CO527: Advanced Database Systems
Lab 03 - Transaction Processing

Ranage R.D.P.R.
E/19/310

```
MySQL localhost:3306 ssl company ★ SQL > start transaction;
Query OK, 0 rows affected (0.0018 sec)
MySQL localhost:3306 ssl company ★ SQL > update salary set salary = 1.1*salary where emp_no=43624;
Query OK, 0 rows affected (0.0169 sec)

Rows matched: 0  Changed: 0  Warnings: 0
MySQL localhost:3306 ssl company ★ SQL > select * from salary where emp_no=43624;
Empty set (0.0018 sec)
MySQL localhost:3306 ssl company ★ SQL > rollback;
Query OK, 0 rows affected (0.0033 sec)
MySQL localhost:3306 ssl company SQL >
```

Concurrent Accesses

I. Issue a select query to view the current status of the departments table in both sessions.

Session 1

```
MySQL localhost:3306 ssl company SQL > select * from department
                                    -> ;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
+---------+--------------------+
9 rows in set (0.0022 sec)
```

Session 2

```
MySQL localhost:3306 ssl company SQL > select * from department
                                    -> ;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
+---------+--------------------+
9 rows in set (0.0219 sec)
```

II. Now, start transaction running start transaction in both sessions.
III. Insert a new row into the departments table from the 1st session and check if the changes are visible in the second session.

Session 1

```
MySQL localhost:3306 ssl company SQL > start transaction;
Query OK, 0 rows affected (0.0009 sec)
MySQL localhost:3306 ssl company ★ SQL > INSERT INTO department (dept_no,dept_name) VALUES ("d010","New D
epartment");
Query OK, 1 row affected (0.0258 sec)
MySQL localhost:3306 ssl company ★ SQL >
```

Session 2

```
MySQL localhost:3306 ssl company SQL > start transaction;
Query OK, 0 rows affected (0.0008 sec)
MySQL localhost:3306 ssl company ★ SQL > select * from department;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
+---------+--------------------+
9 rows in set (0.0092 sec)
```

IV. Commit changes in the 1st command window and check if you can see the updates done at 1st window in 2nd command window.

Session 1

```
MySQL localhost:3306 ssl company ★ SQL > commit;
Query OK, 0 rows affected (0.0147 sec)
MySQL localhost:3306 ssl company SQL > select * from department;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
| d010    | New Department     |
+---------+--------------------+
10 rows in set (0.0027 sec)
```

Session 2

```
MySQL localhost:3306 ssl company ★ SQL > select * from department;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
+---------+--------------------+
9 rows in set (0.0009 sec)
MySQL localhost:3306 ssl company ★ SQL >
```

V. Explain your observations before and after running the commit in the 1st window.

The output from session 02 remained unchanged.

The session 02 modifications were not displayed before committing the session 01 changes as well as after committing the changes.

This was a result of session 2 still being in process. (The transaction hasn't been completed yet.) The changes were visible following session 02's commit.

Concurrent Updates

I. Try to do a concurrent update to the same row in departments table during two transactions.

Session 1

```
MySQL localhost:3306 ssl company SQL > start transaction;
Query OK, 0 rows affected (0.0011 sec)
MySQL localhost:3306 ssl company ★ SQL > update department
                                      -> set dept_name = "Accounting"
                                      -> where dept_no = "d010";
Query OK, 1 row affected (0.0121 sec)

Rows matched: 1  Changed: 1  Warnings: 0
MySQL localhost:3306 ssl company ★ SQL >
```

Session 2

```
MySQL localhost:3306 ssl company SQL > start transaction;
Query OK, 0 rows affected (0.0007 sec)
MySQL localhost:3306 ssl company ★ SQL > update department
                                     -> set dept_name = "HR"
                                     -> where dept_no = "d010";
ERROR: 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
MySQL localhost:3306 ssl company ★ SQL >
```

II. Explain what happens before ending any of the transactions.

Because a UPDATE statement locks a field, session 01 forced session 02 to wait in line while locking the field that needed to be changed.

whenever a row in session 01 is updated. Once the commit is complete, this row is locked. Therefore, until session 01 commits, session 02 is not permitted to change the same row.

III. What happens when you commit your changes in the 1st session?

Session 1

```
MySQL localhost:3306 ssl company ★ SQL > commit;
Query OK, 0 rows affected (0.0125 sec)
MySQL localhost:3306 ssl company SQL > select * from department;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
| d010    | Accounting         |
+---------+--------------------+
10 rows in set (0.0011 sec)
```

Session 2

```
ERROR: 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
MySQL localhost:3306 ssl company ★ SQL > update department
                                      -> set dept_name = "HR"
                                      -> where dept_no = "d010";
Query OK, 1 row affected (0.0028 sec)

Rows matched: 1  Changed: 1  Warnings: 0
MySQL localhost:3306 ssl company ★ SQL > select * from department;
+---------+--------------------+
| dept_no | dept_name          |
+---------+--------------------+
| d001    | Marketing          |
| d002    | Finance            |
| d003    | Human Resources    |
| d004    | Production         |
| d005    | Development        |
| d006    | Quality Management |
| d007    | Sales              |
| d008    | Research           |
| d009    | Customer Service   |
| d010    | HR                 |
+---------+--------------------+
10 rows in set (0.0023 sec)
MySQL localhost:3306 ssl company ★ SQL >
```

After committing the session 1, it is possible to update the same row in the session 2.

