Class 4 Postgres

Elements of Databases

Feb 18, 2022
Announcements

Preparing for Midterm 1:

• End-of-chapter exercises (requires Sakila sample database)
• Practice SQL on [Hacker Rank](https://www.hackerrank.com)
• Practice SQL on [Leetcode](https://leetcode.com)

On the horizon:

• BigQuery starting next week (no setup needed)
• Review session for Midterm 1 (week of March 7th)
Homework: remodeled college tables

1. Connect to your Postgres database.

2. Run this query: `select count(*) from college.takes;`

3. Answer the instapoll.
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(sid, fname, lname, dob, status)
Class(cno, cname, credits)
Instructor(tid, name, dept)
Takes(sid, cno, grade)
Teaches(tid, cno)
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?

For each class with zero enrollment, return its cno, cname and credits.
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

<table>
<thead>
<tr>
<th>Time</th>
<th>Client 1</th>
<th>Client 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_0$</td>
<td>UPDATE account</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SET balance = balance - 100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WHERE name = 'Alice';</td>
<td></td>
</tr>
<tr>
<td>$t_1$</td>
<td></td>
<td>SELECT name, balance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FROM account</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WHERE name IN ('Alice', 'Bob');</td>
</tr>
<tr>
<td>$t_2$</td>
<td>UPDATE account</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SET balance = balance + 100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WHERE name = 'Bob';</td>
<td></td>
</tr>
</tbody>
</table>
# A World without Transactions

<table>
<thead>
<tr>
<th>Time</th>
<th>Client 1</th>
<th>Client 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_0$</td>
<td>UPDATE playlist</td>
<td>UPDATE playlist</td>
</tr>
<tr>
<td></td>
<td>SET count = count + 1</td>
<td>SET count = count + 1</td>
</tr>
<tr>
<td></td>
<td>WHERE user = 'Alice';</td>
<td>WHERE user = 'Alice';</td>
</tr>
<tr>
<td></td>
<td>SELECT count</td>
<td>SELECT count</td>
</tr>
<tr>
<td></td>
<td>FROM playlist</td>
<td>FROM playlist</td>
</tr>
<tr>
<td></td>
<td>WHERE user = 'Alice';</td>
<td>WHERE user = 'Alice';</td>
</tr>
</tbody>
</table>
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