UNIVERSIDADE FEDERAL DE ITAJUBÁ

Banco de Dados II

COM 231

Transações

Vanessa Cristina Oliveira de Souza

Transação

"Transação é uma unidade lógica de trabalho, envolvendo diversas operações de bancos dados."

(C. J. Date - Introdução a Sistemas de Bancos de Dados

Transação

- Está envolvida com os comandos Update, Delete e Insert.

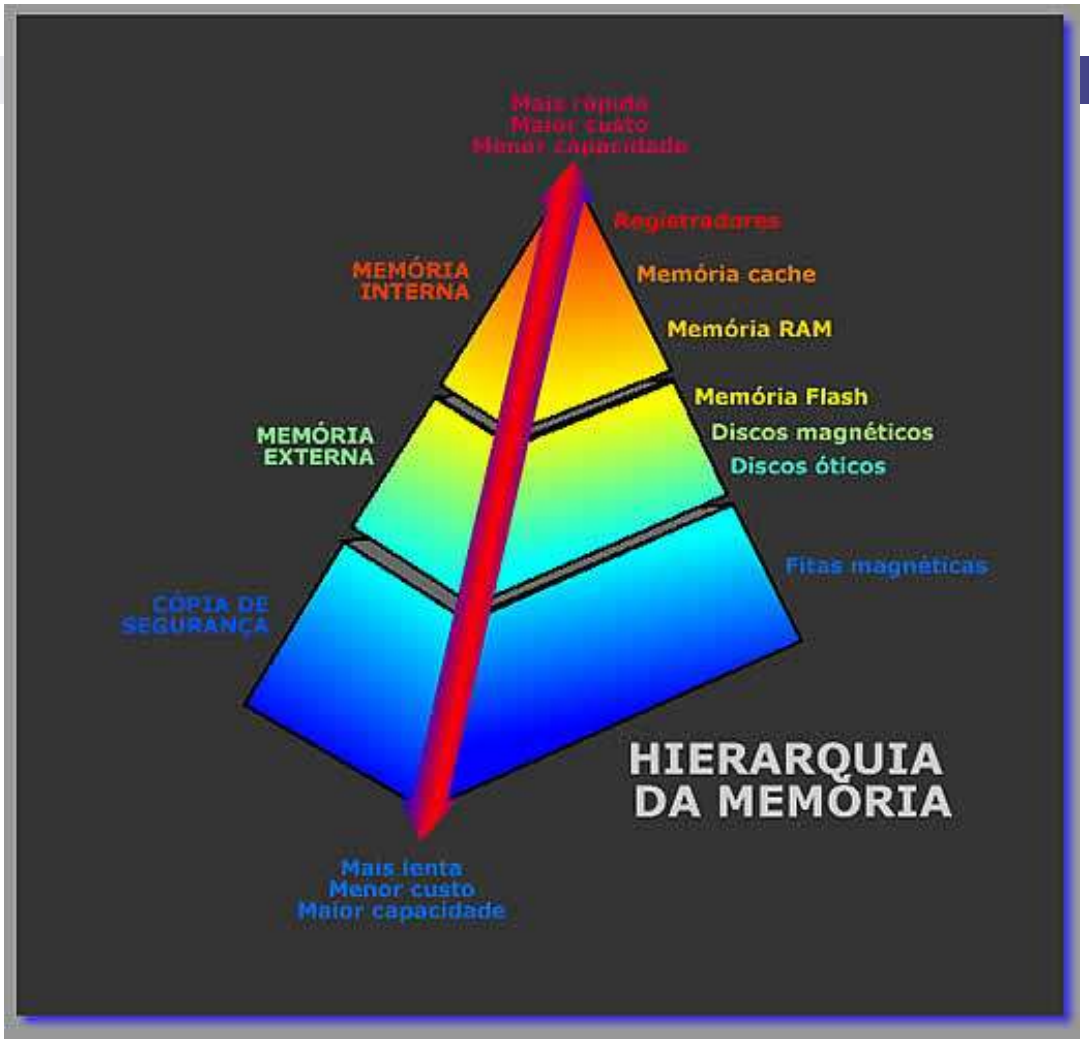
- Do ponto de vista do SGBD, uma transação é uma sequência de operações que são tratadas como um bloco único e indivisível (atômico) no que se refere à sua recuperação.

