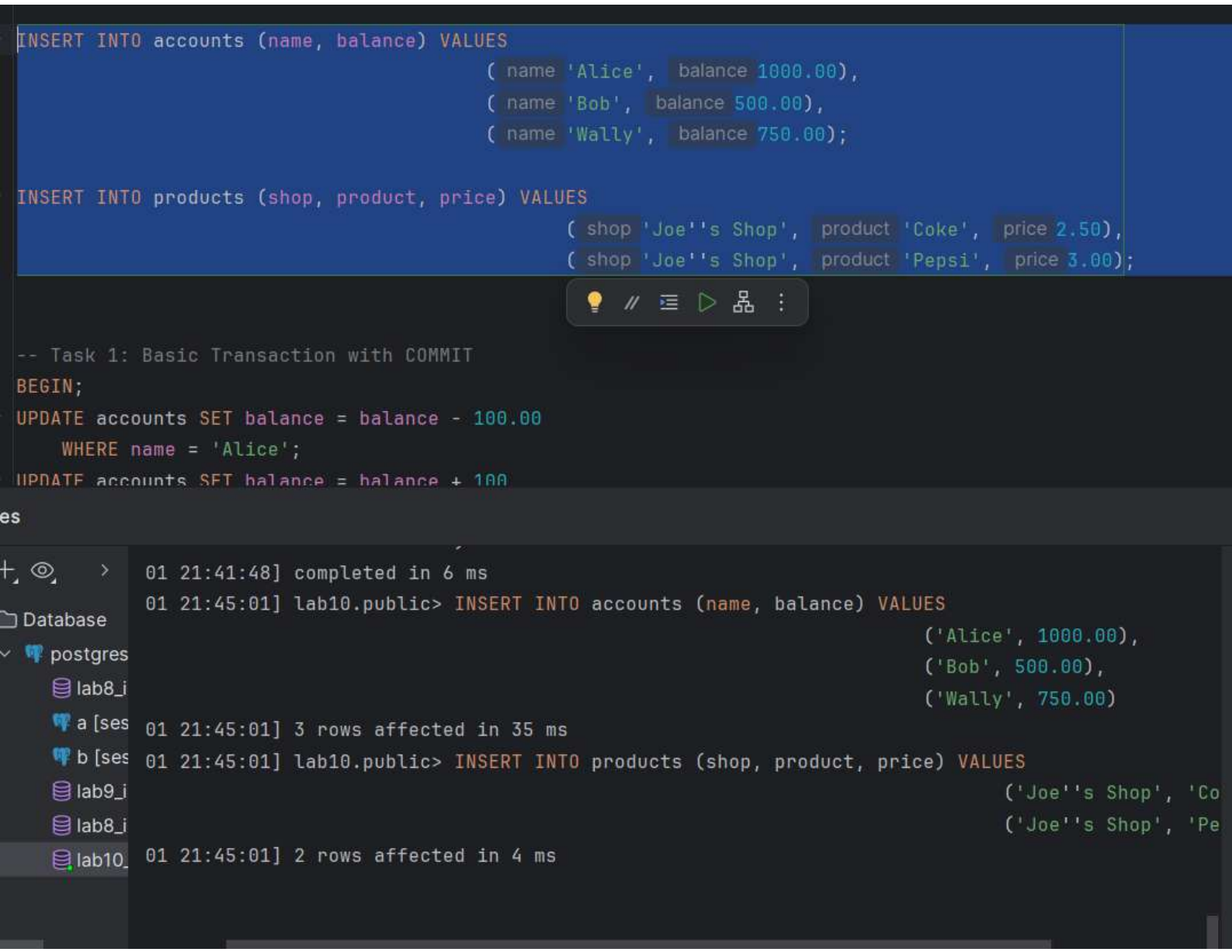
```sql
CREATE TABLE accounts (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    balance DECIMAL(10, 2) DEFAULT 0.00
);

CREATE TABLE products (
    id SERIAL PRIMARY KEY,
    shop VARCHAR(100) NOT NULL,
    product VARCHAR(100) NOT NULL,
    price DECIMAL(10, 2) NOT NULL
);

INSERT INTO accounts (name, balance) VALUES
                                    ( name 'Alice', balance 1000.00),
                                    ( name 'Bob'   balance 500.00)
```

s

Database
postgres
lab8_i
a [ses
b [ses
lab9_i
lab8_i
lab10_

```
[2025-12-01 21:41:48] lab10.public> CREATE TABLE accounts (
                                        id SERIAL PRIMARY KEY,
                                        name VARCHAR(100) NOT NULL,
                                        balance DECIMAL(10, 2) DEFAULT 0.00
                                    )
[2025-12-01 21:41:48] completed in 38 ms
[2025-12-01 21:41:48] lab10.public> CREATE TABLE products (
                                        id SERIAL PRIMARY KEY,
                                        shop VARCHAR(100) NOT NULL,
                                        product VARCHAR(100) NOT NULL,
                                        price DECIMAL(10, 2) NOT NULL
                                    )
[2025-12-01 21:41:48] completed in 6 ms
```


INSERT INTO accounts (name, balance) VALUES
(name 'Alice', balance 1000.00),
(name 'Bob', balance 500.00),
(name 'Wally', balance 750.00);
INSERT INTO products (shop, product, price) VALUES
(shop 'Joe''s Shop', product 'Coke', price 2.50),
(shop 'Joe''s Shop', product 'Pepsi', price 3.00);
-- Task 1: Basic Transaction with COMMIT
BEGIN;
UPDATE accounts SET balance = balance - 100.00
WHERE name = 'Alice';
Database
postgres
lab8_i
a [ses
b [ses
lab9_i
lab8_i
lab10_
01 21:41:48] completed in 6 ms
01 21:45:01] lab10.public> INSERT INTO accounts (name, balance) VALUES
('Alice', 1000.00),
('Bob', 500.00),
('Wally', 750.00)
01 21:45:01] 3 rows affected in 35 ms
01 21:45:01] lab10.public> INSERT INTO products (shop, product, price) VALUES
('Joe''s Shop', 'Co
('Joe''s Shop', 'Pe
01 21:45:01] 2 rows affected in 4 ms

```sql
-- Task 1: Basic Transaction with COMMIT
BEGIN;
UPDATE accounts SET balance = balance - 100.00
    WHERE name = 'Alice';
UPDATE accounts SET balance = balance + 100.00
    WHERE name = 'Bob';
COMMIT;

SELECT name, balance FROM accounts WHERE name IN ('Alice','Bob');
/*
a)Alice: 900.00, Bob: 600.00.
b)atomicity – both updates succeed together or not at all.
c)Without a transaction, partial update could persist leaving inconsistent state
*/

-- Task 2: Using ROLLBACK
```

