GROUP 7
DALISAY, Andres
PINAWIN, Timothy
SALVADOR, Antoinne Bryce
SY, Matthew

Concurrency Control and Consistency

Case	Iteration	Isolation Level	Steps	Expected	Actual	Pass or Fail
Case 1 Clicking the edit button on the same data item	1	read uncommitted	1. Both users from Node 2 and Node 3 will click on the edit button of the same movie	Both users should see the same movie data	Both users see the same movie data	**Pass**
	1	read committed		Both users should see the same movie data	Both users see the same movie data	**Pass**
	1	read repeatable		Both users should see the same movie data	Both users see the same movie data	**Pass**
	1	serializable		Both users should see the same movie data	Both users see the same movie data	**Pass**
	2	read uncommitted	1. Both users from Node 2 and Node 3 will click on the edit button of the same movie	Both users should see the same movie data	Both users see the same movie data	**Pass**
	2	read committed		Both users should see the same movie data	Both users see the same movie data	**Pass**
	2	read repeatable		Both users should see the same movie data	Both users see the same movie data	**Pass**
	2	serializable		Both users should see the same movie data	Both users see the same movie data	**Pass**
	3	read uncommitted	1. Both users from Node 2 and Node 3 will click on the edit button of the same movie	Both users should see the same movie data	Both users see the same movie data	**Pass**
	3	read committed		Both users should see the same movie data	Both users see the same movie data	**Pass**

	3	read repeatable		Both users should see the same movie data	Both users see the same movie data	**Pass**
	3	serializable		Both users should see the same movie data	Both users see the same movie data	**Pass**
Case 2 Editing 1 movie while another user is viewing the movie	1	read uncommitted	1. A user on Node 1 will edit the Director field of a certain movie 2. A User on Node 3 will refresh the main page where the edited row is seen.	Node 3 sees the edited Director field while the transaction from Node 1 is **still not** committed.	Node 3 sees the edited Director field while the transaction from Node 1 is **still not** committed.	**Pass**
	1	read committed		Node 3 sees the edited Director field when the transaction from Node 1 is finished.	Node 3 sees the edited Director field when the transaction from Node 1 is finished.	**Pass**
	1	read repeatable		Node 3 sees the edited Director field when the transaction from Node 1 is finished.	Node 3 sees the edited Director field when the transaction from Node 1 is finished.	**Pass**
	1	serializable		Node 3 sees the edited Director field when the transaction from Node 1 is finished.	Node 3 sees the edited Director field when the transaction from Node 1 is finished.	**Pass**
	2	read uncommitted	1. A user on Node 1 will edit the Title field of a certain movie 2. A User on Node 3 will refresh the main page where the edited row is seen.	Node 3 sees the edited Director field while the transaction from Node 1 is **still not** committed.	Node 3 sees the edited Director field while the transaction from Node 1 is **still not** committed.	**Pass**
	2	read committed		Node 3 sees the	Node 3 sees the	**Pass**

				edited Title field when the transaction from Node 1 is finished.	edited Title field when the transaction from Node 1 is finished.	
	2	read repeatable		Node 3 sees the edited Title field when the transaction from Node 1 is finished.	Node 3 sees the edited Title field when the transaction from Node 1 is finished.	**Pass**
	2	serializable		Node 3 sees the edited Title field when the transaction from Node 1 is finished.	Node 3 sees the edited Title field when the transaction from Node 1 is finished.	**Pass**
	3	read uncommitted	1. A user on Node 1 will edit the Actor field of a certain movie 2. A User on Node 3 will refresh the main page where the edited row is seen.	Node 3 sees the edited Actor field while the transaction from Node 1 is **still not** committed.	Node 3 sees the edited Actor field while the transaction from Node 1 is **still not** committed.	**Pass**
	3	read committed		Node 3 sees the edited Actor field when the transaction from Node 1 is finished.	Node 3 sees the edited Actor field when the transaction from Node 1 is finished.	**Pass**
	3	read repeatable		Node 3 sees the edited Actor field when the transaction from Node 1 is finished.	Node 3 sees the edited Actor field when the transaction from Node 1 is finished.	**Pass**
	3	serializable		Node 3 sees the edited Actor field	Node 3 sees the edited Actor field	**Pass**