- Assegura que atualizações inacabadas ou atividades corrompidas não sejam executadas no banco de dados.

Tipos de Armazenamento

- Armazenamento volátil
  - Memória principal e cache
- Armazenamento não-volátil
  - Discos e fitas
- Armazenamento estável
  - Duplicar informações em diversos meios não-voláteis, com modos independentes de falhas, e atualizar a informação de uma maneira controlada.

Hierarquia de Armazenamento



Fonte: http://www.bpiropo.com.br/fpc20070903.htm

Armazenamento estável

- Informação armazenada em **RAM** é perdida se faltar energia ou se a máquina falhar.
- Informação armazenada em **disco** é perdida se a cabeça do disco falhar.
- Informação em **armazenamento estável** sobrevive a *tudo*, exceto enchentes, terremotos, ...

Hierarquia de Armazenamento

- As transações transferem blocos de informações do disco para a memória principal e depois os devolvem para o disco.

- Os blocos residentes no disco são chamados blocos físicos, e os residentes na memória, de blocos de *buffer*.

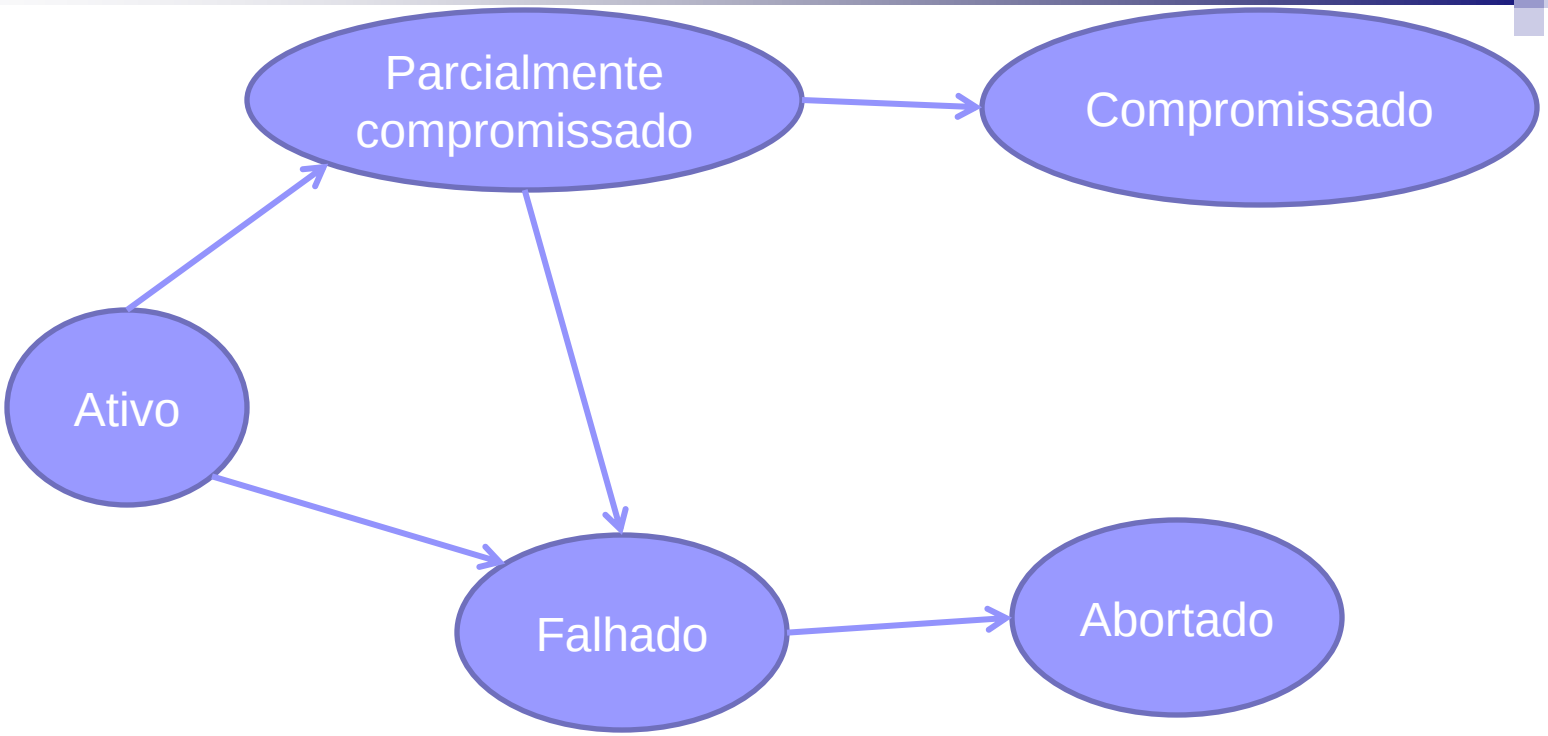
Hierarquia de Armazenamento

- As transações interagem com o sistema de banco de dados transferindo dados de variáveis de programa para o banco de dados e, do banco de dados para as variáveis de programa.

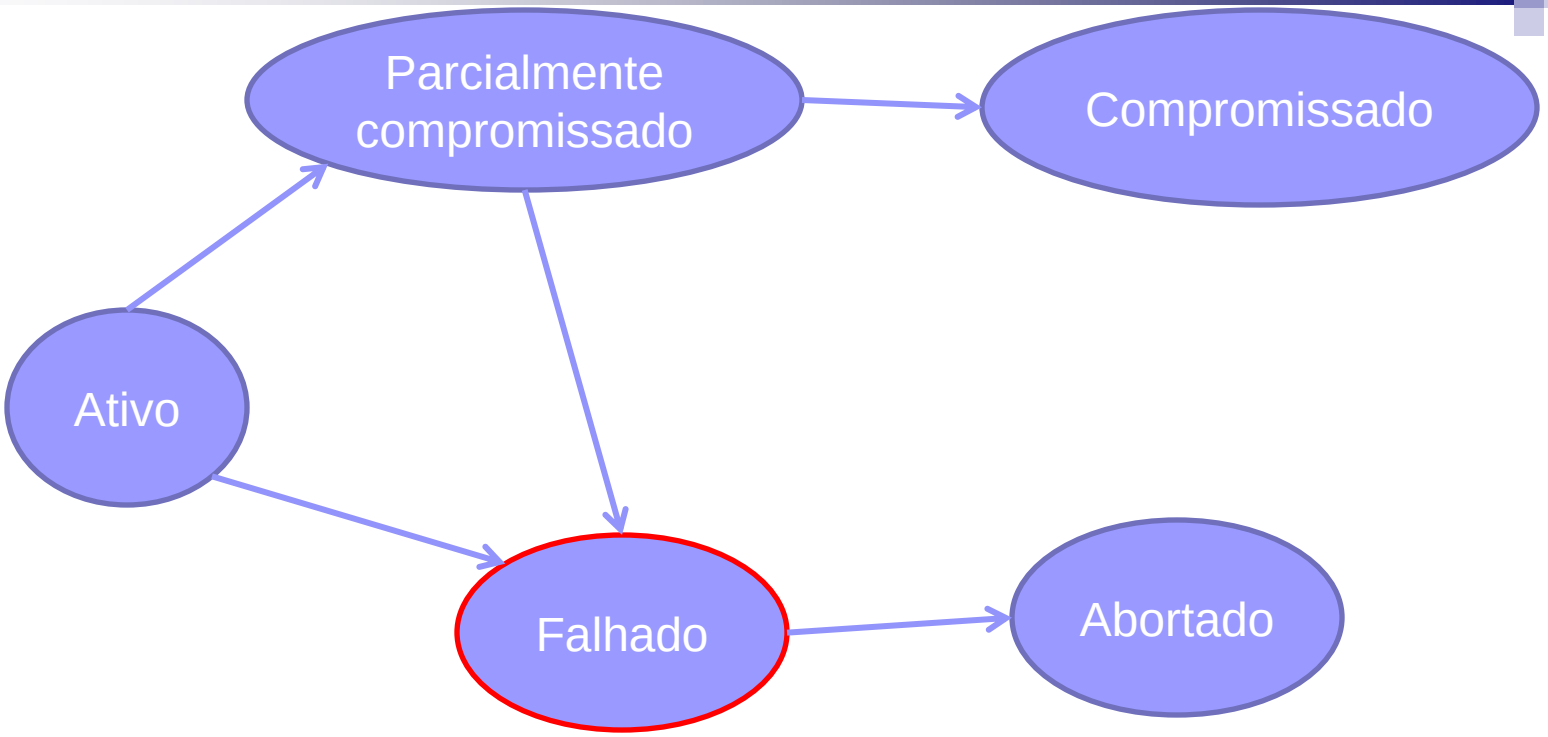
Transação ACID

- A integridade de uma transação depende de 4 propriedades, conhecidas como ACID:
  - Atomicidade
  - Consistência
  - Isolamento
  - Durabilidade

