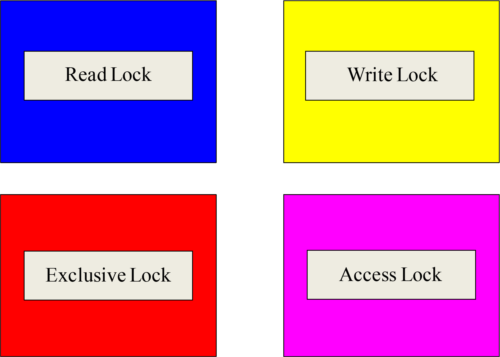
Chapter 10 – Locking

“If you are planning for a year, sow rice; if you are planning for a decade, plant trees; if you are planning for a lifetime, educate people."

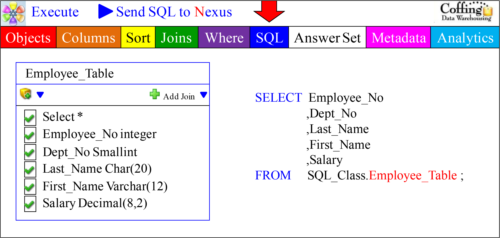
-Chinese Proverb

TheFour Major Locks of Teradata



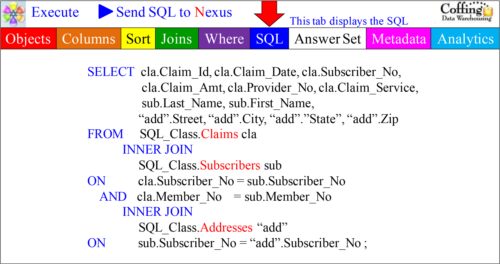
There are four locks used in the Teradata Database, and they are the Read, Write, Exclusive and Access Locks. This is an important concept that will soon have that bright light bulb in your brain lighting up.

TheRead Lock



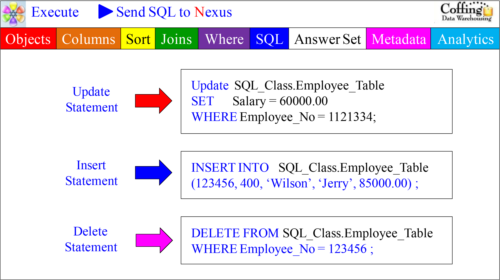
When a user runs an SQL Select statement, then Teradata automatically places a read lock on the table being selected from. Above, we can see that the user has dragged the Employee\_Table inside the Super Join Builder and Selected All Columns. The SQL can now be submitted. The user might not know or see that a Read lock has been placed, but the Teradata lock manager does this for the user in the background.

TheRead Lock and Joins



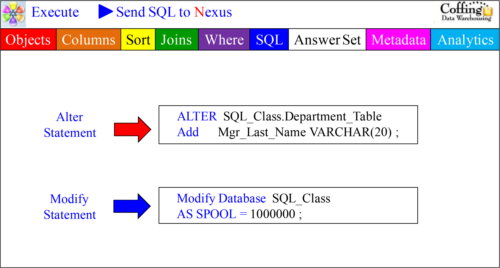
When a user runs SQL that involve a Join, then Teradata automatically places a Read Lock on all of the tables being joined. Above, we can see that the user has performed a 3-table join. All three tables automatically receive a Read Lock from the Lock Manager.

TheWrite Lock



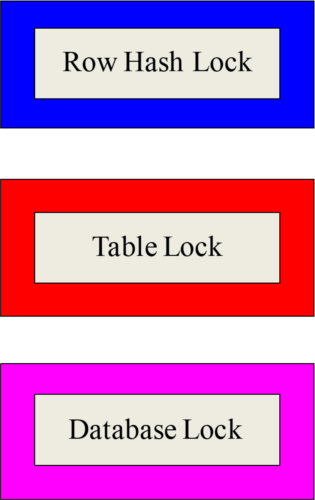
When a user utilizes SQL that does an Update or an Insert or a Delete, the Teradata Lock Manager places a Write Lock in the background on the table. The user merely submits the SQL, and the Teradata Lock Manager manages the locking.

TheExclusive Lock



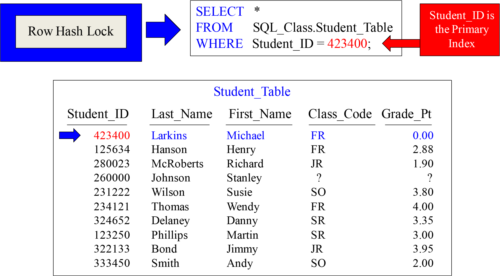
When a user utilizes SQL that does an Alter Table or a Modify Database SQL statement, then the Teradata Lock Manager automatically (behind the scenes) places an Exclusive Lock on the Table or the Entire Database. The user merely submits the SQL, and the Teradata Lock Manager manages the locking.

