

04_Modifying database with SQL

October 30, 2022

1 Using SQL to update a database

We can think of some uses of SQL as “read-only operations” and other uses as “read/write operations”. An example of a “read-only operation” is a data analysis on a data dump from some app or research study. An example of “read/write operations” is a software engineer creating the backend for webapp.

The write operations are necessary if we’re going to use a database to store and update data for an app. **INSERT** is relatively safe, because all it does is add data, but operations like **UPDATE**, **DELETE**, **DROP**, or **ALTER** can be much more dangerous, because they are updating existing data.

2 Changing rows with UPDATE and DELETE

The **UPDATE** statement is used to modify the existing records in a table. Be careful when deleting and updating records in table! Notice the **WHERE** clause in the **DELETE** statement. The **WHERE** clause specifies which record should be updated.

```
CREATE TABLE users (  
    id INTEGER PRIMARY KEY,  
    name TEXT);
```

```
CREATE TABLE diary_logs (  
    id INTEGER PRIMARY KEY,  
    user_id INTEGER,  
    date TEXT,  
    content TEXT  
);
```

```
/* After user submitted their new diary log */
```

```
INSERT INTO diary_logs (user_id, date, content) VALUES (1, "2015-04-01",  
    "I had a horrible fight with OhNoesGuy and I buried my woes in 3 pounds of dark chocolate.");
```

```
INSERT INTO diary_logs (user_id, date, content) VALUES (1, "2015-04-02",  
    "We made up and now we're best friends forever and we celebrated with a tub of ice cream.");
```

```
/* UPDATE clause */
```

```
UPDATE diary_logs SET content = "I had a horrible fight with OhNoesGuy"  
    WHERE user_id=1 AND date="2015-04-01";
```

The **DELETE** statement is used to delete existing records in a table.

```
/* DELETE clause */  
DELETE FROM diary_logs WHERE user_id=1 AND date="2015-04-01";
```

3 Altering tables after creation

The **ALTER TABLE** command adds, deletes, or modifies columns in a table. And the **ALTER COLUMN** command is used to change the data type of a column in a table.

```
/* What we used to originally create the table */  
CREATE TABLE users (  
    id INTEGER PRIMARY KEY,  
    name TEXT);  
  
CREATE TABLE diary_logs (  
    id INTEGER PRIMARY KEY,  
    user_id INTEGER,  
    date TEXT,  
    content TEXT  
);  
  
/* After user submits a diary log */  
INSERT INTO diary_logs (user_id, date, content) VALUES (1, "2015-04-02",  
    "OhNoesGuy and I made up and now we're best friends forever and we celebrated with a tub o  
  
/* Add column to existing table using ALTER TABLE */  
ALTER TABLE diary_logs ADD emotion TEXT default "Unknown";  
  
INSERT INTO diary_logs (user_id, date, content, emotion) VALUES (1, "2015-04-02",  
    "OhNoesGuy and I made up and now we're best friends forever and we celebrated with a tub o  
  
/* Drop existing table */  
DROP TABLE diary_logs;
```

4 Make our SQL safer

4.1 Avoiding bad updates/deletes

Before we issue an UPDATE, run a SELECT with the same WHERE to make sure we're updating the right column and row.

Before running :

```
UPDATE users SET deleted = true WHERE id = 1;
```

We could run :

```
SELECT id, deleted FROM users WHERE id = 1;
```

4.2 Using transactions

A **transaction** is a sequence of operations treated as a single logical piece of work, and in the world of databases, a transaction must comply to the ACID principles to make sure the operations are processed reliably.

We can wrap multiple command in BEGIN TRANSACTION and COMMIT:

```
BEGIN TRANSACTION;
UPDATE people SET husband = "Winston" WHERE user_id = 1;
UPDATE people SET wife = "Winnefer" WHERE user_id = 2;
COMMIT;
```

If the database is unable to issue both those UPDATE command for some reason, then it will rollback the transaction and leave the database how it was when it started.

4.3 Making backups

Most companies make backups of their databases on an hourly, daily, or weekly basis, depending on the size of the database and space available.

4.4 Replication

A related approach is replication - always storing multiple copies of the databases in different places. However, it is a lot more effort to replicate database and it often means slower performance since write operations have to be performed in all of them.

4.5 Granting privileges

Many database systems have users and privileges built into them, because they are stored on a server and accessed by multiple users. As a general rule, there should be only a few users that have full access to the database, since it can be so dangerous.

5 Project : App impersonator

```
/* What does the app's SQL look like? */
```

```
CREATE TABLE games (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  title TEXT,
  scores INTEGER,
  post TEXT
);
```

```
INSERT INTO games (title, scores, post) VALUES
('Doom', 9, '2022-09-14'),
('Scar', 8, '2021-08-12'),
('Iron Heart', 10, '2022-02-22');
```

```
UPDATE games SET scores=7 WHERE title='Iron Heart';
```

```
SELECT * FROM games;
```

```
DELETE FROM games WHERE scores < 9;
```

```
SELECT * FROM games;
```