

Администрирование баз данных



MySQL™

Изменение данных в таблицах

Оператор

UPDATE

Операция обновления позволяет менять значения полей в уже существующих записях.

```
update tbl_name
```

```
set column1=expr1, [column2=expr2, ...column=exprN]
```

```
[WHERE where_definition]
```

tbl_name – таблица, которая подвергается изменению
в предложении **set** указываются столбцы, которые изменяются, а также устанавливается их значение

Необязательное условие **WHERE** позволяет задать критерий отбора строк – обновлению подвергаются только те строки, которые соответствуют критерию отбора

Используем базу **lesson**

3. Добавить в таблицу **cars** колонку **reg_price**

```
alter table cars add reg_price decimal(10,2);
```

Необходимо обновить колонку **reg_price**, установив значение равное 78.50 (обновляется вся колонка целиком)

```
update cars set reg_price=78.50;
```

Проверить

```
select * from cars;
```

Обновление записей, удовлетворяющих определенным условиям.

В таблице cars всем машинам марки BMW 2010 года поменять пробег на 120000

```
update cars set km=120000 where make='BMW' and reg_year=2010;
```

Проверить

результат

```
select * from cars where make='BMW' and reg_year=2010;
```

Ограничение записей, подвергающихся изменению, с помощью инструкций *ORDER BY* и *LIMIT*.

В таблице cars необходимо цену регистрации (reg_price) для 5 самых дорогих машин установить 0

```
update cars set reg_price=0
ORDER BY price desc
LIMIT 5;
```

Проверить

результат

```
select * from cars
order by price
limit 5;
```

Обновление нескольких столбцов таблицы. Обновление столбцов результатами математических действий над исходными данными.

В таблице cars для машин Volkswagen старше 2012 года уменьшить цену на 300 и цену регистрации (reg_price) установить 15

```
update cars set price=price-300, reg_price=15  
where make='VOLKSWAGEN' and reg_year<2012;
```

Обновляемые колонки и их новые значения объявляются через запятую

price=price-300 каждое значение строки в колонке price принимает значение равное значению строки минус 300

Удаление данных в таблицах

Оператор **DELETE**

Оператор
TRUNCATE

Оператор

DELETE

`DELETE from tbl_name`

`WHERE where_definition`

`[ORDER BY ...]`

`[LIMIT]`

Оператор **DELETE** удаляет из таблицы *tbl_name* строки, которые удовлетворяют условию *where_definition*

```
delete from cars where make='CITROEN';
```

Оператор

TRUNCATE

Оператор **TRUNCATE** полностью очищает содержимое таблицы и не допускает условного удаления.

Оператор **TRUNCATE** идентичен оператору **DELETE** без условия **WHERE**. **TRUNCATE** работает быстрее, чем **DELETE**, т.к. не происходит перебор каждой записи.

```
create table cars_delete as select * from cars;  
truncate table cars_delete;  
drop table cars_delete;
```