TheThree Levels of Locking



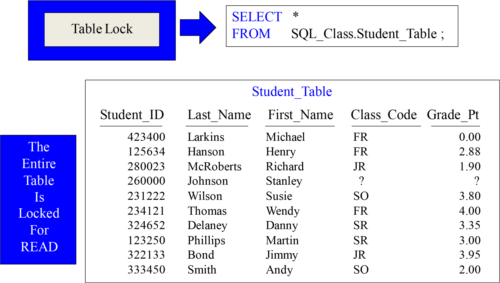
The Teradata Lock Manager is responsible for placing the lock type on a Table or Database and does so automatically based on the SQL the user has submitted. The Lock Manager will also decide whether or not to place the lock at the Row Hash level, the Table level, or the Database level. The next couple of slides will clear this up.

Lockingat the Row Hash Level



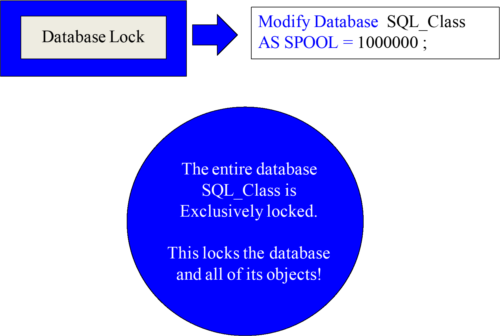
Why lock the entire table when you can lock a single row? If the User SQL uses the Primary Index of a table, then Teradata will NOT lock the entire table. It will only lock the row(s) it needs to. This is ideal in a Teradata environment. A Row Hash lock will almost always be placed on queries in which the WHERE clause uses a Unique index.

Lockingat the Table Level



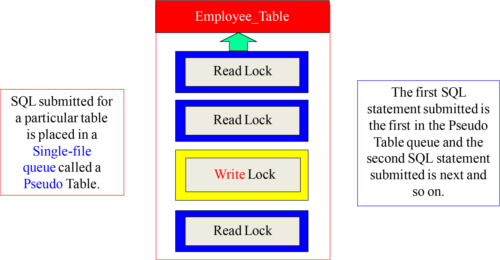
In this case, Teradata is selecting all columns and all rows from the table. There is no WHERE clause so the Teradata Lock Manager will automatically place a READ Lock on the entire table. If you don’t use a WHERE clause or you don’t use an Index in the WHERE clause, you can depend that Teradata will automatically lock at the Table level.

Lockingat the Database Level



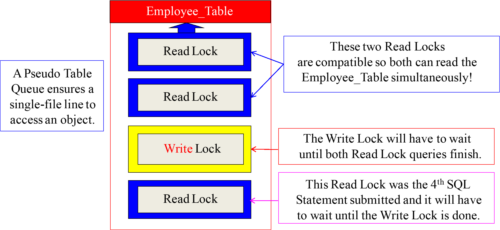
Although this is the rarest of lock levels, the Teradata Lock Manager will lock an entire database. This will be the case when a database is modified. Think of this as driving down the road and a bridge is under construction. No cars get through until the work on the bridge is completed. The lock is released above once the database is modified.

TheOngoing Battle between Read and Write Locks



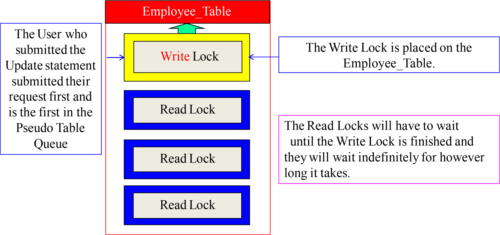
Read and Write Locks have been battling each other for years. A Read Lock is placed so data can be read, and a Write Lock is placed when data is being Inserted, Updated, or Deleted. Teradata locking works on a first come first serve basis. The user who submits their SQL first is the first user in the queue and the first to gain access to the table. This locking table is called a “Pseudo Table”. Remember that the queue is called a Pseudo table because this will be an important concept when reading an Explain plan.

Compatibilitybetween Read Locks



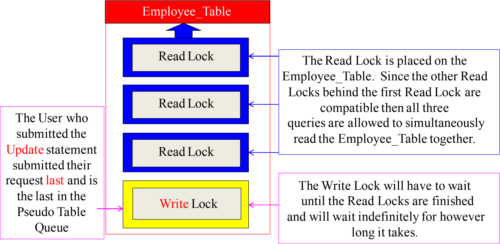
Read Locks are compatible, but Write Locks are not. The Pseudo Table Queue works on a first come first serve basis. The first two queries were obviously Select queries, and since Read Locks are compatible then both queries can read the Employee\_Table simultaneously. The Write Lock was the third query submitted and it will have to wait until both Read Lock queries finish. The final Read Lock will have to wait on the Write to finish. This is the idea behind the queue and that is to control the locking.

