Quiz Practice Problems (Transactions and Locks)

Instructions: For each problem, you must perform all of the following tasks correctly in order to get credit.

1. Circle all SQL commands that block.
2. Write an X through all commands that error.
3. If a command causes a deadlock, write `DEADLOCK` and stop the problem.
4. Write the output of each `SELECT` statement.

You will be allowed to use a computer and access the internet, but you will not be allowed to connect to postgres and run postgres commands.

1 Basic Transactions

Problem 1:

Session 1

```
1  CREATE TABLE t ( a INT );
2
3  INSERT INTO t VALUES (5);
4  INSERT INTO t VALUES (6);
5
6  INSERT INTO t VALUES (2);
```

Session 2

```
1
2  SELECT count(*) FROM t;
3
4
5  SELECT count(*) FROM t;
6
7  SELECT count(*) FROM t;
```

Problem 2:

Session 1

```
1  CREATE TABLE t ( a INT );
2
3  BEGIN;
4  INSERT INTO t VALUES (5);
5  INSERT INTO t VALUES (6);
6
7  INSERT INTO t VALUES (2);
8  COMMIT;
```

Session 2

```
1
2  SELECT count(*) FROM t;
3
4
5
6  SELECT count(*) FROM t;
7
8
9  SELECT count(*) FROM t;
```

Problem 3:

Session 1

```
1  CREATE TABLE t ( a INT );
2
3  BEGIN;
4  INSERT INTO t VALUES (5);
5  INSERT INTO t VALUES (6);
6
7  INSERT INTO t VALUES (2);
8  ROLLBACK;
```

Session 2

```
1
2  SELECT count(*) FROM t;
3
4
5
6  SELECT count(*) FROM t;
7
8
9  SELECT count(*) FROM t;
```

Problem 4:

Session 1

```
1  CREATE TABLE t ( a INT );
2
3  BEGIN;
4  INSERT INTO t VALUES (5);
5  INSERT INTO t VALUES (6);
6
7  INSERT INTO t VALUES (2);
8  ABORT;
```

Session 2

```
1
2  SELECT count(*) FROM t;
3
4
5
6  SELECT count(*) FROM t;
7
8
9  SELECT count(*) FROM t;
```

2 Isolation Levels

Problem 5:

Session 1

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

Session 2

```
1
2  BEGIN;
3  SELECT count(*) FROM t;
4
5
6
7  SELECT count(*) FROM t;
8
9
10 SELECT count(*) FROM t;
11 COMMIT;
12 SELECT count(*) FROM t;
```

Problem 6:

Session 1

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

Session 2

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

Problem 7:

Session 1

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

Session 2

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

Note 1. From here on out, you should understand all of the examples as if any transaction was in any isolation level.

Problem 8:

Session 1

```
1  CREATE TABLE t ( a INT );
2
3
4  BEGIN;
5  INSERT INTO t VALUES (5);
6  INSERT INTO t VALUES (6);
7  INSERT INTO t VALUES (2);
8  COMMIT;
```

Session 2

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

Problem 9:

Session 1

```
1  CREATE TABLE t ( a INT );
2
3
4
5  BEGIN ISOLATION LEVEL
6  REPEATABLE READ;
7  INSERT INTO t VALUES (5);
8  SELECT count(*) FROM t;
9  INSERT INTO t VALUES (6);
10 SELECT count(*) FROM t;
11 INSERT INTO t VALUES (2);
12 SELECT count(*) FROM t;
13 COMMIT;
```

Session 2

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

3 Explicit Locks

Problem 10:

Session 1

```
1  CREATE TABLE t ( a INT );
2  BEGIN;
3  LOCK TABLE t IN EXCLUSIVE MODE;
4
5
6  COMMIT;
```

Session 2

```
1
2
3
4  BEGIN;
5  LOCK TABLE t IN SHARE MODE;
6
7  COMMIT;
```

Problem 11:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN SHARE MODE;	3
4	4 BEGIN;
5	5 LOCK TABLE t IN EXCLUSIVE MODE;
6 COMMIT;	6
	7 COMMIT;

Problem 12:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN SHARE MODE;	3
4	4 BEGIN;
5	5 LOCK TABLE t IN ROW SHARE MODE;
6 COMMIT;	6
	7 COMMIT;

Problem 13:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN EXCLUSIVE MODE;	3
4 COMMIT;	4
	5 BEGIN;
	6 LOCK TABLE t IN EXCLUSIVE MODE;
	7 COMMIT;

Problem 14:

Session 1

```
1  CREATE TABLE t ( a INT );
2  BEGIN;
3  LOCK TABLE t IN ROW EXCLUSIVE MODE;
4
5
6
7  COMMIT;
```

Session 2

```
1
2
3
4  BEGIN;
5  LOCK TABLE t IN ROW SHARE MODE;
6  COMMIT;
```

4 Deadlocks

Problem 15:

