Task 1:

1.

T1	T2	T3	x: 5 y: 8
a = read(x)	b = read(y)	c = read(y)	x: 5 y: 8 a: 5 b: 8 c: 8
write(x,2*a+1)	write(x,b+3)	write(y,c+1)	x: 11 y: 9 a: 5 b: 8 c: 8
commit	commit	commit	x: 11 y: 9

2.

T1	T2	T3	x: 1 y: 2 z: 3
a = read(x)	b = read(y)	c = read(z)	x: 1 y: 2 z: 3 a: 1 b: 2 c: 3
write(x,a+1)	write(x,b+2)	write(y,c+3)	x: 2 y: 6 z: 3 a: 1 b: 2 c: 3
commit	commit	commit	x: 2 y: 6 z: 3

3.

T1	T2	x: 1 y: 2
a = read(x)		x: 1 y: 2 a: 1
	b = read(y)	x: 1 y: 2 b: 2
write(y,2*a+1)		x: 1 y: 3 a: 1
	write(x,b+3)	x: 5 y: 2 b: 2
commit		x: 1 y: 3 a: 1
	commit	x: 5 y: 3

Task 2:

1.

T1	T2	T3	x: 1 y: 2 z: 3
lock(x) a = read(x)			x: 1 a: 1
write(y,a+1)			y: 2 a: 1
	lock(y) b = read(y)		y: 2 b: 2
	write(y,b+2)		y: 4 b: 2
		lock(z) c = read(z)	z: 3 c: 3
		write(x,c+2)	x: 1
write(x,a-1)			x: 0 a: 1
unlock(x)			x: 0
commit	unlock(y)		x: 0 y: 4
	commit	unlock(z)	x: 0 y: 4 z: 3
		commit	x: 0 y: 4 z: 3

2.

T1	T2	T3	x
timestamp(t1)			
a = read(x)			x:t1
write(y,a+1)			y:t1
	timestamp(t2)		
	b = read(y)		y:t1:t2
	write(y,b+2)		
		timestamp(t3)	
		c = read(z)	z:t3
		write(x,c+2)	x:t1:t3
write(x,a-1)			x:t1:t3:t1
commit			
	commit		
		commit	
		abort	

Task 3:

1.

Transaction 1	**Transaction 2**

UPDATE ORDERS
SET freight = freight + 0.5
WHERE shipped_date < '01-SEP-2020';	

	UPDATE ORDERS
	SET freight = freight + 0.5
	WHERE shipped_date > '01-SEP-2021';

UPDATE ORDERS
SET freight = freight + 1.5
WHERE shipped_date > '01-SEP-2022';	

	Wait
COMMIT;	
	ROLLBACK;

2. Transaction 1 commits the changes to the freight column in the ORDERS table so Transaction 2 will abort and roll back its changes.

Task 4:

1.

Transaction 1	**Transaction 2**

DECLARE
  avg_freight ORDERS.freight%TYPE;
 	
BEGIN
  SELECT AVG(freight)
  INTO avg_freight
  FROM ORDERS
  WHERE order_id IN (383,384,385);
 	
  UPDATE ORDERS
  SET freight = (freight + avg_freight)/2
  WHERE order_id = 384;
  COMMIT;
END;
/	

	DECLARE
	  avg_freight ORDERS.freight%TYPE;
	
	BEGIN
	  SELECT AVG(freight)
	  INTO avg_freight
	  FROM ORDERS
	  WHERE order_id IN (381,382,383,384,385);
	
	  UPDATE ORDERS
	  SET freight = (freight + avg_freight)/2
	  WHERE order_id IN (383, 384);
	  COMMIT;
	END;
	/

2. Transaction 2 corrupts a database as Transaction 1 modifies and commits data that Transaction 2 is accessing.