WhyRead Locks Wait on Write Locks



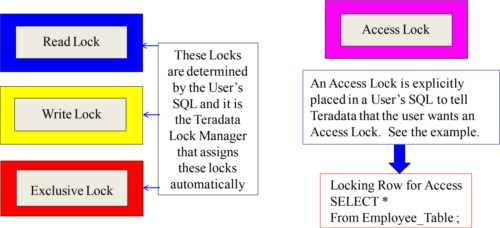
Write Locks don’t just read a table, but they are used because the data is changing because of either an Insert, Update, or Delete statement. The Teradata Lock Manager locks the row being changed or the entire table depending on the SQL. Read Locks behind a Write Lock in the Pseudo Table Queue will wait until the changes are complete and the Write Lock is released. This is done for data integrity. The Read Locks can now be assured that they are dealing with the correct data.

WhyWrite Locks Wait on Read Locks



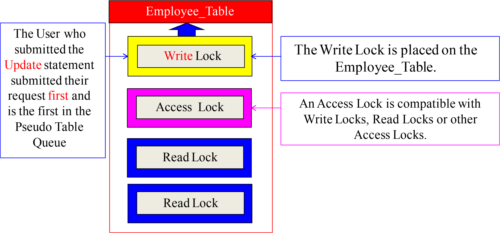
How would you like to be reading a newspaper and have someone take it from you so they can work the crossword puzzle? That is bad form. That is like a user running a Select query and then another who wants to update the table interrupting. Once a user has a Read lock on a table, they are assured that any queries needing a Write Lock to perform an Insert, Update, or Delete will have to wait until the Read Lock has finished.

TheAccess Lock is Different from the Other Locks



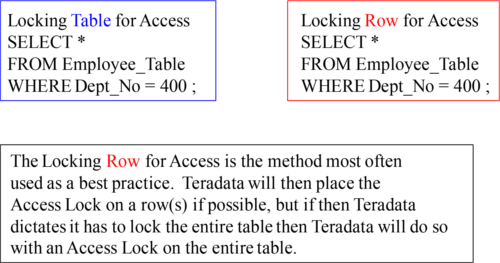
The Access Lock is different than the other Teradata Locks. This is because the Access Lock is controlled by the User. The other Locks are controlled by the Teradata lock Manager. The User must request an Access Lock. Notice the example (red box) and the Locking Row for Access statement. Turn the page to find out more!

Whatis the Purpose of an Access Lock?



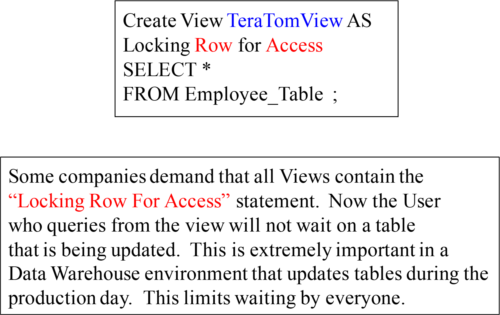
What is the purpose of an Access Lock? So the user who is selecting from a table will NOT wait on a Write Lock. As the table is being updated, the User with an Access Lock can still read the table. An Access Lock is often referred to as a “Dirty Read” or a “Read without Integrity”. This is because an Access Lock reads the current data as it is being changed so the data may not be perfectly up to date. When reading millions or billions of records, who cares? That’s why most views have an Access Lock inside.

LockingModifiers - Locking Row, Table or Database



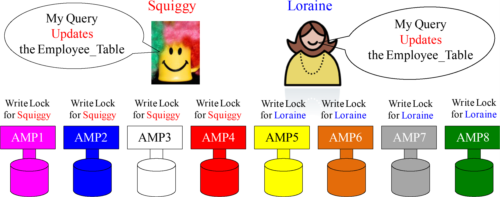
What you see above and what you will read below is called a “Locking Modifier”. Whether you are Selecting from a single table or performing a join, just one “Locking Row for Access” statement is needed in front of your SQL. If Teradata can lock only a single row or some rows, it will try to lock them with an Access Lock. But if Teradata determines it must lock the entire table(s), it will do so still with an Access Lock at the table level. Just use the “Locking Row for Access” when you want an Access Lock.

AllViews should consider the Locking for Access Statement



In an environment where updates are occurring simultaneously as users are also querying the tables, it is important to have the “Locking Row for Access” statement inside the view. Now users won’t have to wait on tables being updated. The data isn’t guaranteed to be the latest and greatest, but that is the only downside. Unless you need the data to be in a perfectly updated state, then you should use the Access Lock!