Session 1

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

Session 2

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

Problem 16:

Session 1

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

Session 2

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

Problem 17:

Session 1

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

Session 2

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

Problem 18:

Session 1

```
1  CREATE TABLE t ( a INT );
2
3  BEGIN;
4  LOCK TABLE t IN SHARE MODE;
5  LOCK TABLE u IN ROW EXCLUSIVE MODE;
6  COMMIT;
```

Session 2

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

Problem 19:

Session 1

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

Session 2

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

Problem 20:

Session 1

```
1  CREATE TABLE t ( a INT );
2
3  BEGIN;
4
5  LOCK TABLE t IN ACCESS SHARE MODE;
6
7
8  LOCK TABLE u IN ROW EXCLUSIVE MODE;
9  COMMIT;
```

Session 2

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

5 Implicit Locks I

Problem 21:

Session 1

```
1  CREATE TABLE t ( a INT );
2  BEGIN;
3  LOCK TABLE t IN ACCESS EXCLUSIVE MODE;
4
5  COMMIT;
```

Session 2

```
1
2
3
4  INSERT INTO t VALUES (5);
```

Problem 22:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN ACCESS EXCLUSIVE MODE;	3
4	4 SELECT count(*) FROM t;
5 COMMIT;	

Problem 23:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN SHARE MODE;	3
4	4 INSERT INTO t VALUES (1);
5 COMMIT;	

Problem 24:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN SHARE MODE;	3
4	4 SELECT count(*) FROM t;
5 COMMIT;	

Problem 25:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN ROW SHARE MODE;	3
4	4 INSERT INTO t VALUES (1);
5 COMMIT;	

6 Implicit Locks II: More interesting commands

Problem 26:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN ROW EXCLUSIVE MODE;	3
4	4 CREATE INDEX ON t(a);
5 COMMIT;	

Problem 27:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN ROW EXCLUSIVE MODE;	3
4	4 CREATE INDEX CONCURRENTLY ON t(a);
5 COMMIT;	

Problem 28:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN SHARE MODE;	3
4	4 VACUUM t;
5 COMMIT;	

Problem 29:

Session 1	Session 2
1 CREATE TABLE t (a INT);	1
2 BEGIN;	2
3 LOCK TABLE t IN ROW SHARE MODE;	3
4	4 VACUUM FULL t;
5 COMMIT;	

Problem 30:

Session 1

```
1  CREATE TABLE t ( a INT );
2  BEGIN;
3  LOCK TABLE t IN ROW EXCLUSIVE MODE;
4
5  COMMIT;
```

Session 2

```
1
2
3
4  ANALYZE t;
```

Problem 31:

Session 1

```
1  CREATE TABLE t ( a INT );
2  BEGIN;
3  LOCK TABLE t IN ROW EXCLUSIVE MODE;
4
5  COMMIT;
```

Session 2

```
1
2
3
4  CLUSTER t;
```

Problem 32:

Session 1

```
1  CREATE TABLE t ( a INT );
2  BEGIN;
3  INSERT INTO t VALUES (5);
4
5
6  COMMIT;
```

Session 2

```
1
2
3
4  BEGIN;
5  INSERT INTO t VALUES (6);
6
7  COMMIT;
```

7 Implicit Locks III: Row Level Locks

Problem 33:

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
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  UPDATE t SET a=7 WHERE a=1;
9  COMMIT;
```

Problem 34:

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
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  DELETE FROM t WHERE a=2;
9  COMMIT;
```

Problem 35:

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
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  DELETE FROM t WHERE a=1;
9  COMMIT;
```

Problem 36:

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
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  DELETE FROM t;
9  COMMIT;
```

Problem 37:

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  DELETE FROM t WHERE a=3;
8
9
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  UPDATE t SET a=5;
9  COMMIT;
```

Problem 38:

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  DELETE FROM t WHERE a=4;
8
9
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  UPDATE t SET a=5;
9  COMMIT;
```

Problem 39:

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
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  UPDATE t SET a=7 WHERE a=2;
9  COMMIT;
```

Problem 40:

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  SELECT * FROM t WHERE a=2 FOR UPDATE;
8
9
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  UPDATE t SET a=7 WHERE a=2;
9  COMMIT;
```

Problem 41:

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  SELECT * FROM t WHERE a=2 FOR UPDATE;
8
9
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  UPDATE t SET a=7 WHERE a=3;
9  COMMIT;
```

Problem 42:

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  SELECT * FROM t FOR UPDATE;
8
9
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7
8  UPDATE t SET a=7 WHERE a=3;
9  COMMIT;
```

8 Implicit Locks IV: Unique constraints

Problem 43:

