ECS 165A

Milestone 3

Team Waifus Forever

Our Team

Caroline Yau
Team Coordinator, Developer

Alvin Ho
System Architect, Developer

Peter Lo
System Architect, Developer

Alejandro Armas
Manager, System Architect, Developer, Tester

Roberto Lozano
System Architect, Developer

01

Overview

Design and Solution

Implementation

01 Transaction Semantics Overview

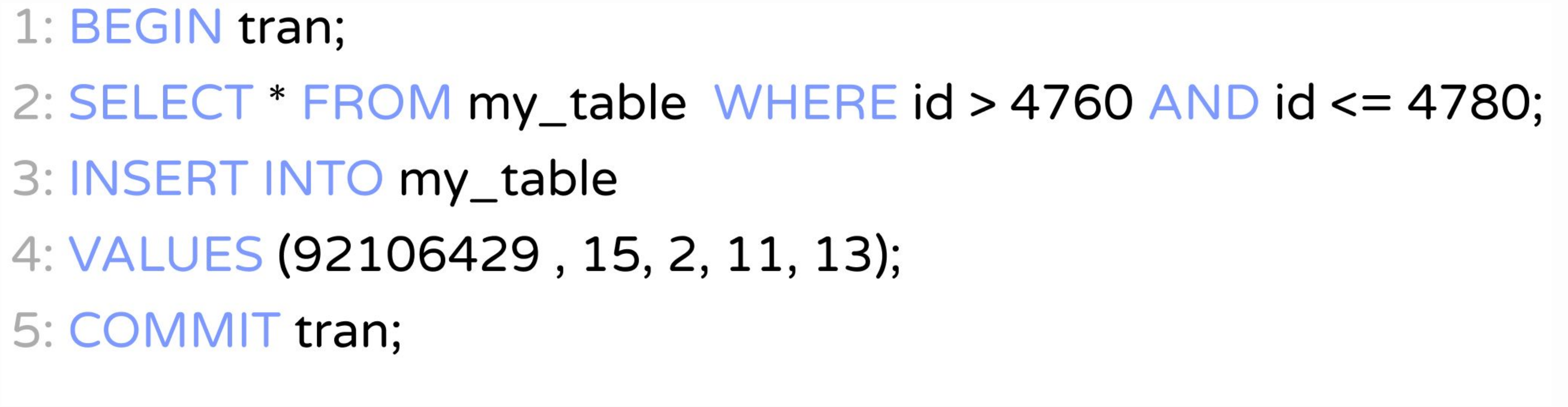
02 QueCC

03 MultiThreading

Bug Fixing

- lazy merge implemented
- restarting database now works
  - we previously could not perform certain queries
- multi-indexing

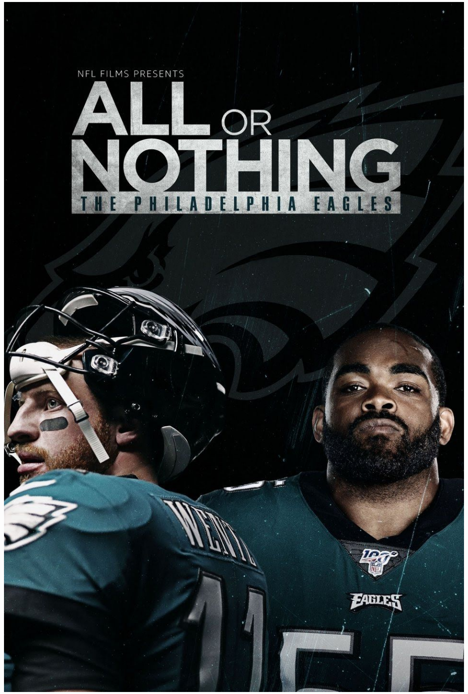
Transaction Semantics

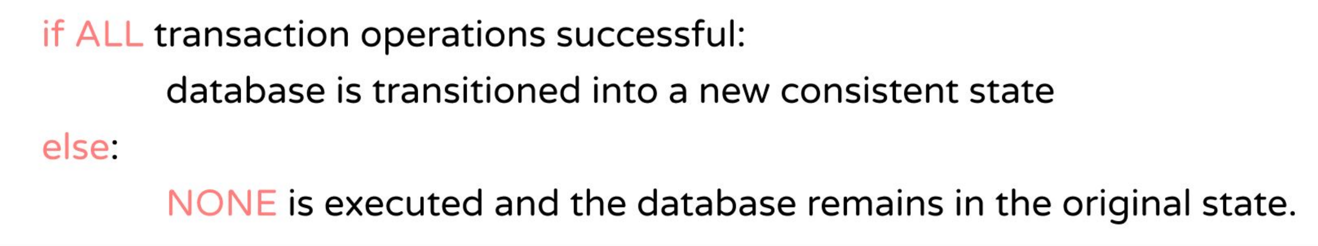


```
BEGIN tran;
SELECT * FROM my_table  WHERE id > 4760 AND id <= 4780;
INSERT INTO my_table
VALUES (92106429 , 15, 2, 11, 13);
COMMIT tran;
```

- set of operations over shared data that transforms the data from one consistent state to another.

