Class 4 Postgres

Elements of Databases

Sep 15, 2023

Announcements

Preparing for Midterm 1:

- End-of-chapter exercises (requires Sakila sample database)
- Practice SQL on Hacker Rank
- Practice SQL on Leetcode

On the horizon:

- BigQuery starting next week (no setup needed)
- Review session for Midterm 1 (week of Oct 2nd)

Homework: remodeled college tables

1. Connect to your Postgres database.
2. Run this query: `select count(*) from college.takes;`
3. Answer the instapoll.

Review Exercise: SQL Joins

Who are the students who take CS329E with Prof. Mitra?

For each student, return their sid, first and last names, and grade sorted by their sid.

Schema:

Student(<u>sid</u>, fname, lname, dob, status)

Class(<u>cno</u>, cname, credits)

Instructor(<u>tid</u>, name, dept)

Takes(<u>*sid*</u>, <u>*cno*</u>, grade)

Teaches(<u>*tid*</u>, <u>*cno*</u>)

Exercise 1: SQL Joins

Who are the students who take both CS329E and CS327E?

For each student, return their sid and first and last names. Sort the results by sid.

Schema:

Student(sid, fname, lname, dob, status)

Class(cno, cname, credits)

Instructor(tid, name, dept)

Takes(*sid*, *cno*, grade)

Teaches(*tid*, *cno*)

Exercise 2: SQL Joins

Which classes have no students taking them (i.e. zero enrollment)?

For each class with zero enrollment, return the cno, cname and credits of the class. Sort the results by cno.

Schema:

Student(sid, fname, lname, dob, status)

Class(cno, cname, credits)

Instructor(tid, name, dept)

Takes(*sid*, *cno*, grade)

Teaches(*tid*, *cno*)

A World without Transactions

Time

	Client 1	Client 2

t_0	`UPDATE account`
	`SET balance = balance - 100`
	`WHERE name = 'Alice';`	

t_1		`SELECT name, balance`
		`FROM account`
		`WHERE name IN ('Alice', 'Bob');`

t_2	`UPDATE account`
	`SET balance = balance + 100`
	`WHERE name = 'Bob';`	

A World without Transactions

Time

	Client 1	Client 2

t_0	`UPDATE playlist`
	`SET count = count + 1`
	`WHERE user = 'Alice';`
		`UPDATE playlist`
		`SET count = count + 1`
		`WHERE user = 'Alice';`

t_1	`SELECT count`
	`FROM playlist`
	`WHERE user = 'Alice';`
		`SELECT count`
		`FROM playlist`
		`WHERE user = 'Alice';`

Transaction Blocks

```
BEGIN TRANSACTION;
   {some SQL statement 1}
   {some SQL statement 2}
   {some SQL statement n}
COMMIT;
```

```
BEGIN TRANSACTION;
   {some SQL statement 1}
   {some SQL statement 2}
   {some SQL statement n}
ROLLBACK;
```

Transaction Guarantees

- Atomicity
- Consistency
- Isolation
- Durability

Postgres Code Lab, Part 2

- Clone snippets repo
- Open postgres tx notebook
- Create the Shopify tables and load them
- Sample the tables
- Create the Foreign Keys
- Walk through an insert transaction
- Write an update transaction
- Write a delete transaction

Project 3

http://www.cs.utexas.edu/~scohen/projects/project-3.pdf