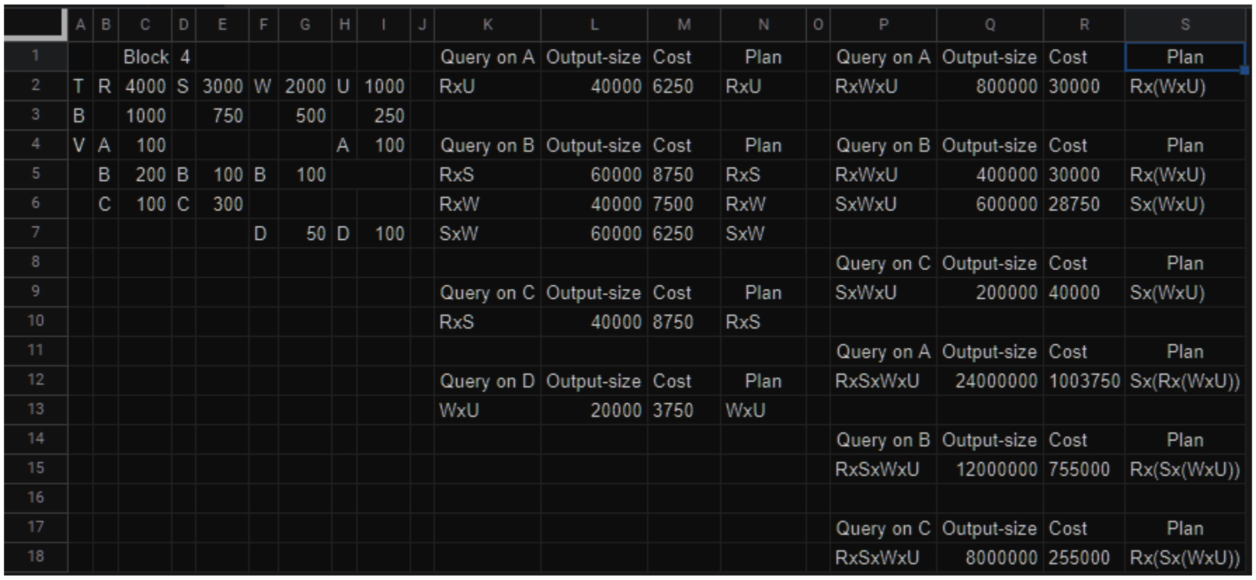
Adrien Protzel | protzela | protzela@oregonstate.edu
Curtiss Haldorson | haldorsc | haldorsc@oregonstate.edu

Due March 1st, Tuesday

1. Sort merge join
    a. Code

2. Query optimization



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | Block | 4 | | | | | | | Query on A | Output-size | Cost | Plan | | Query on A | Output-size | Cost | Plan |
| 2 | T | R | 4000 | S | 3000 | W | 2000 | U | 1000 | | RxU | 40000 | 6250 | RxU | | RxWxU | 800000 | 30000 | Rx(WxU) |
| 3 | B | | 1000 | | 750 | | 500 | | 250 | | | | | | | | | | |
| 4 | V | A | 100 | | | | | A | 100 | | Query on B | Output-size | Cost | Plan | | Query on B | Output-size | Cost | Plan |
| 5 | | B | 200 | B | 100 | B | 100 | | | | RxS | 60000 | 8750 | RxS | | RxWxU | 400000 | 30000 | Rx(WxU) |
| 6 | | C | 100 | C | 300 | | | | | | RxW | 40000 | 7500 | RxW | | SxWxU | 600000 | 28750 | Sx(WxU) |
| 7 | | | | | | D | 50 | D | 100 | | SxW | 60000 | 6250 | SxW | | | | | |
| 8 | | | | | | | | | | | | | | | | Query on C | Output-size | Cost | Plan |
| 9 | | | | | | | | | | | Query on C | Output-size | Cost | Plan | | SxWxU | 200000 | 40000 | Sx(WxU) |
| 10 | | | | | | | | | | | RxS | 40000 | 8750 | RxS | | | | | |
| 11 | | | | | | | | | | | | | | | | Query on A | Output-size | Cost | Plan |
| 12 | | | | | | | | | | | Query on D | Output-size | Cost | Plan | | RxSxWxU | 24000000 | 1003750 | Sx(Rx(WxU)) |
| 13 | | | | | | | | | | | WxU | 20000 | 3750 | WxU | | | | | |
| 14 | | | | | | | | | | | | | | | | Query on B | Output-size | Cost | Plan |
| 15 | | | | | | | | | | | | | | | | RxSxWxU | 12000000 | 755000 | Rx(Sx(WxU)) |
| 16 | | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | Query on C | Output-size | Cost | Plan |
| 18 | | | | | | | | | | | | | | | | RxSxWxU | 8000000 | 255000 | Rx(Sx(WxU)) |

Output size = (T(R)*T(S)) / MAX(V(R,A),V(S,A))
Costs = (5*B(R)) + (5*B(S))        B = T / Block[4]
Joined on A,B,C,D if they share the common key
Formulas are written in cells on page 2

3. Serializability and 2PL
    a. Serializable, but not 2PL. T1 releases it's lock on X, and then puts a new lock on Y. It violates reading X after writing but it is still valid.
    b. Serializable, T2 rereads Y after it was written in T3 which makes it not 2PL. In order for T3 to write to Y, T2 must have given up its shared lock.
    c. Cascade rollback. T1 starts with writing X with garbage data then T2 tries to read X that has already written X.
    d. Not serializable, 2PL, and causes a Cascade rollback. T1 writes over X from T2 and T3 tries to read X that was already written in T2 and T1.

4. Degrees of Consistency
    a. T1: violates 3 because T2 reads and tries to write over X and so T1 = 2.
    b. T2: violates 0 because it tries to read and write X after T1 has read and written it.T2 violates 2 because it reads and writes X that has been modified by T1. T2 = 0.