

Registro de Operaciones	
Sesion 1	Tiempo
Saldo inicial para cuenta con número 4000000 (CUENTA_1) = 10000	
SET AUTOCOMMIT OFF; SET TRANSACTION ISOLATION LEVEL READ COMMITTED;	t1
UPDATE Cuentas SET saldo = saldo + 1000000 WHERE numero = '4000000';	t2
INSERT INTO Operaciones (id, tipo, monto, fechaHora, puntoDeAtencion, producto) VALUES (idOperaciones.nextval, 'consignacion_cuenta', 1000000.00, SYSDATE, 3, 1);	t3
UPDATE Cuentas SET saldo = saldo - 50000 WHERE numero = '4000001';	t4

INSERT INTO Operaciones (id, tipo, monto, fechaHora, puntoDeAtencion, producto) VALUES (idOperaciones.nextval, 'retiro_cuenta', 50000.00, SYSDATE, 3, 4);	t5
	t6
COMMIT;	t7
	t8
	t9

	t10
SELECT saldo FROM Cuentas WHERE numero IN ('4000000', '4000001');	t11
	t12
SELECT saldo FROM Cuentas WHERE numero IN ('4000000', '4000001');	t13

5
Sesion 2
Saldo inicial para cuenta con número 4000001 (CUENTA_2) = 5000
SET AUTOCOMMIT OFF; SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

```
UPDATE Cuentas SET saldo = saldo - 30000  
WHERE numero = '4000000';
```

```
INSERT INTO Operaciones (id, tipo, monto, fechaHora, puntoDeAtencion, producto)  
VALUES (idOperaciones.nextval, 'retiro_cuenta', 30000.00, SYSDATE, 3, 1);
```

```
UPDATE Cuentas SET saldo = saldo + 5000  
WHERE numero = '4000001';
```


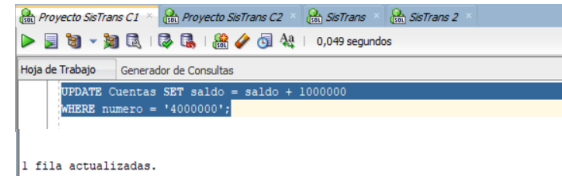
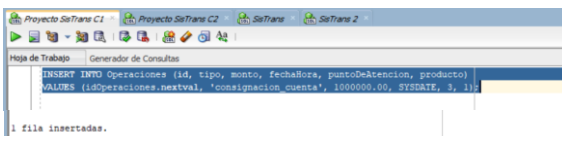
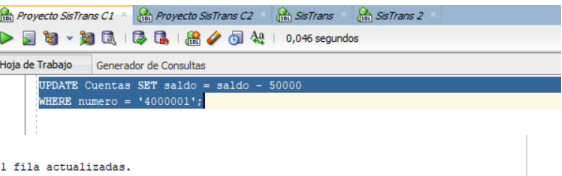
```
INSERT INTO Operaciones (id, tipo, monto, fechaHora, puntoDeAtencion, producto)
VALUES (idOperaciones.nextval, 'consignacion_cuenta', 5000.00, SYSDATE, 3, 4);
```

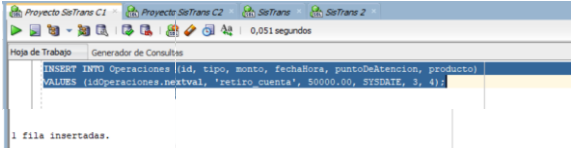
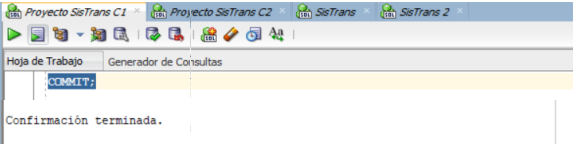
```
COMMIT;
```

```
SELECT saldo
FROM Cuentas
WHERE numero IN ('4000000', '4000001');
```

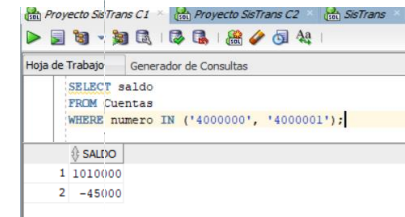
sesion 1

fotos

Inicia bloque T1	
Inicia T1. El saldo de la cuenta 1 pasa a ser 1010000	
Se registra la operación anterior de consignación a T1	
El saldo de la cuenta 2 pasa a ser -45000	

<p>Se registra la operación anterior de retiro de T1</p>	
<p>Finaliza T1</p>	

Como T2 aún no ha hecho commit, T1 sigue viendo los cambios que ya confirmó

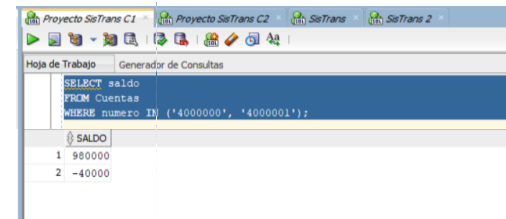


The screenshot shows a SQL query window with the following content:

```
SELECT saldo  
FROM Cuentas  
WHERE numero IN ('4000000', '4000001');
```

SALDO	
1	1010000
2	-45000

Ya después de que T2 hiciera commit, T1 se da cuenta de que los datos fueron modificados por alguna otra transacción diferente a los cambios que ya hizo