s

Output | lab10.public.accounts

Database

postgres
lab8_i
a [ses
b [ses
lab9_i
lab8_i
lab10

Tx: Auto | DDL | CSV

	name	balance
1	Alice	900.00
2	Bob	600.00

```sql
-- Task 2: Using ROLLBACK
BEGIN;
UPDATE accounts SET balance = balance - 500.00
    WHERE name = 'Alice';
SELECT * FROM accounts WHERE name = 'Alice';

ROLLBACK;

SELECT * FROM accounts WHERE name = 'Alice';
```

```
a)After UPDATE before ROLLBACK you see decreased balance in this session
b)After ROLLBACK balance returns to original
c)Use ROLLBACK when an error occurs or validation fails
*/
```

ces

Output | lab10.public.accounts | lab10.public.accounts 2

Database

postgres

lab8_i

a [ses

b [ses

lab9_i

lab8_i

lab10

Tx: Auto | DDL

	id	name	balance
1	1	Alice	900.00

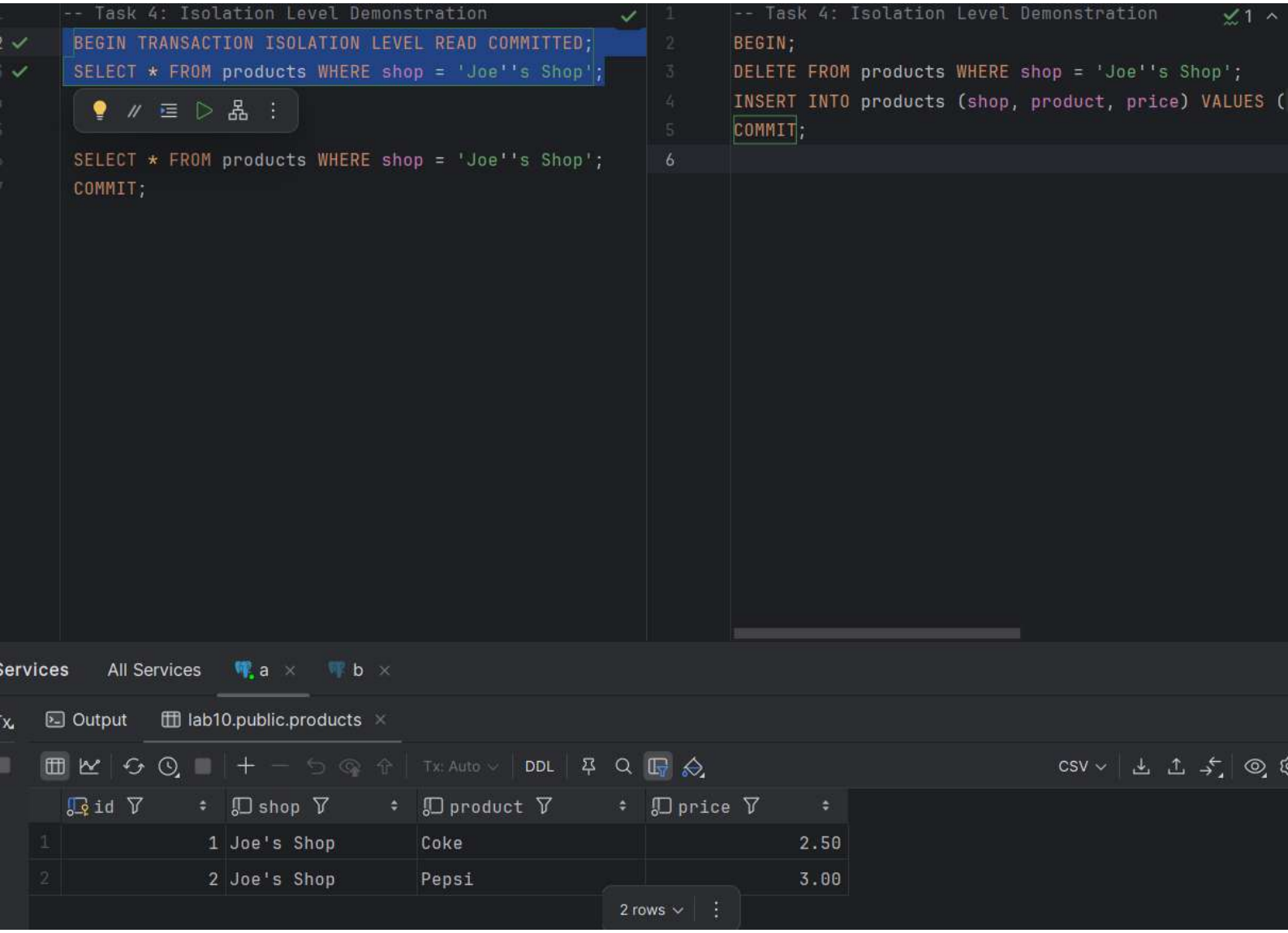
```sql
-- Task 3: Working with SAVEPOINTs
BEGIN;
UPDATE accounts SET balance = balance - 100.00
    WHERE name = 'Alice';

SAVEPOINT my_savepoint;

UPDATE accounts SET balance = balance + 100.00
    WHERE name = 'Bob';

ROLLBACK TO my_savepoint;

UPDATE accounts SET balance = balance + 100.00
    WHERE name = 'Wally';

COMMIT;

/*
a)Alice: decreased by 100 Bob: unchanged Wally: increased by 100
b) Bob was not credited in final state because we rolled back the step that credited Bob before commit.
c) SAVEPOINT allows partial undo inside a long transaction without aborting everything.
*/
```