Atomicity

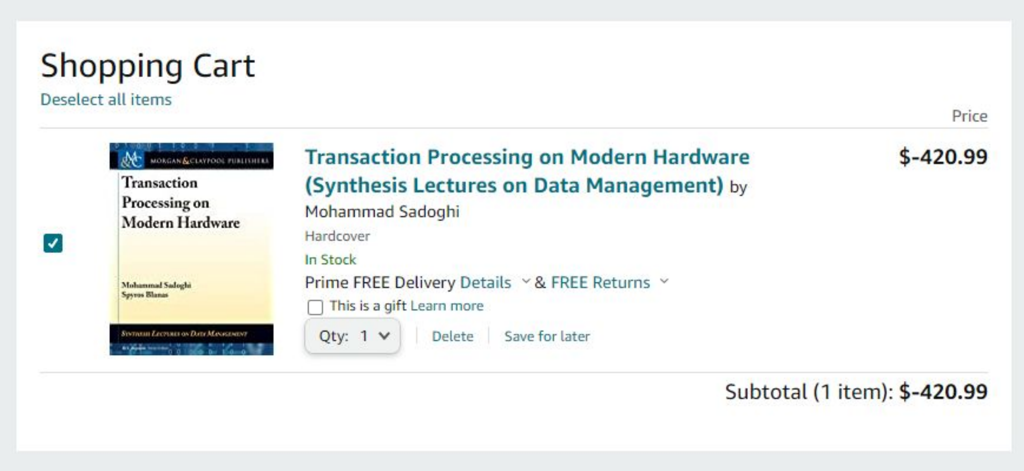




```
if ALL transaction operations successful:
        database is transitioned into a new consistent state
else:
        NONE is executed and the database remains in the original state.
```

Consistency

- integrity constraints set by users



MORGAN & CLAYPOOL PUBLISHERS
Transaction Processing on Modern Hardware
Mohammad Sadoghi
Spyros Blanas
Synthesis Lectures on Data Management

Shopping Cart

Deselect all items

Price

☑ Transaction Processing on Modern Hardware (Synthesis Lectures on Data Management) by Mohammad Sadoghi $-420.99

Hardcover

In Stock

Prime FREE Delivery Details & FREE Returns

□ This is a gift Learn more

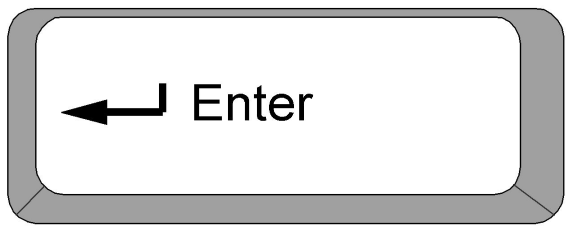
Qty: 1 | Delete | Save for later

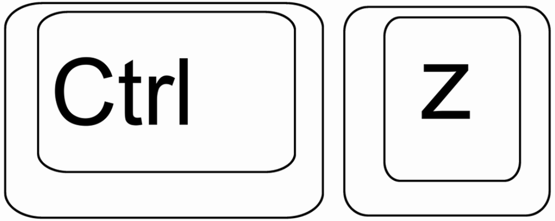
Subtotal (1 item): **$-420.99**

Isolation

- we need to avoid conflicting operations when we interleave concurrent transactions
- CC protocols facilitate coordination among transactions to ensure correct ordering of operations

Durability





- Achieved maintaining an ordered undo and/or redo actions
- Necessary for rolling back aborted transactions when dealing with weak isolation

Queue Oriented Control Free Concurrency[1]

Goal: Abandon complex concurrency:

- Hardware trends point to opportunities in leveraging parallelism
  - more contention

- simply execute transactions serially on disjoint partitions of data
  - H-Store introduced this idea[2]

- Exploit determinism through planning[3]

