ComS 363 Fall 2022
Class Participation for Week 8

Learning objectives:

- Explore the impact of transaction support by MySQL.

Like several database management systems, MySQL supports transaction management to ensure correctness of concurrent executions of users' programs to a shared database. Developers can specify the level of support through setting of database configurations. By default, the system-defined autocommit variable is set to one, which means that DBMS saves the change to the database since the last commit permanently (aka commit) for every SQL statement. By default, the transaction isolation level for MySQL's default database engine InnoDB is REPEATABLE_READ. See https://dev.mysql.com/doc/refman/8.0/en/innodb-transaction-isolation-levels.html.

To check the values of these system-defined variables, do the following in MySQL Workbench. The two @ signs indicate that the variable is a system-defined variable.

select @@autocommit;

select @@transaction_isolation;

Instruction: Do all the questions.

1. Make sure the cost for the row of sid=1 and pid=102 is initially 100.

2. Open two sessions: one as coms363 and the other as root.

3. Find out what the code below does and run the following code as coms363 user. You need to give coms363 user all the privileges to access the suppliersparts database from the Exam 1 practice problems.

```
set autocommit=1; -- default
use suppliersparts;
select cost
from catalog
where sid=1 and pid=102;

-- add 5 to the cost and update the catalog table
update catalog
set cost = cost+5
where sid=1 and pid=102;
rollback; -- cancelation of the change

select cost
from catalog
where sid=1 and pid=102;
```

What is the value of the cost attribute for this row?

Answer:

4. As coms363, run the following code.

```
set autocommit=0; -- indicate that permanent update to DBMS since the last change is not done until commit is made.
use suppliersparts;
select cost
from catalog
where sid=1 and pid=102;

-- add 5 to the cost and update the catalog table
update catalog
set cost = cost+5
where sid=1 and pid=102;
rollback; -- cancelation of the change

select cost
from catalog
where sid=1 and pid=102;
```

What is the value of the attribute cost for the row where sid=1 and pid=102?
Answer:

5. As coms363, run the code below.

```
set autocommit=0; -- indicate that permanent update to DBMS since the last change is not done until commit is made.
use suppliersparts;
select cost
from catalog
where sid=1 and pid=102;

-- add 5 to the cost and update the catalog table
update catalog
set cost = cost+5
where sid=1 and pid=102;
commit; -- make the change permanent

select cost
from catalog
where sid=1 and pid=102;
```

What is the value of the attribute cost for the row where sid=1 and pid=102?
Answer:

6. Close the previous connections. Open two different connections in MySQL Workbench, one as coms363 and the other as root. We are going to explore the impact of different transaction isolation

levels. The READ UNCOMMITTED isolation level allows another program to see results that have not been guaranteed to be permanently saved in the database. Make sure you answer Question 5 first.

Connection 1 (coms363) adds 5	Connection 2 (root) deducts 5

use suppliersparts;
set autocommit=0;
set session transaction isolation level READ UNCOMMITTED;
 	
update catalog
set cost = cost + 5
where sid=1 and pid=102;	

	set autocommit = 0;
	-- this setting allows other transactions to read intermediate results
	set session transaction isolation level READ UNCOMMITTED;
	use suppliersparts;
	-- initialize the cost
	set @mycost = 0;
	
	select cost into @mycost
	from catalog
	where sid=1 and pid=102;
	
	select @mycost;

rollback;	

	set @mycost = @mycost-5;
	update catalog
	set cost = @mycost
	where sid=1 and pid=102;
	
	commit;

	select cost from catalog where sid=1 and pid=102;

What is the value of the cost attribute for the row with sid=1 and pid=102?
Answer:

If we start with the same cost value in question 5, after running statements in the left column in its entirety first before running the statements in the right column, do we get the same value as in the answer for question 6? **Answer:**