

(Practice) Quiz: Transactions and Locks

Total Score: /15

Printed Name:

Instructions:

1. Each problem is worth 3 points.
2. For each problem:
 - (a) Circle all SQL commands that block.
 - (b) Write an X through all commands that error, even if the command previously blocked.
 - (c) Write the output of each `SELECT` statement. (You do not need to write column names, and the order of rows does not matter.)
 - (d) If the sequence of commands results in a deadlock, say so.

Rules:

1. You MAY:
 - (a) Use any handwritten/printed notes
 - (b) Use your computer
 - (c) Reference arbitrary written material on the internet, including the Postgres documentation
2. You MAY NOT:
 - (a) Communicate with another human being
 - (b) Connect to postgres and run commands

Problem 1:

Session 1	Session 2
1 CREATE TABLE t (a INT UNIQUE);	1
2 CREATE TABLE u (2
3 b INT REFERENCES t(a)	3
4 DEFERRABLE INITIALLY DEFERRED	4
5);	5
6 INSERT INTO t VALUES (1);	6
7 INSERT INTO t VALUES (2);	7
8 INSERT INTO t VALUES (3);	8
9 INSERT INTO u VALUES (1);	9
10 INSERT INTO u VALUES (2);	10
11 BEGIN;	11
12	12 BEGIN;
13 DELETE FROM u WHERE b=1;	13
14 DELETE FROM u WHERE b=3;	14
15	15 INSERT INTO u VALUES (4);
16 INSERT INTO t VALUES (4);	16
17	17 INSERT INTO u VALUES (1);
18 COMMIT;	18
19 SELECT count(*) FROM u;	19
	20 COMMIT;

Problem 2:

Session 1

```
1
2
3 BEGIN;
4 LOCK TABLE t IN EXCLUSIVE MODE;
5
6
7
8 CREATE INDEX ON u(a);
9
10 COMMIT;
```

Session 2

```
1 CREATE TABLE t ( a INT );
2 CREATE TABLE u ( a INT UNIQUE );
3
4
5 BEGIN;
6 LOCK TABLE u IN EXCLUSIVE MODE;
7 INSERT INTO t VALUES (5);
8
9 COMMIT;
```

Problem 3:

Session 1

```
1 CREATE TABLE t ( a INT );
2 INSERT INTO t VALUES (1);
3 INSERT INTO t VALUES (2);
4 INSERT INTO t VALUES (3);
5
6 BEGIN;
7 UPDATE t SET a=6 WHERE a=1;
8
9 INSERT INTO t VALUES (0);
10 COMMIT;
11
12
13 SELECT count(*) FROM t;
```

Session 2

```
1
2
3
4
5 BEGIN;
6
7
8 UPDATE t SET a=7 WHERE a=1;
9
10
11 DELETE FROM t WHERE a=6;
12 COMMIT;
```

Problem 4:

Session 1	Session 2
1 CREATE TABLE t (2 a INT UNIQUE 3 DEFERRABLE INITIALLY DEFERRED 4); 5 INSERT INTO t VALUES (1); 6 INSERT INTO t VALUES (2); 7 INSERT INTO t VALUES (3); 8 9 BEGIN; 10 LOCK TABLE t IN ROW SHARE MODE; 11 UPDATE t SET a=6 WHERE a=1; 12 UPDATE t SET a=6 WHERE a=2; 13 14 15 COMMIT; 16 SELECT * FROM t;	1 2 3 4 5 6 7 8 BEGIN; 9 10 11 12 13 UPDATE t SET a=7 WHERE a=2; 14 LOCK TABLE t IN SHARE MODE; 15 16 17 COMMIT;

Problem 5:

Session 1

```
1 CREATE TABLE t ( a INT UNIQUE );
2 INSERT INTO t VALUES (1);
3 INSERT INTO t VALUES (2);
4 INSERT INTO t VALUES (3);
5
6
7
8 BEGIN;
9 INSERT INTO t VALUES (5);
10 INSERT INTO t VALUES (6);
11
12 INSERT INTO t VALUES (2);
13 COMMIT;
```

Session 2

```
1
2
3
4
5 BEGIN ISOLATION LEVEL
6 REPEATABLE READ;
7 SELECT count(*) FROM t;
8
9
10
11 SELECT count(*) FROM t;
12
13
14 SELECT count(*) FROM t;
15 COMMIT;
```