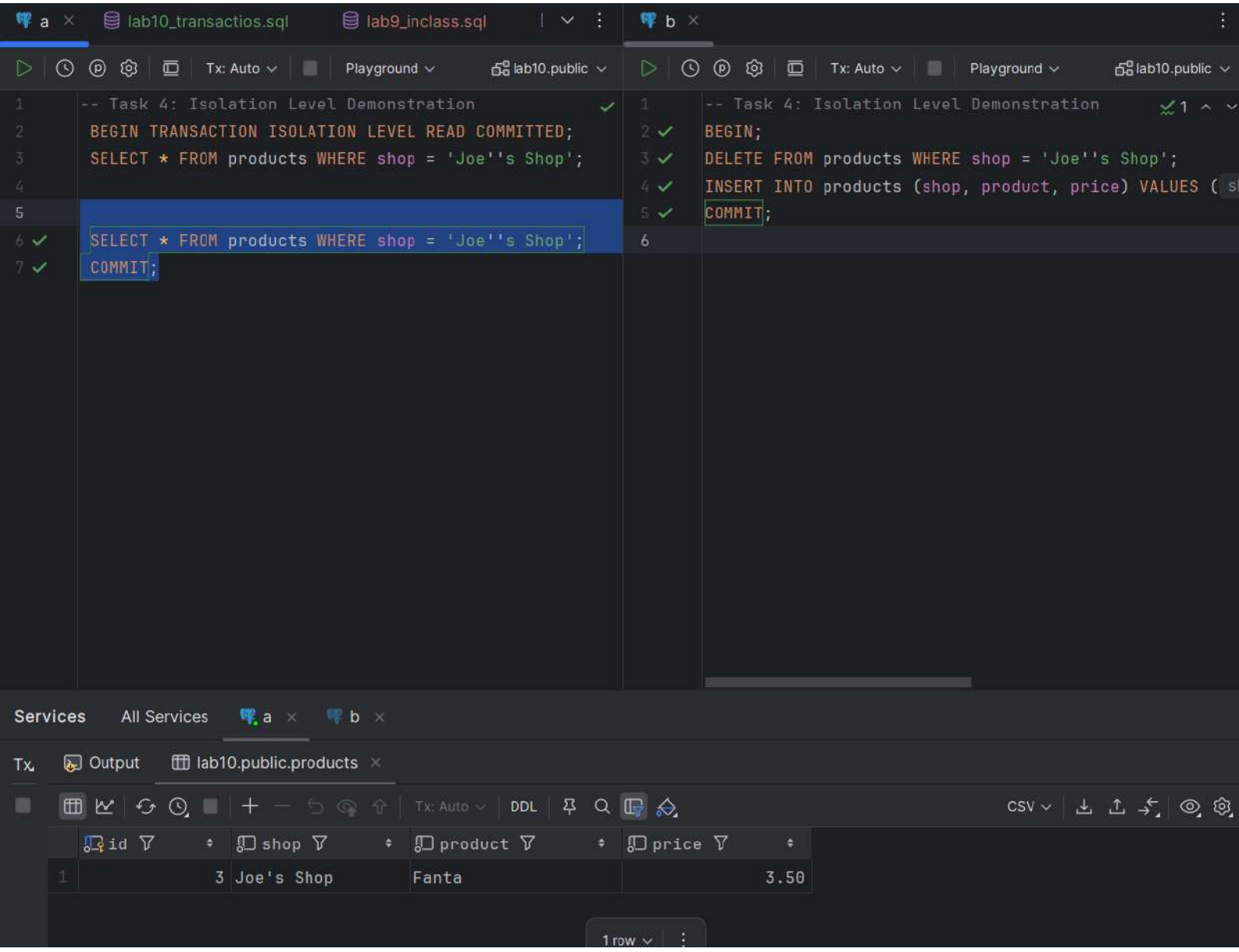
ices

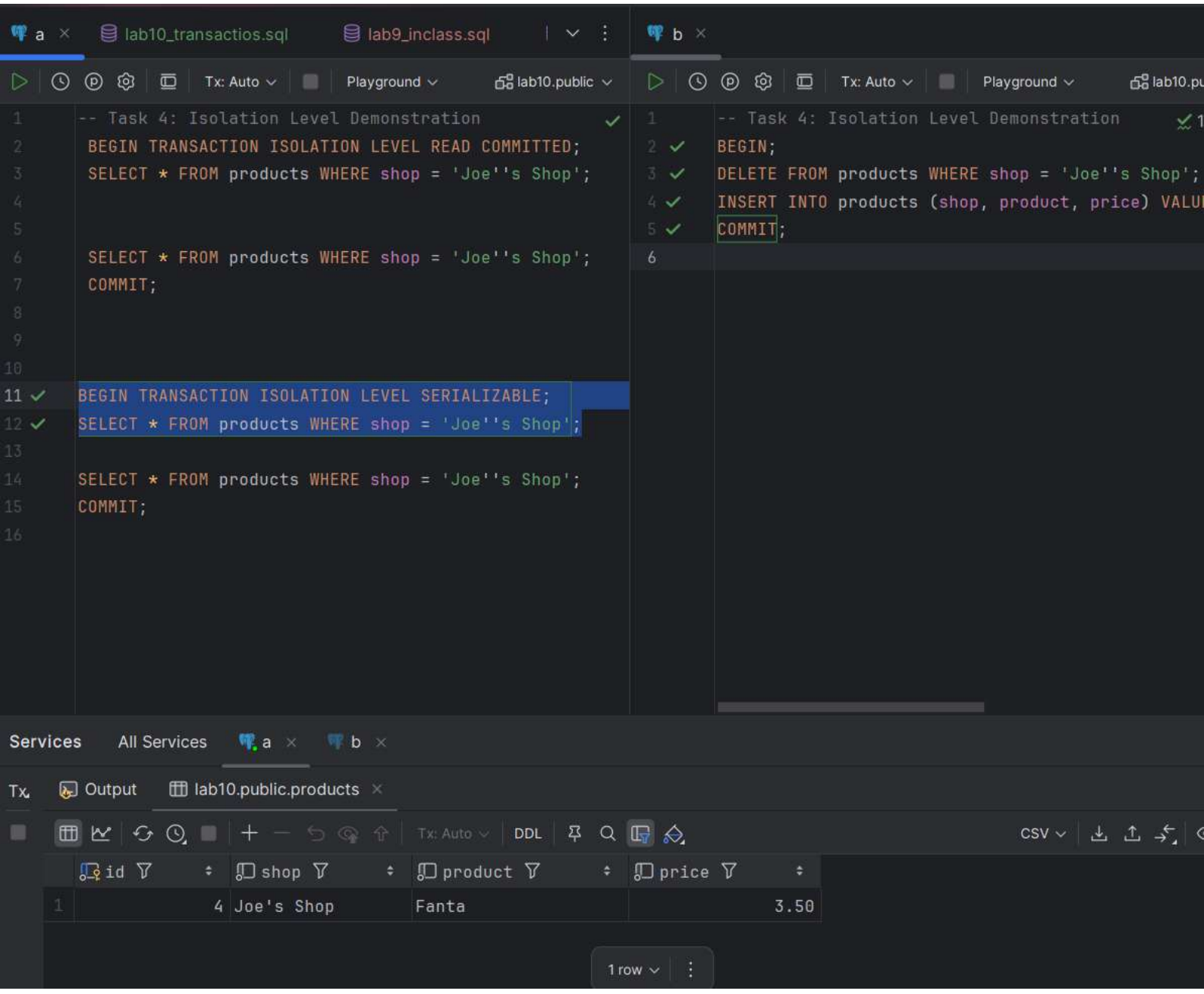
Output | lab10.public.accounts | lab10.public.accounts 2

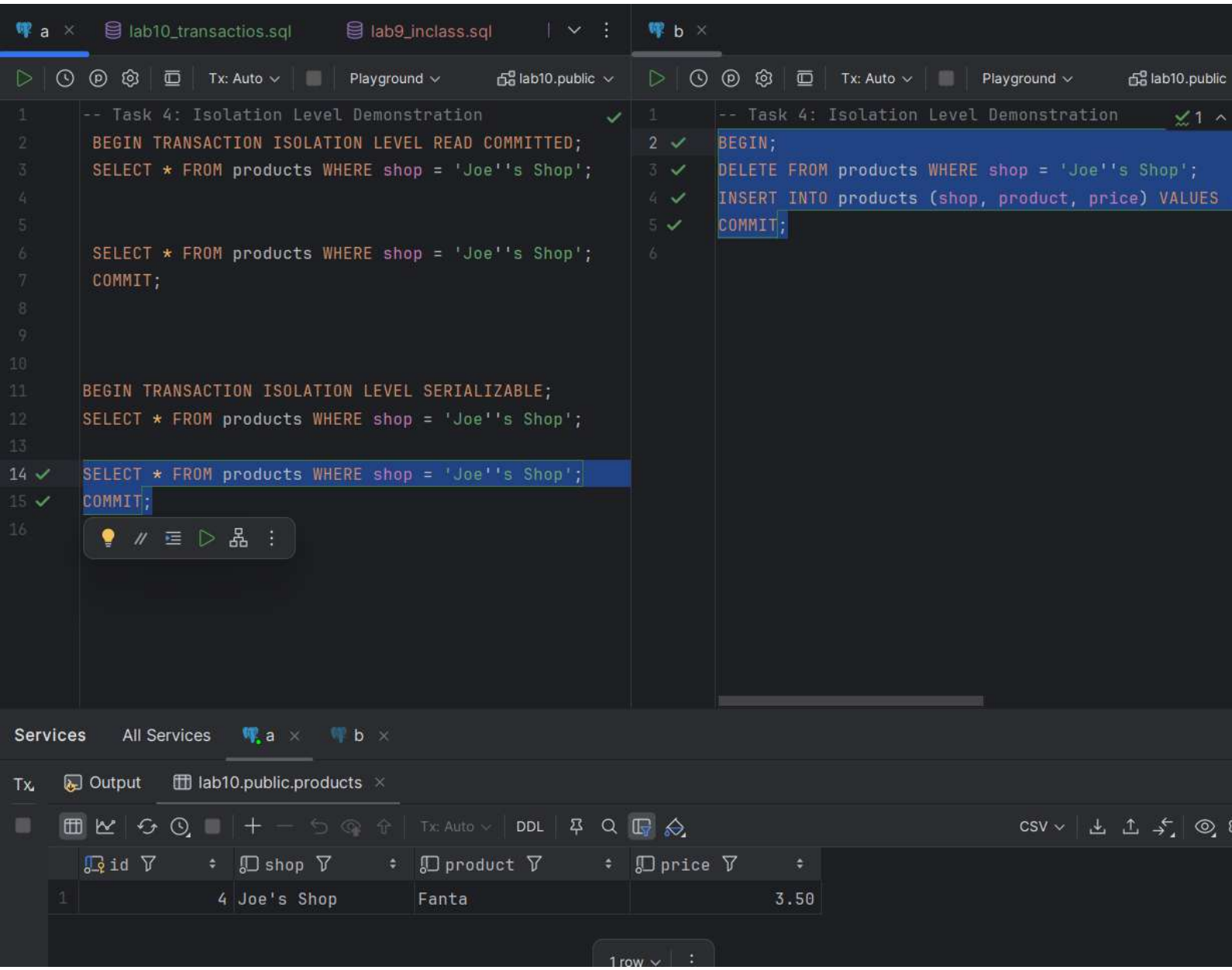
Database
postgres
lab8_i
a [ses
b [ses

```
                                WHERE name = 'Wally'
01 21:50:27] 1 row affected in 2 ms
01 21:50:29] lab10.public> COMMIT
01 21:50:29] completed in 2 ms
```