What to Turn In

Use your imagination and words to write a scenario where using transactions is essential and then create the required tables and test how the transaction will effect your tables,

1. **during the transaction execution.**
2. **after rollback statement.**
3. **after the commit statement.**
4. **during 2 concurrent transactions, both of them update a record and both of them commit it.**

Simple database for a food store was created.

```
MySQL localhost:3306 ssl food_store SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  980.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1400.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0026 sec)
MySQL localhost:3306 ssl food_store SQL >
```

1. Transaction

First add 10% discount for all Chinese food and then rollback it

Session 1

```
MySQL localhost:3306 ssl food_store SQL > start transaction;
Query OK, 0 rows affected (0.0008 sec)
MySQL localhost:3306 ssl food_store ★ SQL > update menu
                                          -> set price = 0.90 * price
                                          -> where category = "Chinese";
Query OK, 2 rows affected (0.0021 sec)

Rows matched: 2  Changed: 2  Warnings: 0
MySQL localhost:3306 ssl food_store ★ SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  882.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1400.00 |
| I003    | Mix Fried Noodles       | Chinese   |  810.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0006 sec)
```

Session 2

```
MySQL localhost:3306 ssl food_store SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  980.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1400.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0023 sec)
```

Here updates cannot be seen

2. Rollback

Session 1(After rolling back)

```
MySQL localhost:3306 ssl food_store ★ SQL > rollback;
Query OK, 0 rows affected (0.0088 sec)
MySQL localhost:3306 ssl food_store SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  980.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1400.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0016 sec)
MySQL localhost:3306 ssl food_store SQL >
```

Session 2

```
MySQL localhost:3306 ssl food_store SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  980.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1400.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
```

3. Commit

Add a 5% discount to Italian foods and check it from the 2nd session without commit

Session 1

```
MySQL localhost:3306 ssl food_store SQL > start transaction;
Query OK, 0 rows affected (0.0016 sec)
MySQL localhost:3306 ssl food_store ★ SQL > update menu
                                          -> set price = 0.95 * price
                                          -> where category = "Italian";
Query OK, 1 row affected (0.0015 sec)

Rows matched: 1  Changed: 1  Warnings: 0
MySQL localhost:3306 ssl food_store ★ SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  980.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1330.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
```

Session 2

```
MySQL localhost:3306 ssl food_store SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  980.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1400.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0015 sec)
MySQL localhost:3306 ssl food_store SQL >
```

After committing,

Session 1

```
MySQL localhost:3306 ssl food_store ★ SQL > commit;
Query OK, 0 rows affected (0.0099 sec)
MySQL localhost:3306 ssl food_store SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  980.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1330.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0011 sec)
MySQL localhost:3306 ssl food_store SQL >
```

Session 2

```
MySQL localhost:3306 ssl food_store SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  980.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1330.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1650.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1500.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0012 sec)
MySQL localhost:3306 ssl food_store SQL >
```

4. Concurrent Updates

First in the session 1, the prices of all items are increased by 100. Then, without committing the session 1 transaction, session 2 tries to update the same rows.

Session 1

```
MySQL localhost:3306 ssl food_store SQL > start transaction;
Query OK, 0 rows affected (0.0010 sec)
MySQL localhost:3306 ssl food_store ★ SQL > update menu
                                          -> set price = price + 100;
Query OK, 5 rows affected (0.0085 sec)

Rows matched: 5  Changed: 5  Warnings: 0
MySQL localhost:3306 ssl food_store ★ SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   | 1080.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1430.00 |
| I003    | Mix Fried Noodles       | Chinese   | 1000.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1750.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1600.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0020 sec)
```

Session 2

```
5 rows in set (0.0012 sec)
MySQL localhost:3306 ssl food_store SQL > start transaction;
Query OK, 0 rows affected (0.0008 sec)
MySQL localhost:3306 ssl food_store ★ SQL > update menu
                                          -> set price = 0.90 * price
                                          -> where category = "Chinese";
ERROR: 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
MySQL localhost:3306 ssl food_store ★ SQL >
```

But session 2 cannot update the same row because session 1 has not been committed yet.

After session 1 is committed, session 2 can update.

Session 1

```
MySQL localhost:3306 ssl food_store ★ SQL > commit;
Query OK, 0 rows affected (0.0086 sec)
MySQL localhost:3306 ssl food_store SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   | 1080.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1430.00 |
| I003    | Mix Fried Noodles       | Chinese   | 1000.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1750.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1600.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0022 sec)
MySQL localhost:3306 ssl food_store SQL >
```

Session 2

```
ERROR: 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
MySQL localhost:3306 ssl food_store ★ SQL > update menu
                                          -> set price = 0.90 * price
                                          -> where category = "Chinese";
Query OK, 2 rows affected (0.0029 sec)

Rows matched: 2  Changed: 2  Warnings: 0
MySQL localhost:3306 ssl food_store ★ SQL > select * from menu;
+---------+-------------------------+-----------+---------+
| item_id | item_name               | category  | price   |
+---------+-------------------------+-----------+---------+
| I001    | Chicken Fried Rice      | Chinese   |  972.00 |
| I002    | Spaghetti Carbonara     | Italian   | 1430.00 |
| I003    | Mix Fried Noodles       | Chinese   |  900.00 |
| I004    | Nasi Gorenge Rice       | Thai      | 1750.00 |
| I005    | Seafood Mongolian Rice  | Mongolian | 1600.00 |
+---------+-------------------------+-----------+---------+
5 rows in set (0.0013 sec)
MySQL localhost:3306 ssl food_store ★ SQL >
```