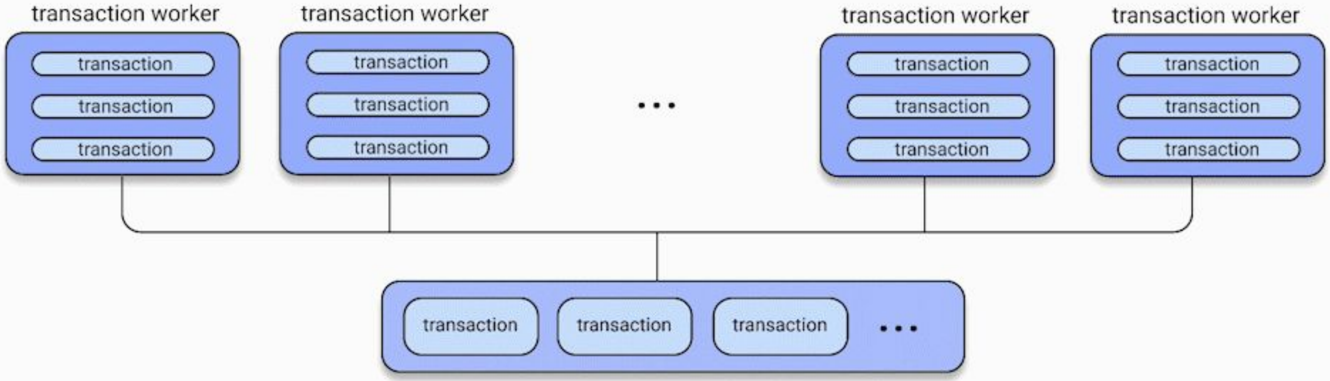
- Deterministic schemes eliminate all execution induced aborts
  - e.g. deadlocks

QueCC: A Queue-oriented, Control-free Concurrency Architecture [1]
Thamir M. Qadah, Mohammad Sadogh, 2018

H-store: A high performance, distributed main memory transaction processing system.[2]
R. Kallman, H. Kimura, J. Natkins, A. Pavlo, A. Rasin, S. Zdonik, E. P. C. Jones, S. Madden, M. Stonebraker, Y. Zhang, J. Hugg, and D. J. Abadi, 2008

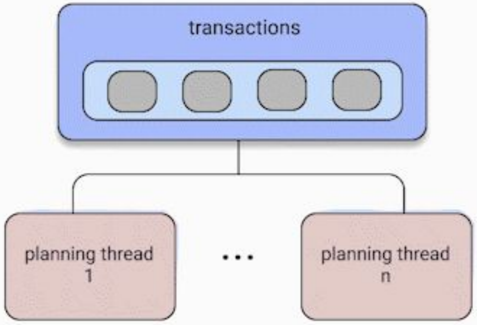
Calvin: Fast distributed transactions for partitioned database systems[3]
A. Thomson, T. Diamond, S. C. Weng, K. Ren, P. Shao, and D. J. Abadi, 2012

QueCC


transaction worker
transaction
transaction
transaction
transaction worker
transaction
transaction
transaction
...
transaction worker
transaction
transaction
transaction
transaction worker
transaction
transaction
transaction
transaction
transaction
transaction
...