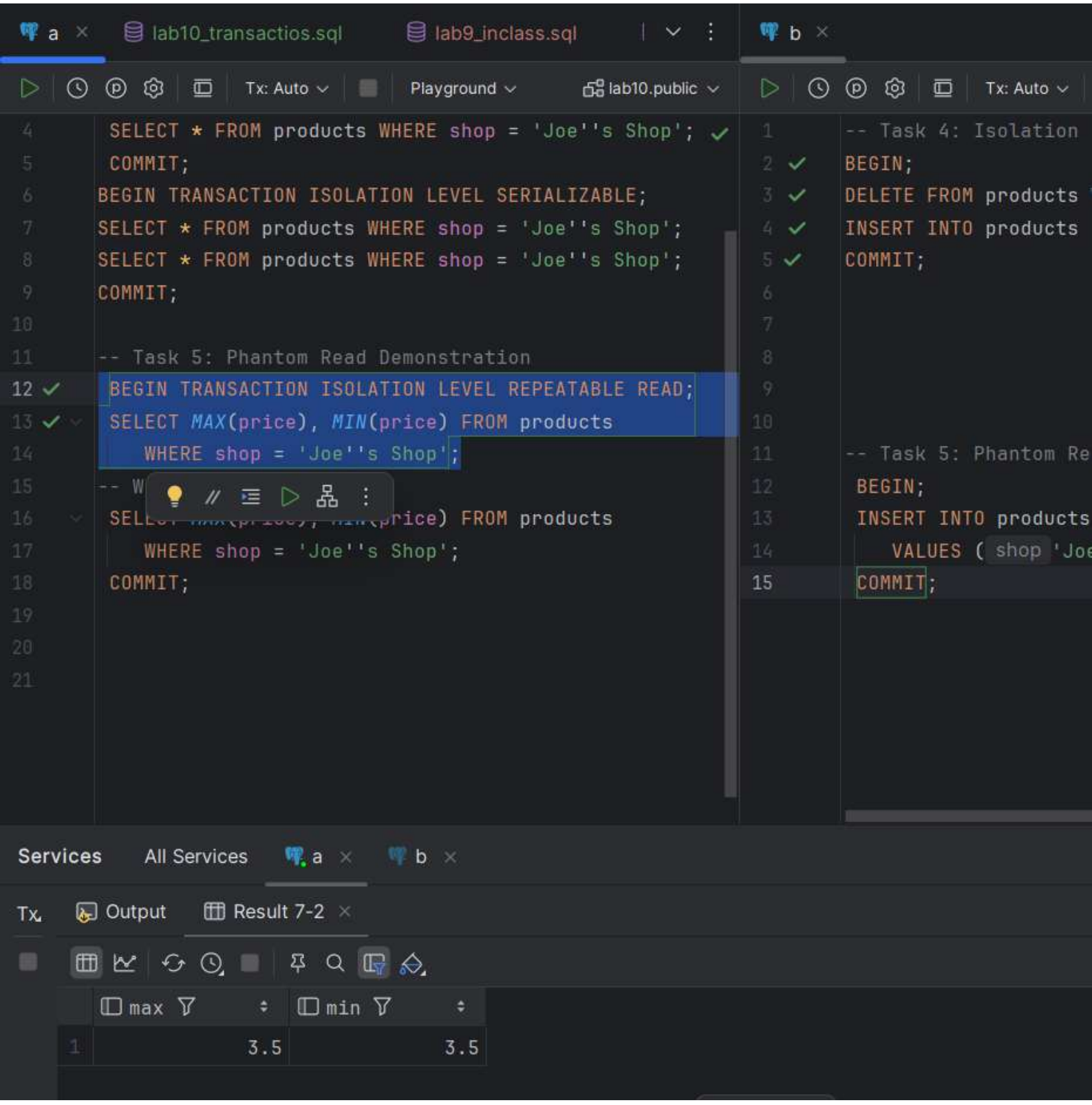

-- Task 4: Isolation Level Demonstration
BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 4: Isolation Level Demonstration
BEGIN;
DELETE FROM products WHERE shop = 'Joe''s Shop';
INSERT INTO products (shop, product, price) VALUES (
COMMIT;
Services
All Services
a
b
Output
lab10.public.products
Tx: Auto
DDL
CSV
id
shop
product
price
1 1 Joe's Shop Coke 2.50
2 2 Joe's Shop Pepsi 3.00
2 rows

