# CSI 3130 Databases II (Fall 2016)

**Project (part D): Pass 1 (30%) -- Due November 13**

You are required to open **two concurrency sessions**, and run the following two sets of transactions at the same time, against your database. Note that these should be handled as separate transactions, i.e. each transaction should have its own BEGIN and END transactions statement. These refer to the transactions you wrote for part B of the project.

**Session 1** T1, T2, T4, T7, T10

**Session 2** T3, T5, T6, T8, T9

For this part of the project, use:

**Isolation Level** : Read Committed (Default)

**Lock mode** : Default

**What to Submit**

Collect the following relevant transaction statistics into your own **"transaction statistics log"** and then display it at the end of the two sessions.

a. the names of the actual disk pages used, as they are accessed by each transaction,

b. the actual locks, if any, that are being held, and the transaction-id of the transaction that holds it,

c. the level of explicit locking used, if any,

d. the status of a transaction as it is executed, including active, prepared and commit (or abort), together with the initial time it started, and

e. whether a deadlock was detected and which transaction was aborted(if any).

You are also required to display the end result of the sessions and then determine whether serializability has been ensured. That is, you need to determine whether the end results leave the database in a consistent state, i.e. no unrepeatable reads and phantoms have occurred. (Refer to Chapters 20 and 21 of B1 text book or Chapters 16 and 17 of B2 text book.)

Below are links to descriptions and instructions on setting the isolation levels and lock modes, as well as collecting statistics, FYI:

**Isolation Levels in Postgresql**

To see the standard isolation levels implemented in Postgresql, see

https://www.postgresql.org/docs/9.1/static/transaction-iso.html

**Lock Modes in Postgresql**

PostgreSQL provides various lock modes to control concurrent access to data in tables, see

https://www.postgresql.org/docs/9.1/static/explicit-locking.html

**Setting Transaction Isolation Levels**

Use the command SET TRANSACTION within the transaction that must have the desired isolation level, that is, after the BEGIN statement. For details, see

https://www.postgresql.org/docs/9.1/static/sql-set-transaction.html

**Viewing Collected Statistics**

Note : For more details on the statistics functions see

https://www.postgresql.org/docs/9.1/static/monitoring-stats.html

Several predefined views are available to show the results of statistics collection. Alternatively, one can build custom views using the underlying statistics functions. Note: a query or transaction still in progress does not affect the displayed totals. You can find information about system stats: Catalogs -> pg_catalog -> Views, Tables, Functions