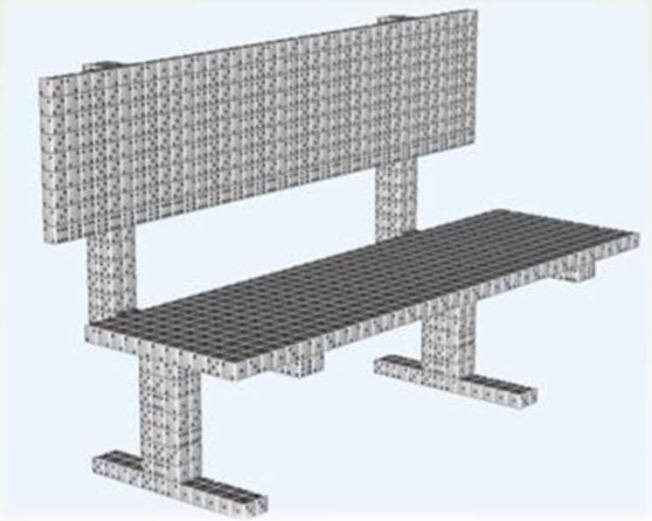

Banco de Dados II
- Gerenciamento de transações

Prof. Angelo Augusto Frozza, Dr.

http://about.me/TilFrozza

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Roteiro

- Transações na prática

INSTITUTO FEDERAL Catarinense Campus Camboriú

Transações em SQL

- Por *default*, todo comando individual é considerado uma transação
  - ✓ exemplo: DELETE FROM Pacientes
    - ➨ exclui todas ou não exclui nenhuma *tupla* de *Pacientes*, deve manter o BD consistente etc.

- Comandos SQL mais comuns para transações:
  - **START TRANSACTION**
    - ➨ inicia e configura características de uma transação
  - **COMMIT**
    - ➨ encerra a transação (solicita efetivação das suas ações)
  - **ROLLBACK**
    - ➨ solicita que as ações da transação sejam desfeitas

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Transações em SQL

- Principais configurações (**START TRANSACTION**)
  - modo de acesso
    - READ ONLY (somente leitura), ou
    - READ WRITE (leitura e escrita - *default*)

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Transações em SQL

- Principais configurações (**START TRANSACTION**)
  - nível de isolamento
    - indicado pela cláusula ISOLATION LEVEL *nível*
    - *nível* para uma transação T_i pode assumir (várias configurações possíveis):
      - SERIALIZABLE (T_i executa com completo isolamento - *default*)
      - ...
      - READ COMMITTED (T_i só lê dados efetivados, mas outras transações podem escrever em dados lidos por T_i)
        - aceitável, por exemplo, em casos em que transações que leem dados não irão escrever esses dados posteriormente

Transações em SQL

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Exemplo (SQL embutida)

```
EXEC SQL START TRANSACTION
                ISOLATION LEVEL SERIALIZABLE
                READ WRITE;
...
for(;;)
{ ...
  EXEC SQL INSERT INTO Empregados
           VALUES (:ID,:nome,:salario)
  ...
  EXEC SQL UPDATE Empregados
           SET salário = salário + 100.00
           WHERE ID = :cod_emp
  if (SQLCA.SQLCODE <= 0) EXEC SQL ROLLBACK;
  ...
}
EXEC SQL COMMIT;
...
```

Transações em SQL

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Adotado por alguns SGBDs (*script* SQL)

```
BEGIN TRANSACTION T1

UPDATE Medicos
    SET nroa = NULL
    WHERE nroa = @nroAmb

IF @@ERROR <> 0 ROLLBACK TRANSACTION T1

DELETE FROM Ambulatorios
    WHERE nroa = @nroAmb

IF @@ERROR <> 0 ROLLBACK TRANSACTION T1
    ELSE COMMIT TRANSACTION T1
...
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Transações no PostgreSQL

- No *PostgreSQL* existem dois comandos para iniciar e finalizar uma transação:

```
BEGIN     -- iniciar
          -- comandos
COMMIT    -- comitar/confirmar
ROLLBACK  -- parar/cancelar
END       -- mesma função do COMMIT
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Transações no PostgreSQL

- **Transação implícita**

```
--BEGIN (implícito)
   UPDATE conta SET saldo = 100 WHERE id = 1;
--COMMIT (implícito)
```

- Qualquer comando SQL executado isoladamente é tratado dentro de uma transação, mesmo se o usuário não colocar os comandos **begin … commit/rollback.**

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Transações no PostgreSQL

- Para voltar a pontos anteriores depois de um **ROLLBACK**, o *PostgreSQL* trabalha com o conceito de ***Snapshots***
  - "***fotos***" do momento atual do banco antes de iniciar a transação;
  - Esta *foto* que será confirmada ("*comitada*") se o *snapshot* não conflitar com dados de outras transações.

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Transações no PostgreSQL

- Considere o exemplo de uma operação bancária:

```
create table teste.conta(
  id integer primary key,
  cliente varchar(30) not null,
  saldo numeric(15,2) default 0
)
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Transações no PostgreSQL

- Considere o exemplo de uma operação bancária:

```
create table teste.conta(
    id integer primary key,
    cliente varchar(30) not null,
    saldo numeric(15,2) default 0
);
```

- Insira registros:

```
(1,'Osmar Dito',1000);
(2,'P. Lucia',15000);
(3,'Oscar A Melo',450);
(4,'G. Ladeira',40000);
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Transações no PostgreSQL

- Considere as segintes operações que formam uma única transação:

```
select * from teste.conta where id = 1;
-- transação
begin;
 update teste.conta set saldo = 120 where id = 1;
rollback;
select * from teste.conta where id = 1;
```

- Na primeira linha são mostrados os dados da consulta antes de iniciar a transação.
  - Em seguida inicia-se uma transação e atualiza-se o saldo da conta “1”.
- Depois executa-se um *rollback*.
  - O resultado das execuções acima não tem efeito nos dados do SGBD, isso porque o *rollback* anulou todas as alterações e o resultado do primeiro SELECT será igual ao resultado do último SELECT.

Transações no PostgreSQL

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Outro recurso muito interessante do *PostgreSQL* é o "***SavePoint***", que possibilita que salvar determinado ponto dentro de uma transação e voltar para ele, quando achar necessário.

```
begin;
   update teste.conta set saldo = 120 where id =
1;
   savepoint savepoint_1;
   update teste.conta set saldo = saldo - 1000
       where id = 2;
   select * from teste.conta where id = 2;
   rollback to savepoint_1;
   select * from teste.conta where id = 2;
commit;
```

Contato

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Prof. Angelo Augusto Frozza, Dr.

angelo.frozza@ifc.edu.br

http://www.ifc-camboriu.edu.br/~frozza

@TilFrozza

http://www.twitter.com/TilFrozza

http://about.me/TilFrozza