Estados de Transações



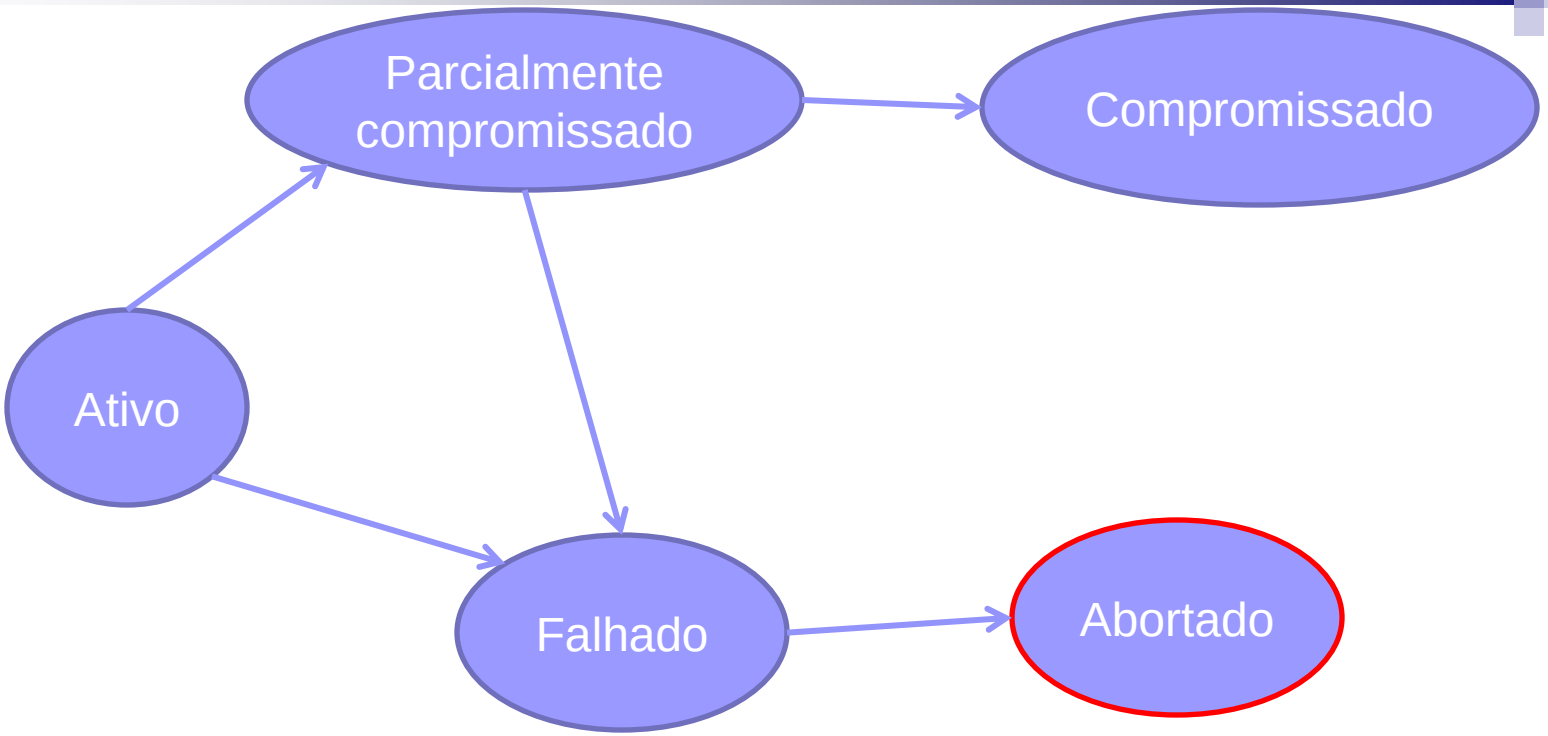
Estados de Transações



Estado Falhado

- Uma transação entra no estado falhado depois de ser determinado que a transação não pode mais prosseguir com sua execução normal (por exemplo, devido a erros de hardware ou a erros lógicos).
- Tal transação precisa ser desfeita.
- Uma vez que isso é conseguido, a transação entra no estado abortado.

