

Ликбез по операциям UPDATE, INSERT, DELETE

1. INSERT (Добавление данных)

Базовый синтаксис:

```
INSERT INTO table_name (column1, column2, ...)  
VALUES (value1, value2, ...);
```

Примеры:

Добавление покупателя:

```
INSERT INTO customers (birth_dt, first_name, last_name, phone, email, gender)  
VALUES ('1990-05-15', 'Иван', 'Петров', '9123456789', 'ivan@mail.ru', 1);
```

Добавление транзакции с UUID:

```
INSERT INTO transactions (id, dt, customer_id, store_id, qnty, cheque_sum)  
VALUES (gen_random_uuid(), NOW(), 1, 1, 3, 1500.50);
```

Массовое добавление:

```
INSERT INTO categories (name) VALUES  
( 'Электроника' ),  
( 'Одежда' ),  
( 'Продукты' ),  
( 'Книги' );
```

INSERT с SELECT (копирование данных):

```
INSERT INTO transactions_archive  
SELECT * FROM transactions  
WHERE dt < '2023-01-01';
```

2. UPDATE (Обновление данных)

Базовый синтаксис:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Примеры:

Обновление email покупателя:

```
UPDATE customers
SET email = 'new_email@mail.ru'
WHERE id = 1;
```

Обновление с вычислением:

```
-- Увеличить все цены на 10%
UPDATE products
SET price = price * 1.1;
```

Обновление с JOIN:

```
-- Обновить магазины в определенном городе
UPDATE stores
SET frmt = 3
FROM cities
WHERE stores.city_id = cities.id
      AND cities.name = 'Москва';
```

Обновление с подзапросом:

```
-- Закрыть магазины без транзакций за последний год
UPDATE stores
SET close_dt = CURRENT_DATE
WHERE id NOT IN (
    SELECT DISTINCT store_id
    FROM transactions
    WHERE dt >= CURRENT_DATE - INTERVAL '1 year'
);
```

3. DELETE (Удаление данных)

Базовый синтаксис:

```
DELETE FROM table_name
WHERE condition;
```

Примеры:

Удаление конкретного покупателя:

```
DELETE FROM customers
WHERE id = 1;
```

Удаление по дате:

```
-- Удалить старые транзакции
DELETE FROM transactions
WHERE dt < '2020-01-01';
```

Удаление с JOIN:

```
-- Удалить магазины в определенном городе
DELETE FROM stores
USING cities
WHERE stores.city_id = cities.id
AND cities.name = 'Каменноостровский';
```

Удаление с подзапросом:

```
-- Удалить покупателей без транзакций
DELETE FROM customers
WHERE id NOT IN (SELECT DISTINCT customer_id FROM transactions);
```

4. Критически важные правила

✅ ВСЕГДА используйте WHERE в UPDATE/DELETE:

```
-- ОПАСНО! Удалит ВСЕХ покупателей:
DELETE FROM customers;
```

```
-- БЕЗОПАСНО! Удалит только одного:
DELETE FROM customers WHERE id = 1;
```

✅ Проверяйте что обновляете/удаляете:

```
-- Сначала посмотреть
SELECT * FROM customers WHERE last_name = 'Иванов';

-- Потом удалять
DELETE FROM customers WHERE last_name = 'Иванов';
```

✅ Используйте транзакции для групп операций:

```
BEGIN;
DELETE FROM transactions_details WHERE customer_id = 1;
DELETE FROM transactions WHERE customer_id = 1;
DELETE FROM customers WHERE id = 1;
COMMIT;
-- или ROLLBACK если что-то пошло не так
```

Ключевые выводы:

1. **INSERT** - для добавления новых записей
2. **UPDATE** - для изменения существующих данных
3. **DELETE** - для удаления записей
4. **ВСЕГДА** используйте **WHERE** в **UPDATE/DELETE**
5. **ПРОВЕРЯЙТЕ** данные перед изменением
6. **ИСПОЛЬЗУЙТЕ** транзакции для групп операций
7. **ТЕСТИРУЙТЕ** на тестовых данных перед продакшеном