SET добавим имя столбца (raiting) и новое значение (125) и получим полный запрос на обновление содержимого столбца raiting:

```
UPDATE Customers
SET    raiting=125
WHERE caddress="Москва";
```

- Таким образом, используя конструктор и просматривая запросы в режиме SQL, можно строить достаточно сложные и корректные запросы на обновление.

## 4.3. Запросы на удаление данных из таблиц

- Аналогичным образом строятся и запросы на удаление записей из таблицы. Общая форма запроса на удаление имеет вид:

**DELETE \***

**FROM** <имя таблицы>

**WHERE** <условия отбора записей>



# Пример запроса на удаление

- Напишем запрос на удаление всех продавцов, которые живут в городе «Тула». Для этого сначала запишем запрос, который отбирает все записи с продавцами, у которых выполняется условие `saddress = "Тула"`:

```
SELECT * FROM Salespeople WHERE saddress = "Тула" ;
```

- В результате выйдут только две строки. Теперь войдем в режим конструкторов запросов и выберем команду: **Запрос**→**Удаление**. Запрос на отбор строк трансформируется в запрос на удаление и примет вид:

```
DELETE *  
FROM Salespeople  
WHERE saddress = "Тула";
```

- После выполнения запроса ранее отобранные две строки будут удалены из таблицы `Salespeople`

# Заключение

- Раздел DML ( **Data Manipulation Language**) языка SQL состоит из команд:

**INSERT** – вставка данных в таблицу

**UPDATE** – изменение данных в таблице

**DELETE** – удаление данных из таблиц