

Four Non-Conflicting Transactions

START transaction;

-- start T1

START transaction;

```
Select e.employee_id, CONCAT(e.first_name, ' ', e.last_name) AS full_name
from employee e
join employee_service es on e.employee_id = es.employee_id
where es.service_id = 2 and
e.status = 'unoccupied' and
e.rating > (select AVG(ee.rating) from employee ee);
```

-- start T2

START transaction;

```
SELECT employee_id, email, experience, rating
from employee
order by employee_id ASC;
```

-- commit T2

commit;

-- commit T1

commit;

commit;

STEP	T1	T2
1.	START	
2.		START
3.	R(employee)	
4.		R(employee)
5.		COMMIT
6.	COMMIT	

```
-- Transaction 1
START TRANSACTION;
    -- Updating the quantity of a specific product
    UPDATE product
    SET quantity = quantity + 10
    WHERE product_id = 10;
COMMIT;
```

```
-- Transaction 2
START TRANSACTION;
    -- Updating the price of a different product
    UPDATE product
    SET price = price * 1.1
    WHERE product_id = 10;
COMMIT;
```

STEP	T1	T2
1.	START	
2.		START
3.	W(product)	
4.	COMMIT	
5.		W(product)
6.		COMMIT

```
START TRANSACTION;
    -- Updating the quantity of a specific product
    UPDATE product
    SET quantity = quantity + 10
    WHERE product_id = 11;

    -- Retrieving the details of the updated product
    SELECT product_id, name, quantity
    FROM product
    WHERE product_id = 11;
COMMIT;
```

```
-- Transaction 2
START TRANSACTION;
    -- Updating the price of a different product
    UPDATE product
    SET price = price * 1.1
    WHERE product_id = 12;

    -- Retrieving the details of the updated product
    SELECT product_id, name, price
    FROM product
    WHERE product_id = 12;
COMMIT;
```

```
-- Transaction 3
START TRANSACTION;
    -- Adding a new product
    INSERT INTO product (product_id, name, price, quantity)
    VALUES (21, 'New Product', 250, 15);

    -- Retrieving the details of the newly added product
    SELECT product_id, name, price, quantity
    FROM product
    WHERE product_id = 21;
COMMIT;
```

STEP	T1	T2	T3
1.	START		
2.		START	
3.			START
4.	W(product)		
5.		W(product)	
6.		R(product)	
7.		COMMIT	
8.			W(product)
9.			R(product)
10.			COMMIT
11.	R(product)		
12.	COMMIT		

```
START TRANSACTION;
```

```
-- start T1
```

```
START TRANSACTION;
```

```
-- R1(X)
```

```
SELECT product_id, name
```

```
FROM product
```

```
WHERE pet_category = "Cat";
```

```
-- start T2
```

```
START TRANSACTION;
```

```
-- R2(X)
```

```
SELECT name, brand, rating
```

```
FROM product
```

```
WHERE pet_category = 'Dog';
```

```
-- W1(Y)
```

```
INSERT INTO pets VALUES (61, 'Leo', 'Lahsa Aapso', 2, null, 'Dog');
```

```
-- W1(X)
```

```
INSERT INTO product (Product_ID, Name, Brand, Description, Rating, Product_Type,  
Pet_category, Quantity, Price)
```

```
VALUES (65, 'Toy-11', 'XYZ', 'Desc', 8, 'Toy', 'Dog', 14, 520);
```

```
-- W1(Z)
```

```
INSERT INTO wallet VALUES (6, 61, 500);
```

```
-- commit T1
```

```
COMMIT;
```

```
-- R2(Y)
```

```
SELECT * FROM pets;
```

```
-- W2(Z)
```

```
INSERT INTO wallet VALUES (3, 24, 1500);
```

```
-- commit T2
```

```
COMMIT;
```

```
COMMIT;
```

STEP	T1	T2
1.	START	
2.	R(product(cat))	
3.		START
4.		R(product(dog))
5.		W(pets)
6.		W(products(dog))
7.		W(wallet)
8.		COMMIT
9.	R(pets)	
10.	W(wallet)	
11.	COMMIT	

Two Conflicting Transactions

START TRANSACTION;

-- start T1

START TRANSACTION;

-- R1(X)

SELECT product_id, name

FROM product

WHERE pet_category = "Dog";

-- start T2

START TRANSACTION;

-- R2(X)

SELECT name, brand, rating

FROM product

WHERE pet_category = 'Dog';

SAVEPOINT t2;

-- W1(Y)

INSERT INTO pets VALUES (61, 'Leo', 'Lahsa Aapso', 2, null, 'Dog');

-- W1(X)

```
INSERT INTO product (Product_ID, Name, Brand, Description, Rating, Product_Type,
Pet_category, Quantity, Price)
```

```
VALUES (65, 'Toy-11', 'XYZ', 'Desc', 8, 'Toy', 'Dog', 14, 520);
```

```
-- W1(Z)
```

```
INSERT INTO wallet VALUES (6, 61, 500);
```

```
-- commit T1
```

```
-- rollback to savepoint t2 within T2
```

```
ROLLBACK TO t2;
```

```
-- R2(Y)
```

```
SELECT * FROM pets;
```

```
-- W2(Z)
```

```
INSERT INTO wallet VALUES (3, 24, 1500);
```

```
-- commit T2
```

```
COMMIT;
```

```
COMMIT;
```

STEP	T1	T2
1.	START	
2.	R(product(dog))	
3.		START
4.		R(product(dog))
5.		SAVEPOINT t2
6.		W(pets)
7.		W(product(dog))
8.		W(wallet)
9.	COMMIT	
10.		ROLLBACK t2
11.		R(pets)
12.		W(wallet)
13.		COMMIT

```

START TRANSACTION;
-- Transaction A
START TRANSACTION;

-- Read the current price of the product
SELECT Price FROM product WHERE Product_ID = 9;

-- Read the current balance of the user's wallet
SELECT Amount FROM wallet WHERE User_id = 2;

-- Simulate some delay to allow Transaction A to complete
DO SLEEP(3);

-- Update the wallet balance if sufficient funds are available
UPDATE wallet SET Amount = Amount - (SELECT Price FROM product WHERE Product_ID
= 9) WHERE User_id = 1;
select amount from wallet where user_id = 2;
COMMIT;

START TRANSACTION;

-- Increase the price of the product
UPDATE product SET Price = Price * 5 WHERE Product_ID = 9;

COMMIT;

COMMIT;

```

STEP	T1	T2
1.	START	
2.	R(product(9))	
3.	R(wallet(2))	
4.		START
5.		W(product(price))
6.		COMMIT

7.	W(wallet(amount))	
8.	COMMIT	