University of Connecticut
Computer Science and Engineering

CSE 4701 - Principles of Databases
Fall 2019 – Project 2 - Help

1. **I forgot the password to my local MySQL Server. What do I do?**
   You can search online to find ways to reset password to your local MySQL server. Alternatively, if you do not have any important data in your local MySQL database, it would probably be easier to just uninstall everything and install again. You can set a new password this way.

2. **Do I have to comment my codes?**
   Comments are NOT required throughout your program. However, it is required in any database transaction related parts of the code. Few words or a sentence should be enough to describe a function or block of code. Please try to make it comprehensive, but not too lengthy.

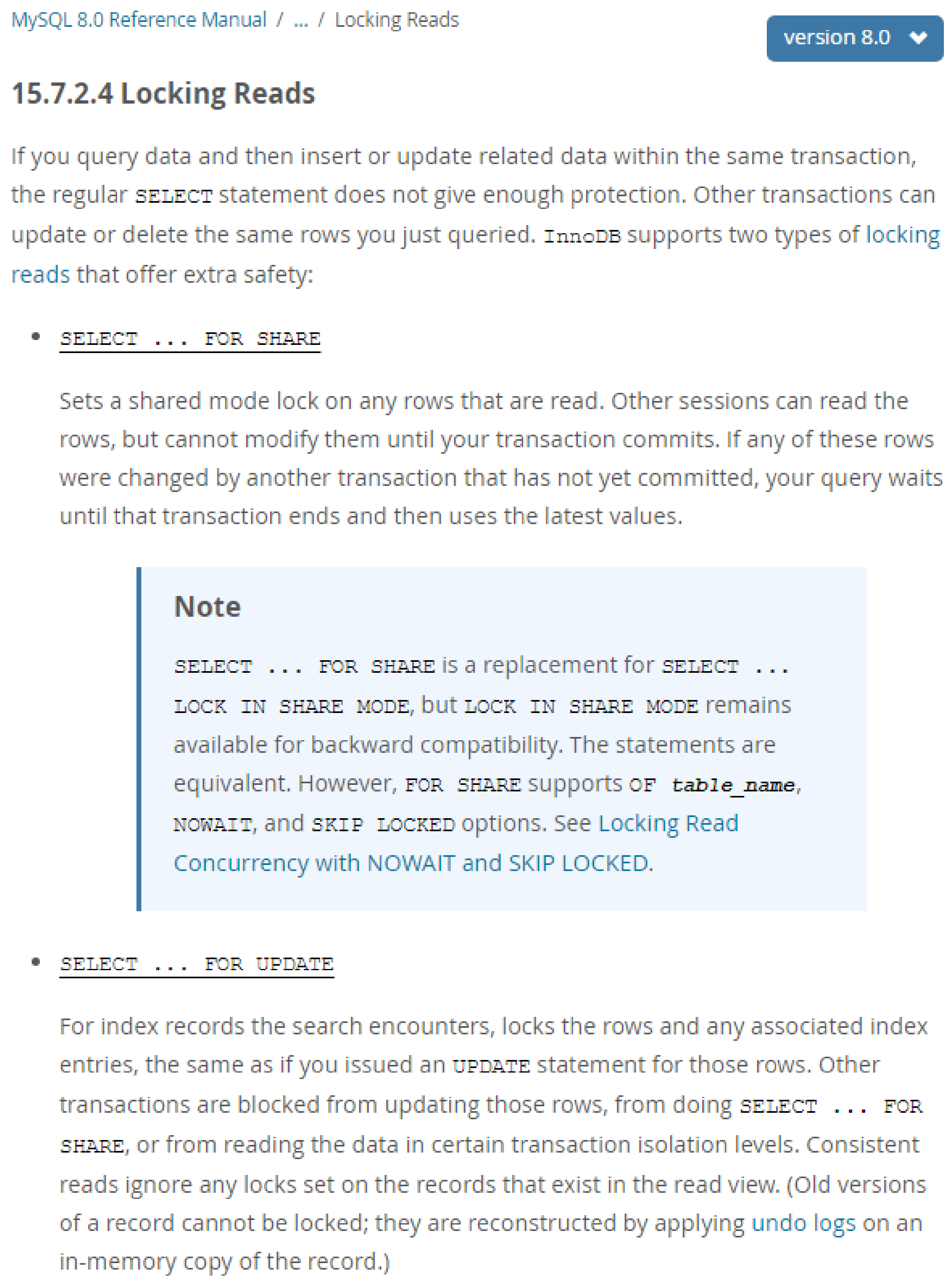
3. **Considering our code will run on TA's machine, what would you like us to set our username and password to?**
   The database credentials (hostname, username, password & database name) should be easily configurable in your code. This part should be commented properly. For our convenience, you can set the following credentials in the version you submit to HuskyCT.
   Hostname = "localhost"
   Database = "cse4701f19_project2"
   Username = "root"
   Password = ""

4. **Any guidance regarding how to invoke SQL from various host languages?**
   There are plenty of resources available online on how to use embedded SQL on various languages.
   - To use python refer to https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html
   - To use Java refer to https://www.javatpoint.com/example-to-connect-to-the-mysql-database
   - To use PHP refer to https://www.w3schools.com/php/php_mysql_connect.asp

5. **What do I do if I can't install MySQL library in python?**
   There are alternate libraries to connect to MySQL database. You can use any other library that you can find online. One example is PyMySQL available here. https://github.com/PyMySQL/PyMySQL

6. **Can I use the LOCK TABLES MySQL features to lock my records?**
   Please refer to Deliverables Section 4(iv) of Project description document. Information on MySQL locks are available at https://dev.mysql.com/doc/refman/8.0/en/innodb-locking-reads.html

   For example, see next page showing the beginning of MySQL 8.0 Reference Manual.



MySQL 8.0 Reference Manual / ... / Locking Reads

version 8.0

15.7.2.4 Locking Reads

If you query data and then insert or update related data within the same transaction, the regular `SELECT` statement does not give enough protection. Other transactions can update or delete the same rows you just queried. `InnoDB` supports two types of locking reads that offer extra safety:

- `SELECT ... FOR SHARE`

  Sets a shared mode lock on any rows that are read. Other sessions can read the rows, but cannot modify them until your transaction commits. If any of these rows were changed by another transaction that has not yet committed, your query waits until that transaction ends and then uses the latest values.

  > **Note**
  >
  > `SELECT ... FOR SHARE` is a replacement for `SELECT ... LOCK IN SHARE MODE`, but `LOCK IN SHARE MODE` remains available for backward compatibility. The statements are equivalent. However, `FOR SHARE` supports `OF` ***`table_name`***, `NOWAIT`, and `SKIP LOCKED` options. See Locking Read Concurrency with NOWAIT and SKIP LOCKED.

- `SELECT ... FOR UPDATE`

  For index records the search encounters, locks the rows and any associated index entries, the same as if you issued an `UPDATE` statement for those rows. Other transactions are blocked from updating those rows, from doing `SELECT ... FOR SHARE`, or from reading the data in certain transaction isolation levels. Consistent reads ignore any locks set on the records that exist in the read view. (Old versions of a record cannot be locked; they are reconstructed by applying undo logs on an in-memory copy of the record.)

Continue reading!