|  |  |
| --- | --- |
| **Name** |  |
| **UID no.** |  |
| **Experiment No.** |  |

|  |  |
| --- | --- |
| **AIM:** |  |
| **Program 1** | |
| **PROBLEM STATEMENT :** | . |
| **Theory :** |  |
| **Output** | Drop:    Create:    Insert:      Initial insert into database  Begin Transaction:      Started a new transaction and insert a new student into the college table:  Commit:    Made the changes become visible to other sessions (or users), committed the transaction by using the COMMIT TRANSACTION statement  Alternative: COMMIT WORK, COMMIT;  After executing the COMMIT statement, PostgreSQL also guarantees that the change will be durable if a crash happens.  ------x--------------------  Update:  Currently score at id = 2 is 75.      Changes is made to id =2 [Varun] and score is updated to 47.5  After committing changes are made visible.  Rollback:  Updated but not committed:    Score at id = 2 is 47.5.    After executing rollback score at id = 2 is reverted to 75  But if transaction is committed then we can’t revert back.    Serializability:  I’m gonna start 2 new transactions, then set their isolation level to serializable.    Get sum of all the scores of students |
| **Conclusion** | |