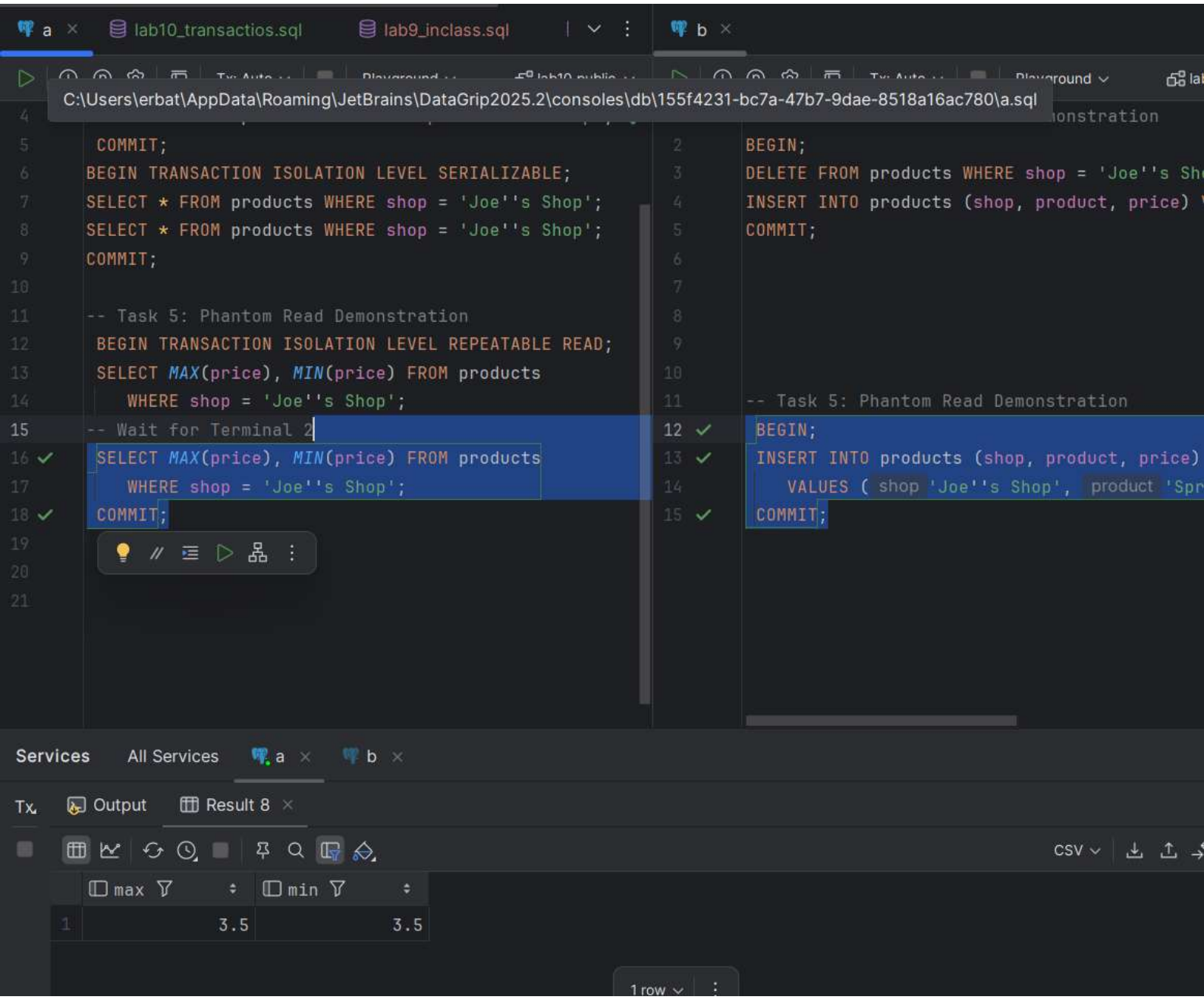

a
lab10_transactios.sql
lab9_inclass.sql
Tx: Auto
Playground
lab10.public
-- Task 4: Isolation Level Demonstration
BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
b
Tx: Auto
Playground
lab10.public
-- Task 4: Isolation Level Demonstration
BEGIN;
DELETE FROM products WHERE shop = 'Joe''s Shop';
INSERT INTO products (shop, product, price) VALUES (
COMMIT;
Services
All Services
a
b
Tx
Output
lab10.public.products
Tx: Auto
DDL
CSV
id
shop
product
price
1
3
Joe's Shop
Fanta
3.50
1 row