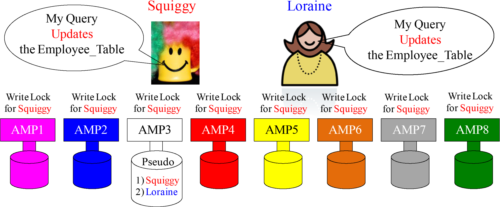
Whatis a Dead Lock or a Deadly Embrace?



Half the AMPs have locked the table for Squiggy and the other half for Loraine.

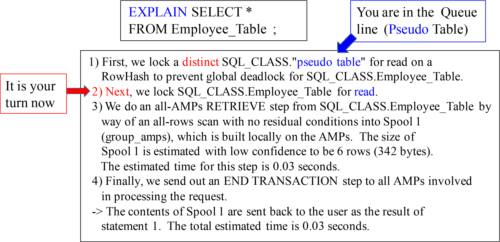
When some of the AMPs lock the Employee\_Table for **Squiggy** and the other half of the AMPs lock the Employee\_Table for **Loraine**, neither query can complete because they are both waiting on the other to release their locks. This is a “Dead Lock”, which is sometimes called a “Deadly Embrace”. If Teradata finds a deadlock, it will abort one of the queries after 240 seconds of waiting (default setting in DBS Control record).

PseudoTables are designed to minimize Dead Locks



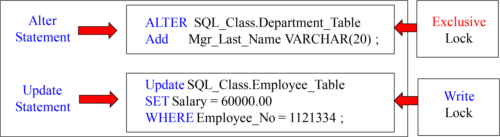
A Pseudo Table is a single-file line (queue) for all-AMP queries designed to prevent dead locks. Since Teradata is a parallel processing database with a potential of thousands of AMPs, Teradata assigns a single AMP to be the gate keeper for a particular table. Squiggy and Loraine are both wanting to Update the Employee\_Table, so Teradata hashes the name Employee\_Table and then takes the row hash and uses it in conjunction with the hash map to assign the AMP chosen to be responsible for the Pseudo Table. In our example, Squiggy was the first in line in the Pseudo Table so now the chosen AMP alerts the other AMPs to place a Write Lock on the table for Squiggy.

PseudoTables are referenced in the Explain Plan



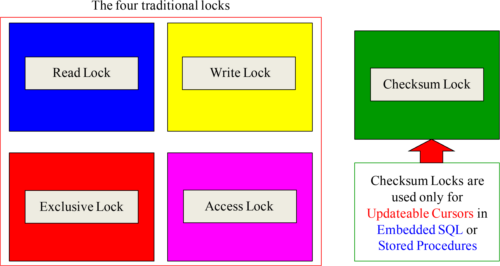
One of the frustrating things about an Explain plan is that most people are confused by the first two lines, so they give up. Now you know that these first lines of an Explain plan refer to the Pseudo Table (single-file line), also called the queue. You are merely waiting your turn. Once it is your turn in line (queue), then Teradata will actually lock the table for you. The word distinct refers to the fact that a single AMP has been chosen for this table to act as the gatekeeper, thus creating the ability for a single-file line.

IncompatibleLocks Wait on each Other



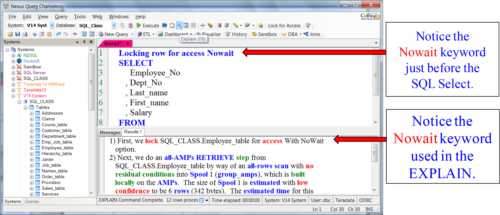
When a user utilizes SQL that does an Alter Table or a Modify Database SQL statement, then the Teradata Lock Manager automatically (behind the scenes) places an Exclusive Lock on the Table or the Entire Database. The user merely submits the SQL and the Teradata Lock Manager manages the locking. If locks are incompatible, then the user waits until the previous lock is released. This will happen when the previous query is finished or aborted. How long will they wait? It could be forever! They wait until the previous query is finished and the lock is released.

TheChecksum Lock of Teradata



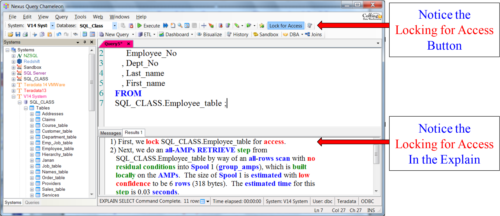
You have probably heard of a Checksum, but not a Checksum Lock. This only happens when Teradata SQL uses Updateable Cursors and this is only done in Embedded SQL or Stored Procedures.

TheNowait Option for Locking



The Nowait option will abort the query if it has to wait. This is often used in Automatic Teller Machines (ATMs) or another application where they don’t want the user to wait. The query either executes immediately or aborts.

TheAutomatic Locking for Access Button inside Nexus



Turn on the Lock for Access button in N exus and all queries will attempt a lock for Access (if possible).

The Nexus has a Lock for Access button, and if the user turns it on each SELECT query will automatically get an Access Lock submitted for the user.