				when the transaction from Node 1 is finished.	when the transaction from Node 1 is finished.	
Case 3 Editing 1 movie while another user edits the same movie	1	read uncommitted	1. A user in Node 2 will edit the director field of a certain movie 2. A user in Node 3 will edit the director field of the same movie from user in Node 3. User in Node 3 will submit first then user in Node 2 submits directly after.	Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	1	read committed		Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	1	read repeatable		Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	1	serializable		Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	2	read uncommitted	1. A user in Node 2 will edit the director field of a certain movie 2. A user in Node 3 will edit the director field of the same movie from user in Node 3. User in Node 3 will submit first then user in Node 2 submits directly after.	Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	2	read committed		Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	2	read repeatable		Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	2	serializable		Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	3	read uncommitted	1. A user in Node 2 will edit the director field of a certain movie 2. A user in Node 3 will	Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**

	3	read committed	edit the director field of the same movie from user in Node 3. User in Node 3 will submit first then user in Node 2 submits directly after.	Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	3	read repeatable		Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**
	3	serializable		Node 1 sees the edited Director field of the user in Node 2.	User in Node 1 sees the edited data from User in Node 2	**Pass**

Global Failure and Recovery

Case	Iteration	Steps	Expected	Actual	Pass or Fail
Case 1 Kill node 1, edit/add/delete a movie in node 2, and turn on node 1	1	1. Turn off Node 1 2. Add a movie record in Node 2 3. Turn on Node 1	Movie record is added to both Node 1 and Node 2	Movie record is added to both Node 1 and Node 2	**PASS**
	2	1. Turn off Node 1 2. Edit a movie record in Node 2 3. Turn on Node 1	Edited value of movie record is consistent in both Node 1 and Node 2	Edited value of movie record is consistent in both Node 1 and Node 2	**PASS**
	3	1. Turn off Node 1 2. Delete a movie record in Node 2 3. Turn on Node 1	Movie record is deleted in both Node 1 and Node 2	Movie record is deleted in both Node 1 and Node 2	**PASS**
Case 2 Kill node 2, edit/add/delete a movie in node 1, and turn on node 2	1	1. Turn off Node 2 2. Add a movie record in Node 1 3. Turn on Node 2	Movie record is added to both Node 1 and Node 2	Movie record is added to both Node 1 and Node 2	**PASS**
	2	1. Turn off Node 2 2. Edit a movie record in Node 1 3. Turn on Node 2	Edited value of movie record is consistent in both Node 1 and Node 2	Edited value of movie record is consistent in both Node 1 and Node 2	**PASS**
	3	1. Turn off Node 2 2. Delete a movie record in Node 1 3. Turn on Node 2	Movie record is deleted in both Node 1 and Node 2	Movie record is deleted in both Node 1 and Node 2	**PASS**
Case 3 Kill node 1, edit/add/delete a movie in node 2, sync	1	1. Turn off Node 1 2. Add a movie record in Node 2 3. Let Node 2 run	Node 2 adds the movie record and will repeatedly try to sync to Node 1.	Node 2 adds the movie record and will repeatedly try to sync to Node 1.	**PASS**
	2	1. Turn off Node 1 2. Edit a movie record in Node 2 3. Let Node 2 run	Node 2 updates the record and will repeatedly try to sync to Node 1.	Node 2 updates the record and will repeatedly try to sync to Node 1.	**PASS**

	3	1. Turn off Node 1 2. Delete a movie record in Node 2 3. Let Node 2 run	Node 2 deletes the movie and will repeatedly try to sync to Node 1.	Node 2 deletes the movie and will repeatedly try to sync to Node 1.	**PASS**
Case 4 Kill node 2, edit/add/delete a movie in node 1, sync	1	1. Turn off Node 2 2. Add a movie record in Node 1 3. Let Node 1 run	Node 1 adds the movie record and will repeatedly try to sync to Node 2.	Node 1 adds the movie record and will repeatedly try to sync to Node 2.	**PASS**
	2	1. Turn off Node 2 2. Edit a movie record in Node 1 3. Let Node 1 run	Node 1 updates the record and will repeatedly try to sync to Node 2.	Node 1 updates the record and will repeatedly try to sync to Node 2.	**PASS**
	3	1. Turn off Node 2 2. Delete a movie record in Node 1 3. Let Node 1 run	Node 1 deletes the movie and will repeatedly try to sync to Node 2.	Node 1 deletes the movie and will repeatedly try to sync to Node 2.	**PASS**