Announcements

• Rubric clarification
• Test 1 details

Exam rules:

• Open-note and open-book
• Piazza will be disabled during exam
• May not consult with any human in any form
# A World without Transactions

<table>
<thead>
<tr>
<th>Time</th>
<th>Client 1</th>
<th>Client 2</th>
</tr>
</thead>
</table>
| \( t_0 \) | 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

<table>
<thead>
<tr>
<th>Time</th>
<th>Client 1</th>
<th>Client 2</th>
</tr>
</thead>
</table>
| $t_0$ | `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 Properties

- Atomicity
- Consistency
- Isolation
- Durability
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;
Database Indexes

- **Critical** to database systems
- At least one index per table
- DBA analyzes workload and chooses which indexes to create (no easy answers)
- Creating indexes can be an expensive operation
- They work “behind the scenes”
- Query optimizer decides which indexes to use during query execution

```sql
CREATE INDEX empid_idx ON Employee(empid);
```
B-Trees

- Standard index implementation in relational databases
- Designed to speed up lookups and range queries
- One tree node maps to one disk page
- Nodes store index entries
- Index entry = (key, ref)
- Branching factor 100+
- Height is $O(\log n)$
- Search speed $\approx$ height of tree
Why Spanner?

- Globally distributed database system
- Regional and multi-regional configurations
- Implements relational model
- Standard SQL (+ table hierarchies)
- ACID transactions
- TrueTime assigns globally consistent time
- Compute and storage are decoupled
- Data splits assigned to Spanner nodes
- Splits based on load and data volume
- Massive scale (PBs, 1000+ nodes)
Set up Spanner (Emulator)

Practice Problem 1

Debug this query and then optimize it.

```sql
SELECT *, c.title
WHERE c.title = 'Productivity'
FROM categories c JOIN apps_categories
ON c.id = category_id
AND reviews_count >= 50
AND rating >= 4.0
JOIN apps ON id = app_id;
```
Practice Problem 2

Write a query to find all foreign key violations on the tables:

- pricing_plans
- key_benefits
Project 3