- CS348 assignment 4
- Name: chen Chen
- UWID: 20518383

- Question 1.
  - Query 1:

$\pi_{\{student.sid,\ student.Name\}}(\sigma((course.semester = 4 \lor course.semester = 5 \lor course.semester) \land (enrolled.semester = 1 \lor enrolled.semester = 2 \lor enrolled.semester = 3))(student \bowtie (student.Sid = enrolled.Sid)\ enrolled \bowtie (enrolled.Cid = course.Cid)\ course))$

  - Query 2:

idea: (Enrolled − (Enrolled₁ ⋈ Enrolled₂)) ⋈ student

where Enrolled₁ represents course, Enrolled₂ represents prereq, and satisfy course.semester > Enrolled.semester.

$\therefore \pi_{\{s.sid,\ s.Name\}}(\sigma(s.sid = e.sid)(\rho_s(student) \times \rho_e(Enrolled)))$
$- \pi_{\{s.sid,\ s.Name\}}(\sigma(c.semester > e_1.semester)((course \bowtie (c.Cid = p.Cid)\ prereq) \bowtie (enrolled \bowtie (e_1.Cid = p.Cid)\ prereq)))(\rho_{e_1}(enrolled) \times \rho_c(course) \times \rho_p(prereq)))$

- Question 2:

① for query 1, since the block size is 256, and we need assume the block factor is 256 for all blocks, then

student: $\frac{10000}{256} = 39$

Course: $\frac{5000}{256} = 19.5$

Enrolled: $\frac{50000}{256} = 195$

Prereq: $\frac{10000}{256} = 39$

Also, we assume students are all uniformaity, such that

Select { ① second year classes filter $\frac{12}{3} = 4$
{ ② filter semester 1-3, 5000/4 = 1400

Join to student { ③ Join course & enrollment cid
{ ④ Join result & student sid

$\therefore$ cost(R) + cost(result)

·1. Schedules:

Since schedule S:

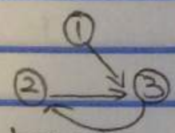
$r_1[x]\ r_1[y]\ w_1[x]\ r_2[y]\ w_3[y]\ w_1[x]\ r_2[y]$

so we can seperate that:

$x:\ r_1,\ w_1,\ w_1$

$y:\ r_1,\ r_2,\ w_3,\ r_2$

for x, it is serializable, for y, we can draw a table.



which contains a cycle between 2 and 3, this is conflict, so it is not serializable.

Therefore, this schedule is not serializable.

Then the way to solve conflict-serializable is to add commit between W_3 and r_2, "$W_3\ c\ r_2$". Also, use this way also can solve recorerable.

2. Locking schedule:

We need to show that locking schedule using the 2PL protocol are always serializable, the hint is that the serialization graph cannot contain a cycle if 2PL is used.

Since there is a theorem that the corresponding percedence graph of schedule S does not contain any cycles.

we can prove by:

($\Leftarrow$): Given the precedence graph G has a cycle, suppose the cycle consist of the transactions $T_1, T_2, \ldots T_k$. like $T_1 \to T_2 \to \ldots \to T_k$

Then the schedule S contains the non-swappable actions

$S:\ op_1(x) \ldots op_2(x) \ldots op_k(x) \ldots op_1(x)$

conflict conflict conflict

This swap will re-order actions.

·Also, there has 2 rule for 2PL:

i. If a transaction T wants to read an object, it first requests a shared lock on the object.

ii. All locks held by a transaction are released when transaction is completed.

Since 2PL strict, the above shedule is not possible. for example.
T_1 first acquires shared lock on x and Y. When T_2 runs, it lock on x.
The schedule leads to deadlock. When T_1 tries to write x, it also blocks
waiting for T_2 to released shared lock. Therefore, 2PL is used and do
not contain a cycle.