Store your data

APPLYING SQL TO REAL-WORLD PROBLEMS

Dmitriy (Dima) Gorenshteyn
Lead Data Scientist, Memorial Sloan Kettering Cancer Center

datacamp

Three ways to store your data

1. Create a TABLE using **new** data
2. Create a TABLE using **existing** data
3. Create a VIEW using **existing** data

Create a TABLE using new data

Columns in `address` table

- `address_id`
- `address`
- `district`
- `city`
- `postal_code`
- `phone`

Create a TABLE using new data

postal_code	distance
53182	3.4
15540	10.2
67912	1.9
81766	21
...	...

Create a TABLE using new data

1) Create a new table

```
CREATE TABLE zip_distance (
    postal_code INT,
    distance FLOAT
);
```

2) Insert data into table

```
INSERT INTO zip_distance (postal_code, distance)
VALUES
(53182, 3.4),
(15540, 10.2),
(67912, 1.9);
```

Create a TABLE using existing data

```
SELECT film_id, title
FROM film
WHERE rating = 'G';
```

Create a TABLE using existing data

```
CREATE TABLE family_films AS
SELECT film_id, title
FROM film
WHERE rating = 'G';
```

Create a VIEW using existing data

```
CREATE VIEW family_films AS
SELECT film_id, title
FROM film
WHERE rating = 'G';
```

TABLE vs VIEW

TABLE

- Data is stored (static)
- Data can be modified directly

VIEW

- Query is stored (dynamic)
- Underlying data must be modified in original tables

Time to store your data.

APPLYING SQL TO REAL-WORLD PROBLEMS

Update your data

APPLYING SQL TO REAL-WORLD PROBLEMS

Dmitriy (Dima) Gorenshteyn
Lead Data Scientist, Memorial Sloan Kettering Cancer Center

datacamp

UPDATE syntax

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

UPDATE a column

Desired Update: Emails of customers must be lowercase.

```
UPDATE customer
SET email = LOWER(email);
```

UPDATE & WHERE

Desired Update: Emails of customers must be lowercase for customers who are still active.

```
UPDATE customer
SET email = LOWER(email)
WHERE active = TRUE;
```

UPDATE using subqueries

Desired Update: Emails of customers must be lowercase for customers reside in city of Woodridge.

```
UPDATE customer
SET email = LOWER(email)
WHERE address_id IN
  (SELECT address_id
   FROM address
   WHERE city = 'Woodridge');
```

Be careful when modifying tables

- Ensure you CAN modify the table.
- Ensure you know how this table is used and how your changes will impact those who use it.
- Test a modification by using a `SELECT` statement first.

Let's UPDATE

APPLYING SQL TO REAL-WORLD PROBLEMS

Delete your data

APPLYING SQL TO REAL-WORLD PROBLEMS

Dmitriy (Dima) Gorenshteyn
Lead Data Scientist, Memorial Sloan Kettering Cancer Center

DROP, TRUNCATE, DELETE

Remove a table

```
DROP TABLE table_name;
```

Clear table of ALL records

```
TRUNCATE TABLE table_name;
```

Clear table of SOME records

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

DELETE inactive customers

Desired Modification: Remove customers who are no longer active

```
DELETE FROM customer
WHERE active = FALSE;
```

DELETE using a subquery

Desired Modification: Removing all customers who live in the city of Woodridge.

```
DELETE FROM customer
WHERE address_id IN
  (SELECT address_id
   FROM address
   WHERE city = 'Woodridge');
```

Let's DELETE some records!

APPLYING SQL TO REAL-WORLD PROBLEMS