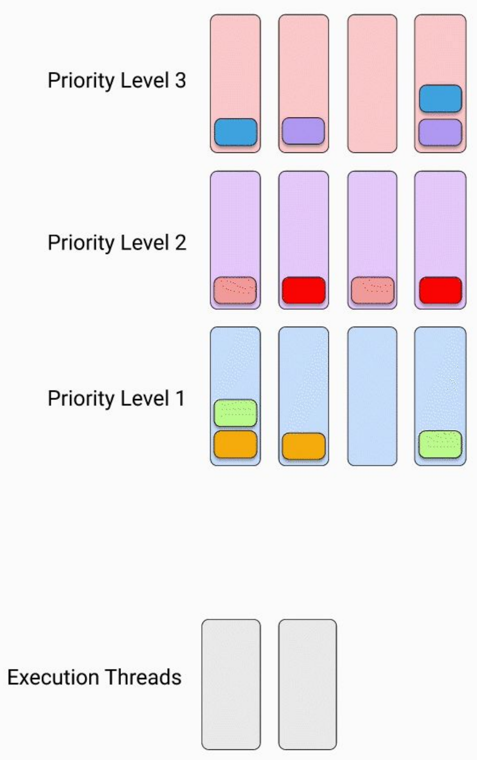
QueCC

Planning Stage

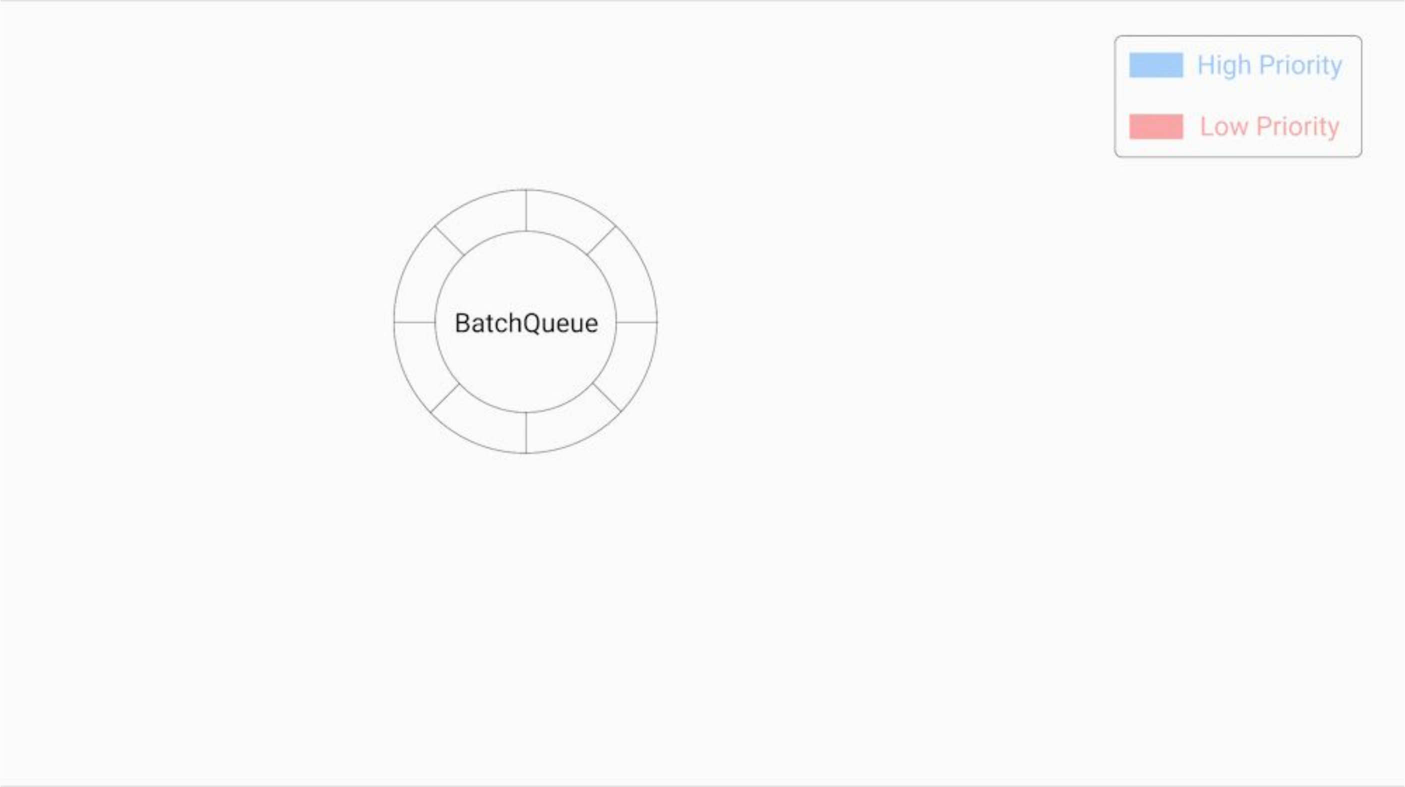


QueCC

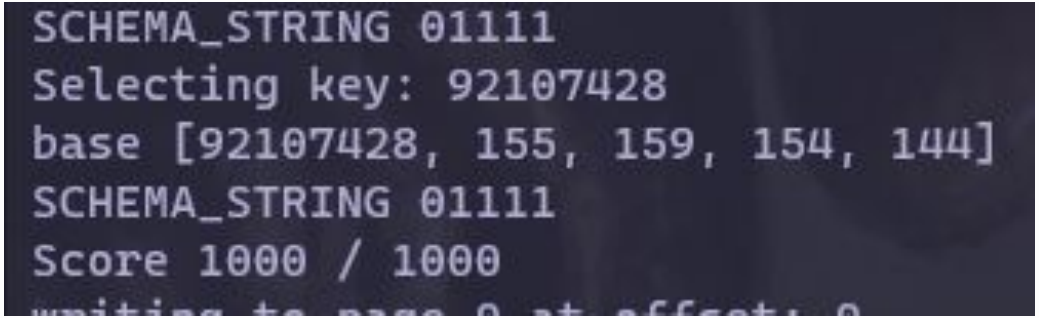
Execution Stage



Concurrent Batch Planning



Code Performance: Score



```
SCHEMA_STRING 01111
Selecting key: 92107428
base [92107428, 155, 159, 154, 144]
SCHEMA_STRING 01111
Score 1000 / 1000
writing to page 0 at offset: 0
```

02

Q/A

Questions about various aspects of the project

03

Demo

A live demonstration of the code