Session 1

```
1  CREATE TABLE t ( a INT UNIQUE );
2  BEGIN;
3  INSERT INTO t VALUES (5);
4
5
6  COMMIT;
```

Session 2

```
1
2
3
4  BEGIN;
5  INSERT INTO t VALUES (6);
6
7  COMMIT;
```

Problem 44:

Session 1

```
1  CREATE TABLE t ( a INT UNIQUE );
2  BEGIN;
3  INSERT INTO t VALUES (NULL);
4
5
6  COMMIT;
```

Session 2

```
1
2
3
4  BEGIN;
5  INSERT INTO t VALUES (6);
6
7  COMMIT;
```

Problem 45:

Session 1

```
1  CREATE TABLE t ( a INT UNIQUE );
2  BEGIN;
3  INSERT INTO t VALUES (5);
4
5
6  COMMIT;
```

Session 2

```
1
2
3
4  BEGIN;
5  INSERT INTO t VALUES (NULL);
6
7  COMMIT;
```

Problem 46:

Session 1

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

Session 2

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

Problem 47:

Session 1

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

Session 2

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

Problem 48:

Session 1

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

Session 2

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

9 Isolation Levels II: Row Level

Problem 49:

Session 1

```
1  CREATE TABLE t ( a INT );
2  INSERT INTO t VALUES (9);
3  INSERT INTO t VALUES (10);
4  BEGIN;
5
6  UPDATE t SET a = a+1;
7
8
9  COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7  DELETE FROM t WHERE a=10;
8  COMMIT;
9  SELECT count(*) FROM t;
```

Problem 50:

Session 1

```
1  CREATE TABLE t ( a INT );
2  INSERT INTO t VALUES (9);
3  INSERT INTO t VALUES (10);
4  BEGIN;
5
6  UPDATE t SET a = a+1;
7
8
9  ABORT;
```

Session 2

```
1
2
3
4
5  BEGIN;
6
7  DELETE FROM t WHERE a=10;
8  COMMIT;
9  SELECT count(*) FROM t;
```

Problem 51:

Session 1

```
1  CREATE TABLE t ( a INT );
2  INSERT INTO t VALUES (9);
3  INSERT INTO t VALUES (10);
4  BEGIN;
5
6
7  UPDATE t SET a = a+1;
8
9
10 COMMIT;
```

Session 2

```
1
2
3
4
5  BEGIN ISOLATION LEVEL
6  REPEATABLE READ;
7
8  DELETE FROM t WHERE a=10;
9  COMMIT;
```

Problem 52:

Session 1

```
1  CREATE TABLE t ( a INT );
2  INSERT INTO t VALUES (9);
3  INSERT INTO t VALUES (10);
4  BEGIN;
5
6
7  UPDATE t SET a = a+1;
8
9
10 ABORT;
```

Session 2

```
1
2
3
4
5  BEGIN ISOLATION LEVEL
6  REPEATABLE READ;
7
8  DELETE FROM t WHERE a=10;
9  COMMIT;
```

10 Foreign Keys

Problem 53:

Session 1

```
1  CREATE TABLE t(a INT UNIQUE);
2  CREATE TABLE u(b INT REFERENCES t(a));
3  BEGIN;
4
5  INSERT INTO t VALUES (9);
6
7  COMMIT;
```

Session 2

```
1
2
3
4  BEGIN;
5
6  INSERT INTO u VALUES (9);
7
8  COMMIT;
```

Problem 54:

Session 1

```
1  CREATE TABLE t(a INT UNIQUE);
2  CREATE TABLE u(b INT REFERENCES t(a));
3
4  INSERT INTO t VALUES (9);
```

Session 2

```
1
2
3  BEGIN;
4
5  INSERT INTO u VALUES (9);
6  COMMIT;
```

Problem 55:

Session 1

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

Session 2

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

Problem 56:

Session 1

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

Session 2

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

Problem 57:

Session 1

```
1  CREATE TABLE t(a INT UNIQUE);
2  CREATE TABLE u(b INT REFERENCES t(a));
3  INSERT INTO t VALUES (8);
4  INSERT INTO t VALUES (9);
5  BEGIN;
6
7  INSERT INTO u VALUES (9);
8
9  ABORT;
```

Session 2

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

Problem 58:

Session 1

```
1  CREATE TABLE t(a INT UNIQUE);
2  CREATE TABLE u(b INT REFERENCES t(a));
3  INSERT INTO t VALUES (8);
4  INSERT INTO t VALUES (9);
5  BEGIN;
6
7  INSERT INTO u VALUES (9);
8
9  ABORT;
```

Session 2

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

Problem 59:

Session 1

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

Session 2

```
1
2
3
4
5
6  BEGIN;
7  DELETE FROM t WHERE a=9;
8
9  COMMIT;
```

Problem 60:

Session 1

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

Session 2

```
1
2
3
4
5
6  BEGIN;
7  DELETE FROM t WHERE a=8;
8
9  COMMIT;
```

11 Deferring Constraints

Note: For each problem, you should consider what would happen both with and without the `DEFERRABLE INITIALLY DEFERRED` line.

