



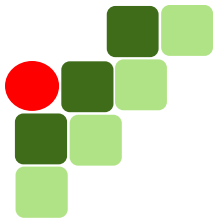
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Araraquara

Banco de Dados II

Cristiane Yaguinuma

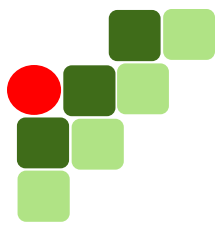
cristiane.yaguinuma@ifsp.edu.br

- **Processamento de transações em PL/SQL**



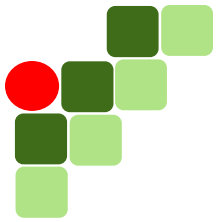
Roteiro da aula

- ▶ Comandos para transações em PL/SQL
- ▶ Exercícios



Comandos para transações em PL/SQL

- ▶ Os mesmos comandos para controle de transações em SQL
 - COMMIT
 - ROLLBACK
 - SAVEPOINT
- ▶ Uma transação pode abranger múltiplos blocos de programa em PL/SQL
- ▶ Um bloco pode conter múltiplas transações



Exemplo: sistema bancário

```
-- CONECTADO COMO SYS OU SYSTEM
```

```
CREATE USER BANKING
```

```
IDENTIFIED BY 123
```

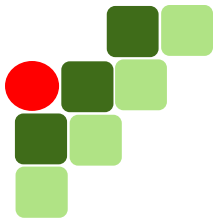
```
DEFAULT TABLESPACE USERS
```

```
QUOTA 5M ON USERS
```

```
TEMPORARY TABLESPACE TEMP;
```

```
-- CONCEDER PRIVILÉGIOS DE SISTEMA
```

```
GRANT CREATE SESSION, CREATE TABLE, CREATE PROCEDURE  
TO BANKING;
```



Exemplo: sistema bancário

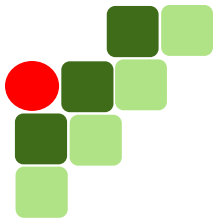
```
-- CONECTADO COMO BANKING
```

```
CREATE TABLE accounts (  
    account_id NUMBER(4),  
    balance NUMBER (10,2)  
);
```

```
INSERT INTO accounts (account_id, balance)  
VALUES (7777, 500.00);
```

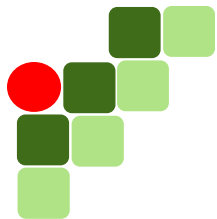
```
INSERT INTO accounts (account_id, balance)  
VALUES (8888, 100.00);
```

```
COMMIT;
```



Exemplo: sistema bancário

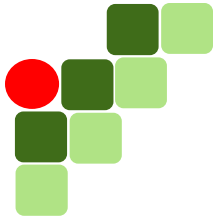
```
CREATE OR REPLACE PROCEDURE transfer (from_acct NUMBER,  
    to_acct NUMBER, amount NUMBER) AS  
BEGIN  
    COMMIT; -- finaliza qualquer transação anterior  
    UPDATE accounts  
    SET balance = balance - amount  
    WHERE account_id = from_acct;  
  
    UPDATE accounts  
    SET balance = balance + amount  
    WHERE account_id = to_acct;  
    COMMIT;  
END;  
/
```



ROLLBACK

- ▶ Geralmente usado quando ocorrem exceções
 - Restaurar o estado do BD anterior à transação

```
CREATE OR REPLACE PROCEDURE transfer (from_acct NUMBER,  
    to_acct NUMBER, amount NUMBER) IS  
    invalid_from_acct EXCEPTION;  
  
BEGIN  
  
    COMMIT; -- finaliza qualquer transação anterior  
  
    UPDATE accounts  
    SET balance = balance - amount  
    WHERE account_id = from_acct;  
  
    IF SQL%NOTFOUND THEN  
        RAISE invalid_from_acct;  
    END IF;
```



ROLLBACK

```
UPDATE accounts
SET balance = balance + amount
WHERE account_id = to_acct;
```

```
COMMIT;
```

```
EXCEPTION
```

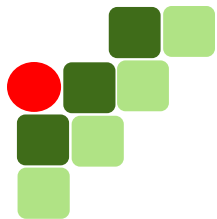
```
WHEN invalid_from_acct THEN
```

```
ROLLBACK;
```

```
DBMS_OUTPUT.PUT_LINE('Invalid origin account.');
```

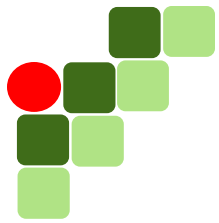
```
END;
```

```
/
```

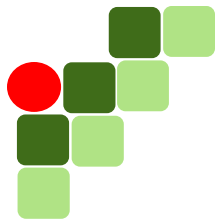
ROLLBACK e SAVEPOINT

- ▶ Quando um comando falha (por erro ou exceção), é realizado um ROLLBACK para um SAVEPOINT implícito anterior ao comando que falhou
 - Não ocorre ROLLBACK para a transação completa
- ▶ Se um bloco de programa PL/SQL termina com uma exceção que não foi tratada, não ocorre ROLLBACK implícito da transação
 - ROLLBACK tem que ser chamado explicitamente



Exercício

- ▶ **Estenda o procedimento transfer para tratar as seguintes exceções considerando gerenciamento de transação:**
 - **Exceção quando a conta de destino for incorreta**
 - **Exceção quando o valor a ser transferido é maior que o saldo na conta de origem**



Exercício

- ▶ Estenda o procedimento transfer para registrar em uma tabela de log todas as tentativas de transferência
 - Usar tratamento de exceções e transações
 - Criar tabela: transfer_log (from_acct NUMBER, to_acct NUMBER, amount NUMBER, status INT, desc VARCHAR2)
 - Status = 1 se transferência com sucesso, status = 0 caso contrário
 - Desc: descrição do problema que impediu a transferência
 - Número de conta inválida (from_acct ou to_acct)
 - Saldo insuficiente em conta



Referências

- ▶ Material e exemplos extraídos de:
 - Oracle Database PL/SQL Language Reference
 - Transaction Processing and Control