The screenshot shows a SQL query window with the following content:

```
SELECT saldo  
FROM Cuentas  
WHERE numero IN ('4000000', '4000001');
```

SALDO	
1	950000
2	-40000

Inicia bloque T2

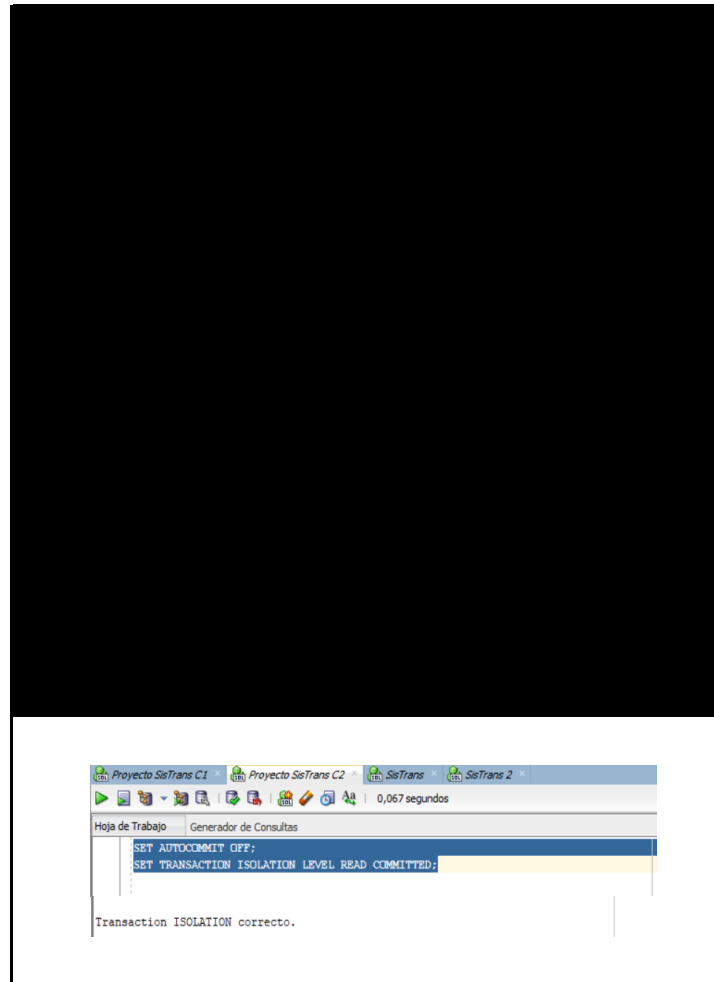
Se intenta retirar de la cuenta 1, pero debido a que T1 modificó la cuenta 1 y no ha hecho commit, queda bloqueada gracias al nivel de serialización
Después de que T1 liberará la cuenta 1 al hacer commit, T2 pudo continuar su ejecución. T2 cree que está modificando el saldo inicial de la cuenta 1 de 10000, pero realmente está modificando los cambios hechos por T1, es decir que los 30000 son descontados de 1010000 y la cuenta 1 pasa a tener un saldo de 980000
Se registra la operación anterior de retiro de T2
T2 cree que está modificando el saldo inicial de la cuenta 2 de 5000, pero realmente está modificando los cambios hechos por T1, es decir que los 5000 son adicionados a -45000 y la cuenta 2 pasa a tener un saldo de -40000

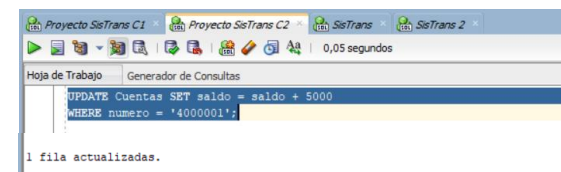
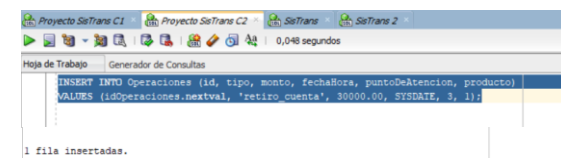
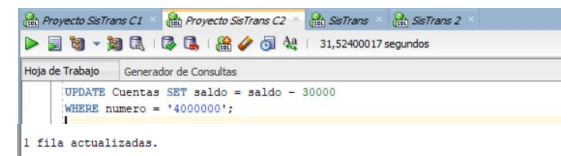
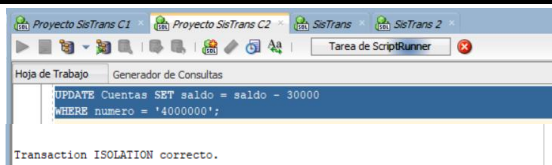
Se registra la operación anterior de consignación de T2

Finaliza T2

Acá se da cuenta de que modificó datos que no correspondían a como los tenía presentes al inicio, sacando a luz a una lectura inconsistente, las cuales son permitidas por este nivel de aislamiento

fotos





```
INSERT INTO Operaciones (id, tipo, monto, fechaHora, puntoDeAtencion, producto)
VALUES (idOperaciones.nextval, 'consignacion cuenta', 5000.00, SYSDATE, 3, 4);
```

1 fila insertadas.

```
COMMIT;
```

Confirmación terminada.

```
SELECT saldo
FROM Cuentas
WHERE numero IN ('4000000', '4000001');
```

	SALDO
1	980000
2	-40000