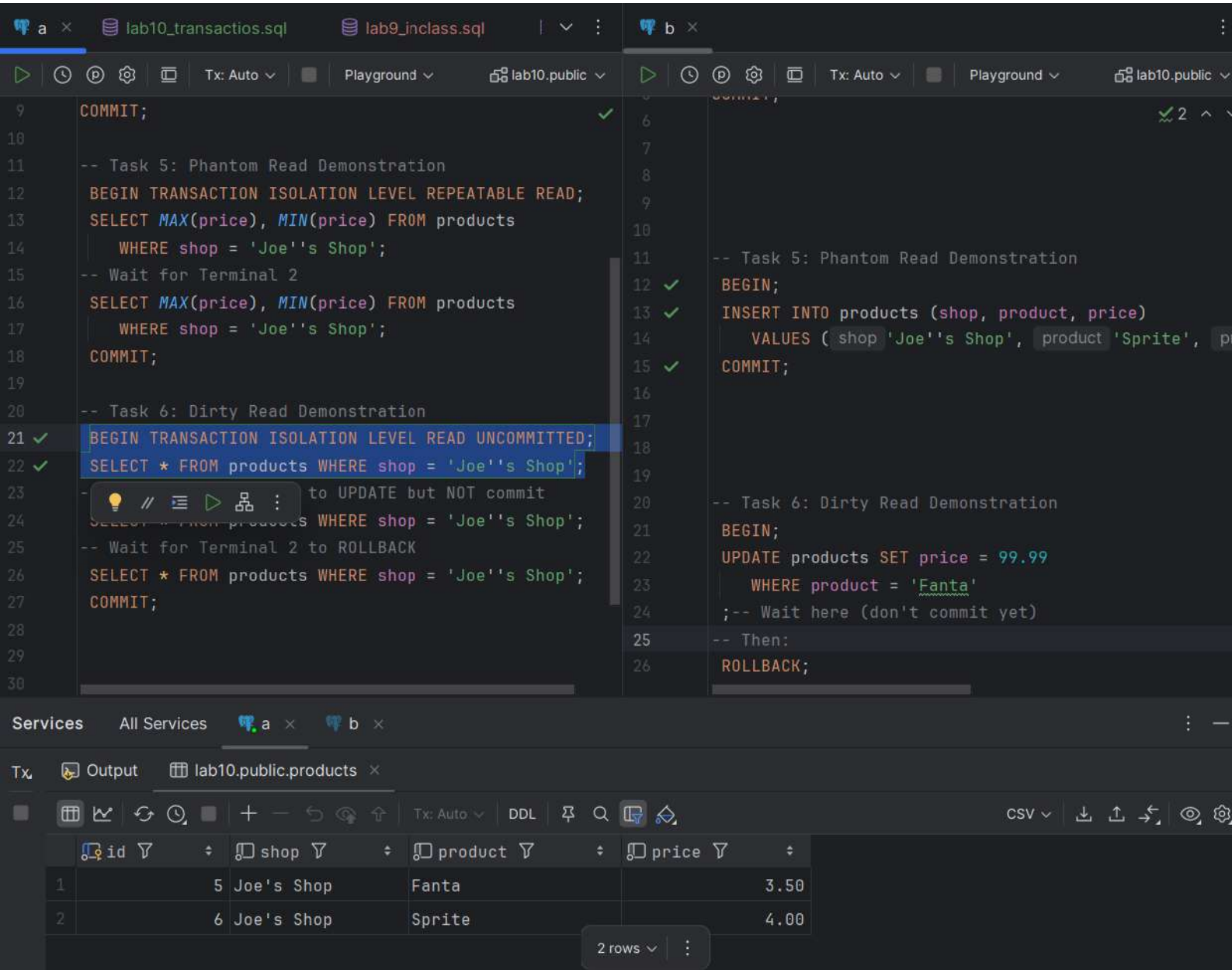

a
lab10_transactios.sql
lab9_inclass.sql
b
Tx: Auto
Playground
lab10.public
-- Task 4: Isolation Level Demonstration
BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 4: Isolation Level Demonstration
BEGIN;
DELETE FROM products WHERE shop = 'Joe''s Shop';
INSERT INTO products (shop, product, price) VALU
COMMIT;
Services
All Services
a
b
Output
lab10.public.products
Tx: Auto
DDL
CSV
id
shop
product
price
1
4
Joe's Shop
Fanta
3.50
1 row

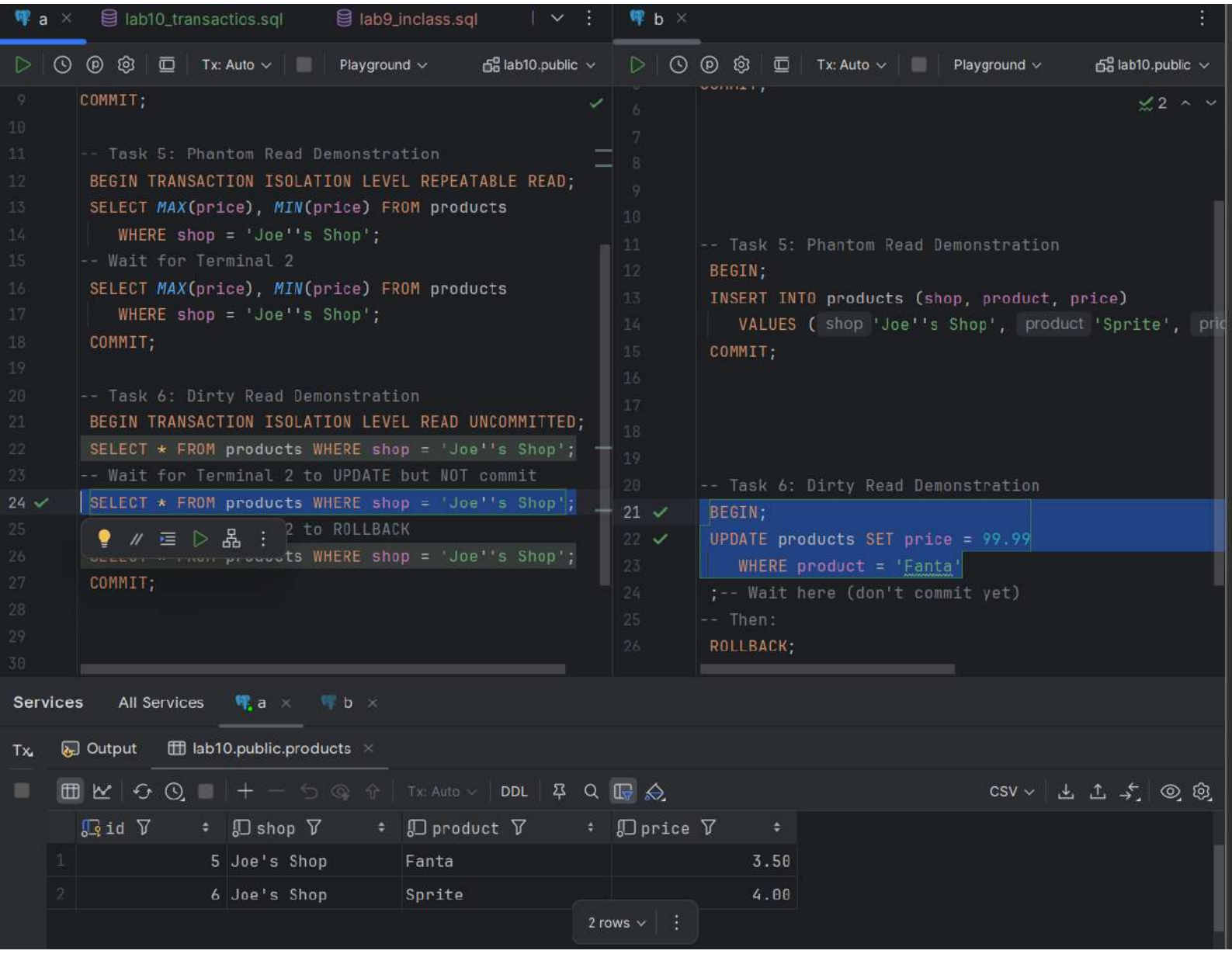

a
lab10_transactios.sql
lab9_inclass.sql
b
Tx: Auto
Playground
lab10.public
-- Task 4: Isolation Level Demonstration
BEGIN TRANSACTION ISOLATION LEVEL READ COMMITTED;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 4: Isolation Level Demonstration
BEGIN;
DELETE FROM products WHERE shop = 'Joe''s Shop';
INSERT INTO products (shop, product, price) VALUES
COMMIT;

