

Any query against a table will actually see a map that of the data as it existed at the time the query began.

In a versioning system, each row has two additional, hidden values associated with it. These values represent when the row was created and when it was expired (or deleted). Rather than storing the actual time at which these events occur, the database stores the version number at the time each event occurred. The database version (or system version) is a number that increments each time a query\* begins. We'll call these two values the creation id and the deletion id.

Under MVCC, a final duty of the database server is to keep track of all the running queries (with their associated version numbers). Let's see how this applies to particular operations:

#### SELECT

When records are selected from a table, the server must examine each row to ensure that it meets several criteria:

- Its creation id must be less than or equal to the system version number. This ensures that the row was created before the current query began.
- Its deletion id, if not null, must be greater than the current system version. This ensures that the row wasn't deleted before the current query began.
- Its creation id can't be in the list of running queries. This ensures that the row wasn't added or changed by a query that is still running.

Rows that pass all of these tests may be returned as the result of the query.

#### INSERT

When a row is added to a table, the database server records the current version number along with the new row, using it as the row's creation id.

#### DELETE

To delete a row, the database server records the current version number as the row's deletion id.

#### UPDATE

When a row is modified, the database server writes a new copy of the row, using the version number as the new row's creation id. It also writes the version number as the old row's deletion id.

The result of all this extra record keeping is that read queries never lock tables, pages, or rows. They simply read data as fast as they can, making sure to select only rows that meet the criteria laid out earlier. The drawbacks are that the server has to store a bit more data with each row and do a bit more work when examining rows. Table 2-1 summarizes the various locking models and concurrency in MySQL.

\* That's not quite true. As you'll see when we start talking about transactions later, the version number is incremented for each transaction rather than each query.