Problem 61:

Session 1

```
1  CREATE TABLE t (
2      a INT UNIQUE
3      DEFERRABLE INITIALLY DEFERRED
4      );
5  BEGIN;
6  INSERT INTO t VALUES (8);
7  INSERT INTO t VALUES (8);
8  COMMIT;
```

Session 2

Problem 62:

Session 1

```
1  CREATE TABLE t (a INT UNIQUE);
2  CREATE TABLE u (
3      b INT REFERENCES t(a)
4      DEFERRABLE INITIALLY DEFERRED
5      );
6  BEGIN;
7
8
9  INSERT INTO t VALUES (8);
10 COMMIT;
```

Session 2

```
1
2
3
4
5
6
7  BEGIN;
8  INSERT INTO u VALUES (8);
9
10
11 COMMIT;
```

Problem 63:

Session 1

```
1  CREATE TABLE t (a INT UNIQUE);
2  CREATE TABLE u (
3      b INT REFERENCES t(a)
4      DEFERRABLE INITIALLY DEFERRED
5      );
6  BEGIN;
7
8
9  INSERT INTO t VALUES (8);
10 ABORT;
```

Session 2

```
1
2
3
4
5
6
7  BEGIN;
8  INSERT INTO u VALUES (8);
9
10
11 COMMIT;
```

Problem 64:

Session 1

```
1  CREATE TABLE t (a INT UNIQUE);
2  CREATE TABLE u (
3      b INT REFERENCES t(a)
4      DEFERRABLE INITIALLY DEFERRED
5      );
6  BEGIN;
7
8
9  LOCK TABLE t IN EXCLUSIVE MODE;
10
11 INSERT INTO t VALUES (8);
12 COMMIT;
```

Session 2

```
1
2
3
4
5
6
7  BEGIN;
8  INSERT INTO u VALUES (8);
9
10 COMMIT;
```

Problem 65:

Session 1

```
1  CREATE TABLE t (a INT UNIQUE);
2  CREATE TABLE u (
3      b INT REFERENCES t(a)
4      DEFERRABLE INITIALLY DEFERRED
5      );
6  BEGIN;
7
8
9  LOCK TABLE t IN EXCLUSIVE MODE;
10
11 COMMIT;
```

Session 2

```
1
2
3
4
5
6
7  BEGIN;
8  INSERT INTO u VALUES (8);
9
10 COMMIT;
```

12 Everything at Once

Problem 66:

Session 1

```
1  CREATE TABLE t (a INT PRIMARY KEY);
2
3
4
5  BEGIN;
6  INSERT INTO t VALUES (1);
7  INSERT INTO t VALUES (2);
8  INSERT INTO u VALUES (2);
9
10
11
12 LOCK TABLE t IN EXCLUSIVE MODE;
13 SELECT count(*) FORM t;
14
15
16 INSERT INTO t VALUES (NULL);
17 COMMIT;
18
19
20
21 SELECT count(*) FROM u;
```

Session 2

```
1
2  CREATE TABLE u (
3      b INT REFERENCES t(a)
4      );
5
6
7
8
9  BEGIN ISOLATION LEVEL
10 REPEATABLE READ;
11 SELECT count(*) FROM u;
12
13
14 INSERT INTO u VALUES (1);
15 SELECT count(*) FROM u;
16
17
18 INSERT INTO u VALUES (NULL);
19 COMMIT;
20 SELECT count(*) FROM t;
```

Problem 67:

Session 1

```
1  CREATE TABLE t (a INT PRIMARY KEY);
2
3  BEGIN;
4  INSERT INTO t VALUES (1);
5  INSERT INTO t VALUES (2);
6  INSERT INTO u VALUES (2);
7
8
9
10 LOCK TABLE u IN ROWEXCLUSIVE MODE;
11 SELECT count(*) FORM t;
12
13
14 INSERT INTO t VALUES (3);
15 COMMIT;
16
17
18
19 SELECT count(*) FROM t;
```

Session 2

```
1
2  CREATE TABLE u (b INT NOT NULL);
3
4
5
6
7  BEGIN ISOLATION LEVEL
8  REPEATABLE READ;
9  INSERT INTO t VALUES (2);
10
11
12 INSERT INTO u VALUES (1);
13 SELECT count(*) FROM u;
14
15
16 UPDATE u SET b=5;
17 COMMIT;
18 SELECT count(*) FROM t;
```