Services
All Services
a
b
Output
lab10.public.products
Tx: Auto
DDL
CSV
id
shop
product
price
1
4
Joe's Shop
Fanta
3.50
1 row


a
lab10_transactios.sql
lab9_inclass.sql
b
Tx: Auto
Playground
lab10.public
Tx: Auto
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 5: Phantom Read Demonstration
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT MAX(price), MIN(price) FROM products
WHERE shop = 'Joe''s Shop';
SELECT MAX(price), MIN(price) FROM products
WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 4: Isolation
BEGIN;
DELETE FROM products
INSERT INTO products
COMMIT;
-- Task 5: Phantom Re
BEGIN;
INSERT INTO products
VALUES (shop 'Joe
COMMIT;
Services
All Services
a
b
Tx
Output
Result 7-2
max
min
1
3.5
3.5


a
lab10_transactios.sql
lab9_inclass.sql
b
C:\Users\erbat\AppData\Roaming\JetBrains\DataGrip2025.2\consoles\db\155f4231-bc7a-47b7-9dae-8518a16ac780\a.sql
COMMIT;
BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 5: Phantom Read Demonstration
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT MAX(price), MIN(price) FROM products
WHERE shop = 'Joe''s Shop';
-- Wait for Terminal 2
SELECT MAX(price), MIN(price) FROM products
WHERE shop = 'Joe''s Shop';
COMMIT;
BEGIN;
DELETE FROM products WHERE shop = 'Joe''s Sh
INSERT INTO products (shop, product, price)
COMMIT;
-- Task 5: Phantom Read Demonstration
BEGIN;
INSERT INTO products (shop, product, price)
VALUES (shop 'Joe''s Shop', product 'Spr
COMMIT;
Services
All Services
a
b
Output
Result 8
CSV
max
min
1
3.5
3.5
1 row


a
lab10_transactios.sql
lab9_inclass.sql
Tx: Auto
Playground
lab10.public
COMMIT;
-- Task 5: Phantom Read Demonstration
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT MAX(price), MIN(price) FROM products
WHERE shop = 'Joe''s Shop';
-- Wait for Terminal 2
SELECT MAX(price), MIN(price) FROM products
WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 6: Dirty Read Demonstration
BEGIN TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
to UPDATE but NOT commit
WHERE shop = 'Joe''s Shop';
-- Wait for Terminal 2 to ROLLBACK
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
b
Tx: Auto
Playground
lab10.public
-- Task 5: Phantom Read Demonstration
BEGIN;
INSERT INTO products (shop, product, price)
VALUES (shop 'Joe''s Shop', product 'Sprite',
COMMIT;
-- Task 6: Dirty Read Demonstration
BEGIN;
UPDATE products SET price = 99.99
WHERE product = 'Fanta'
;-- Wait here (don't commit yet)
-- Then:
ROLLBACK;
Services
All Services
a
b
Tx
Output
lab10.public.products
Tx: Auto
DDL
CSV
id
shop
product
price
1 5 Joe's Shop Fanta 3.50
2 6 Joe's Shop Sprite 4.00
2 rows


a
lab10_transactics.sql
lab9_inclass.sql
b
Tx: Auto
Playground
lab10.public
COMMIT;
-- Task 5: Phantom Read Demonstration
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT MAX(price), MIN(price) FROM products
WHERE shop = 'Joe''s Shop';
-- Wait for Terminal 2
SELECT MAX(price), MIN(price) FROM products
WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 6: Dirty Read Demonstration
BEGIN TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
-- Wait for Terminal 2 to UPDATE but NOT commit
SELECT * FROM products WHERE shop = 'Joe''s Shop';
2 to ROLLBACK
ts WHERE shop = 'Joe''s Shop';
COMMIT;
-- Task 5: Phantom Read Demonstration
BEGIN;
INSERT INTO products (shop, product, price)
VALUES (shop 'Joe''s Shop', product 'Sprite', pric
COMMIT;
-- Task 6: Dirty Read Demonstration
BEGIN;
UPDATE products SET price = 99.99
WHERE product = 'Fanta'
;-- Wait here (don't commit yet)
-- Then:
ROLLBACK;
Services
All Services
Output
lab10.public.products
DDL
CSV
id
shop
product
price
1 5 Joe's Shop Fanta 3.50
2 6 Joe's Shop Sprite 4.00
2 rows

a × lab10_transactios.sql lab9_inclass.sql

Tx: Auto Playground lab10.public

```
COMMIT;

-- Task 5: Phantom Read Demonstration
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT MAX(price), MIN(price) FROM products
   WHERE shop = 'Joe''s Shop';
-- Wait for Terminal 2
SELECT MAX(price), MIN(price) FROM products
   WHERE shop = 'Joe''s Shop';
COMMIT;

-- Task 6: Dirty Read Demonstration
BEGIN TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
SELECT * FROM products WHERE shop = 'Joe''s Shop';
-- Wait for Terminal 2 to UPDATE but NOT commit
SELECT * FROM products WHERE shop = 'Joe''s Shop';
-- Wait for Terminal 2 to ROLLBACK
SELECT * FROM products WHERE shop = 'Joe''s Shop';
COMMIT;
```

b ×

Tx: Auto Playground lab10.public

```
-- Task 5: Phantom Read Demonstration
BEGIN;
INSERT INTO products (shop, product, price)
   VALUES ( shop 'Joe''s Shop', product 'Sprite', p
COMMIT;

-- Task 6: Dirty Read Demonstration
BEGIN;
UPDATE products SET price = 99.99
   WHERE product = 'Fanta'
;-- Wait here (don't commit yet)
-- Then:
ROLLBACK;
```

Services All Services a × b ×

Tx Output lab10.public.products ×

Tx: Auto DDL CSV

	id	shop	product	price
1	5	Joe's Shop	Fanta	3.50
2	6	Joe's Shop	Sprite	4.00

2 rows

```
-- Exercise 2
BEGIN;
INSERT INTO products (shop, product, price) VALUES ( shop 'TestShop', product 'TempProduct', price 10.00);
SAVEPOINT sp1;
UPDATE products SET price = 12.00 WHERE shop='TestShop' AND product='TempProduct';
SAVEPOINT sp2;
DELETE FROM products WHERE shop='TestShop' AND product='TempProduct';

ROLLBACK TO sp1;
COMMIT;

SELECT * FROM products WHERE shop='TestShop';
```

Services | All Services | a | b

Output | lab10.public.accounts | lab10.public.products

Tx: Auto | DDL | CSV

	id	shop	product	price
1	7	TestShop	TempProduct	10.00

1 row