

SQL in Postgres

Insert a row into a table

Insert command follows with

INSERT IN [table name] (key name) **VALUES** (one-to-one correspondence);

SQL: Insert

The **INSERT** statement inserts a row into a table

```
INSERT INTO users (name, email) VALUES ('Chuck', 'csev@umich.edu') ;
INSERT INTO users (name, email) VALUES ('Somesh', 'somesh@umich.edu') ;
INSERT INTO users (name, email) VALUES ('Caitlin', 'cait@umich.edu') ;
INSERT INTO users (name, email) VALUES ('Ted', 'ted@umich.edu') ;
INSERT INTO users (name, email) VALUES ('Sally', 'sally@umich.edu') ;
```

Delete a row with criteria

DELETE FROM [table name] **WHERE** [criteria];

If we delete the **WHERE** clause, it will **delete all rows** from the table

It's similar to "loop + if" statement

SQL: Delete

Deletes a row in a table based on selection criteria

```
DELETE FROM users WHERE email='ted@umich.edu';
```

Update a field (value) with a WHERE Clause

UPDATE [table name] **SET** [field = value] **WHERE** [criteria];

This command applies to all records satisfying the given criteria

SQL: Update

Allows the updating of a field with a **WHERE** clause

```
UPDATE users SET name='Charles' WHERE email='csev@umich.edu';
```

Retrieving records from a table

```
SELECT * FROM [table name] WHERE [criteria]
```

Retrieving Records: Select

Retrieves a group of records - you can either retrieve all the records or a subset of the records with a **WHERE** clause

```
SELECT * FROM users;
```

```
SELECT * FROM users WHERE email='csev@umich.edu';
```

Sorting with ORDER BY

```
SELECT * FROM [table name] ORDER BY [field]
```

Sorting with ORDER BY

You can add an **ORDER BY** clause to **SELECT** statements to get the results sorted in ascending or descending order

```
SELECT * FROM users ORDER BY email;
```

Wildcard matching

```
SELECT * FROM [table name] WHERE [field] LIKE [%letter%];
```

The given SQL is for finding the name contains “e” in anywhere.

The wildcard matching is **slow** since it can not use the advantage of the “index” from the database

The LIKE Clause

We can do wildcard matching in a **WHERE** clause using the **LIKE** operator

```
SELECT * FROM users WHERE name LIKE '%e%';
```

Search data within a given range

```
SELECT * FROM [table name] ORDER BY [field] LIMIT [number]
```

This command only requires certain records from the database.

```
SELECT * FROM [table name] ORDER BY [field] OFFSET [number] LIMIT [number]
```

This command would skip the first n records.

The LIMIT/OFFSET Clauses

- We can request the first "n" rows, or the first "n" rows after skipping some rows.
- The **WHERE** and **ORDER BY** clauses happen *before* the **LIMIT / OFFSET** are applied.
- The **OFFSET** starts from row 0

```
SELECT * FROM users ORDER BY email DESC LIMIT 2;  
SELECT * FROM users ORDER BY email OFFSET 1 LIMIT 2;
```

Counting rows with SELECT

`SELECT COUNT(*) FROM [table name]`

Counting Rows with SELECT

You can request to receive the **count** of the rows that would be retrieved instead of the rows

```
SELECT COUNT(*) FROM users;  
SELECT COUNT(*) FROM users WHERE email='csev@umich.edu';
```

Set required fields

Create table films and table distributors:

```
CREATE TABLE films (  
  code      CHARACTER(5) CONSTRAINT firstkey PRIMARY KEY,  
  title     CHARACTER VARYING(40) NOT NULL,  
  did       DECIMAL(3) NOT NULL,  
  date_prod DATE,  
  kind      CHAR(10),  
  len       INTERVAL HOUR TO MINUTE  
);
```

Delete existing table

DROP TABLE

Name _____

DROP TABLE -- remove a table

Synopsis _____

```
DROP TABLE [ IF EXISTS ] name [, ...] [ CASCADE | RESTRICT ]
```

Summary

SQL Summary

```
INSERT INTO users (name, email) VALUES ('Ted', 'ted@umich.edu');
```

```
DELETE FROM users WHERE email='ted@umich.edu';
```

```
UPDATE users SET name='Charles' WHERE email='csev@umich.edu';
```

```
SELECT * FROM users WHERE email='csev@umich.edu';
```

```
SELECT * FROM users ORDER BY email;
```

```
SELECT * FROM users WHERE name LIKE '%e%';
```

```
SELECT * FROM users ORDER BY email OFFSET 1 LIMIT 2;
```

```
SELECT COUNT(*) FROM users WHERE email='csev@umich.edu'
```