

--BlocksEx

```
create table empregado(  
Codigo integer,  
Nome VARCHAR(255) not null,  
Cargo VARCHAR(255) not null,  
salario integer,  
constraint PK_emp primary key(Codigo)  
);
```

```
insert into Empregado values (01, 'Joao da Silva', 'Programador', 2000.00);  
Begin;  
    update Empregado set salario = salario * 1.1 where codigo = 01;  
    update Empregado set cargo = 'Programador Senior' where codigo = 01;  
    delete Empregado where cargo = 'Programador';  
    select * from empregado;  
    commit;
```

--NONREPEATABLE READ
--CLIENTE

```
begin;  
select * from curso;  
update curso set nomeCurso = 'ADS'  
where id curso = 3;  
end;  
--Phantom Read
```

--Cliente 1

```
begin;  
insert into curso values(4,'BSI',01/01/2021);  
insert into curso values(5,'Engenharia Civil',01/01/2021);  
end;
```

```
create table conta(  
idConta integer,  
cliente VARCHAR(255) not null,  
saldo numeric(15,2) default 0 ,  
constraint PK_CONTA primary key(idConta)  
);
```

```
insert into conta values(1,'JOÃO',1000),  
                        (2,'PEDRO',15000),  
                        (3,'MARIO',450),  
                        (4,'JOAQUIM',40000);
```

--Bloco de Transação

--Exemplo 1

```
begin; --EXECUTAR BEGIN ANTES DE EXECUTAR OS OUTROS COMANDOS
INSERT INTO CONTA VALUES (5,'CARLOS',2000);
SELECT * FROM CONTA;
ROLLBACK; --DESAZ TUDO O QUE FOI FEITO DENTRO DO BLOCO
SELECT * FROM CONTA;
```

--Exemplo 2

```
begin;
update conta set saldo = 120 where idConta = 1;
savepoint sv1;
update conta set saldo = saldo -1000 where idConta = 2;
select * from conta;
rollback to sv1;
select * from conta;
end;
```

--Ex1

```
begin;
insert into conta values (6,'MARA',5000);
rollback; --ROLLBACK FUNCIONA COMO END CASO NÃO RETORNE A UM SAVE POINT
```

--TRANSAÇÃO A

```
BEGIN;
UPDATE CONTA SET CLIENTE = 'JOÃO II' WHERE idCONta = 1;
select * from conta;
COMMIT;
```

--TRANSAÇÃO B

```
BEGIN;
UPDATE conta set saldo = 350 where cliente = 'JOÃO II';
select * from conta;
end;
```