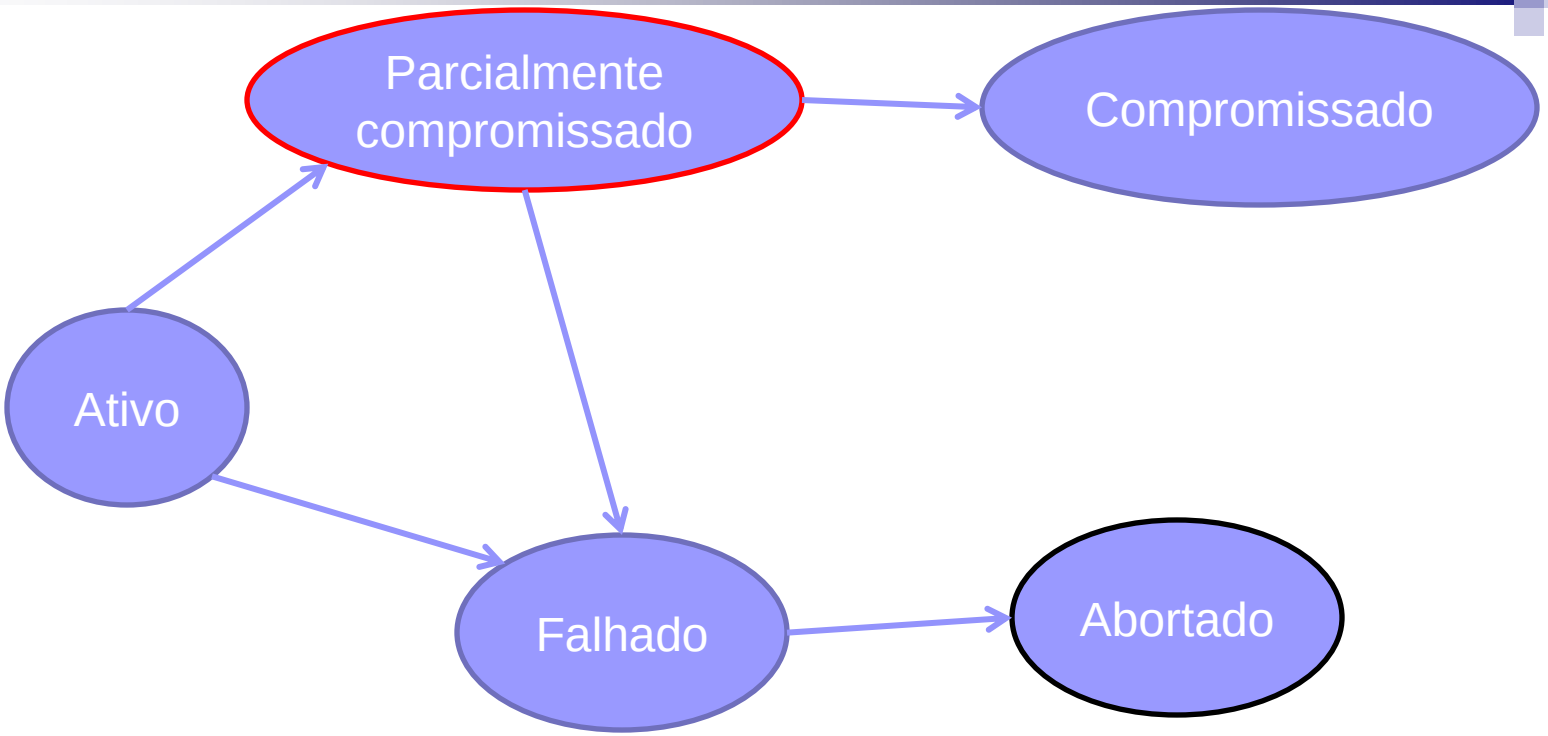
Estados de Transações



Estado Abortado

- Nesse ponto, o sistema tem duas opções:
  - Reiniciar a transação
    - Apenas se a falha foi causada por um erro foi de hardware ou de software
    - Uma transação reiniciada é uma nova transação
  - Matar a transação
    - Normalmente, quando a falha foi causada por um erro de lógica

Estados de Transações

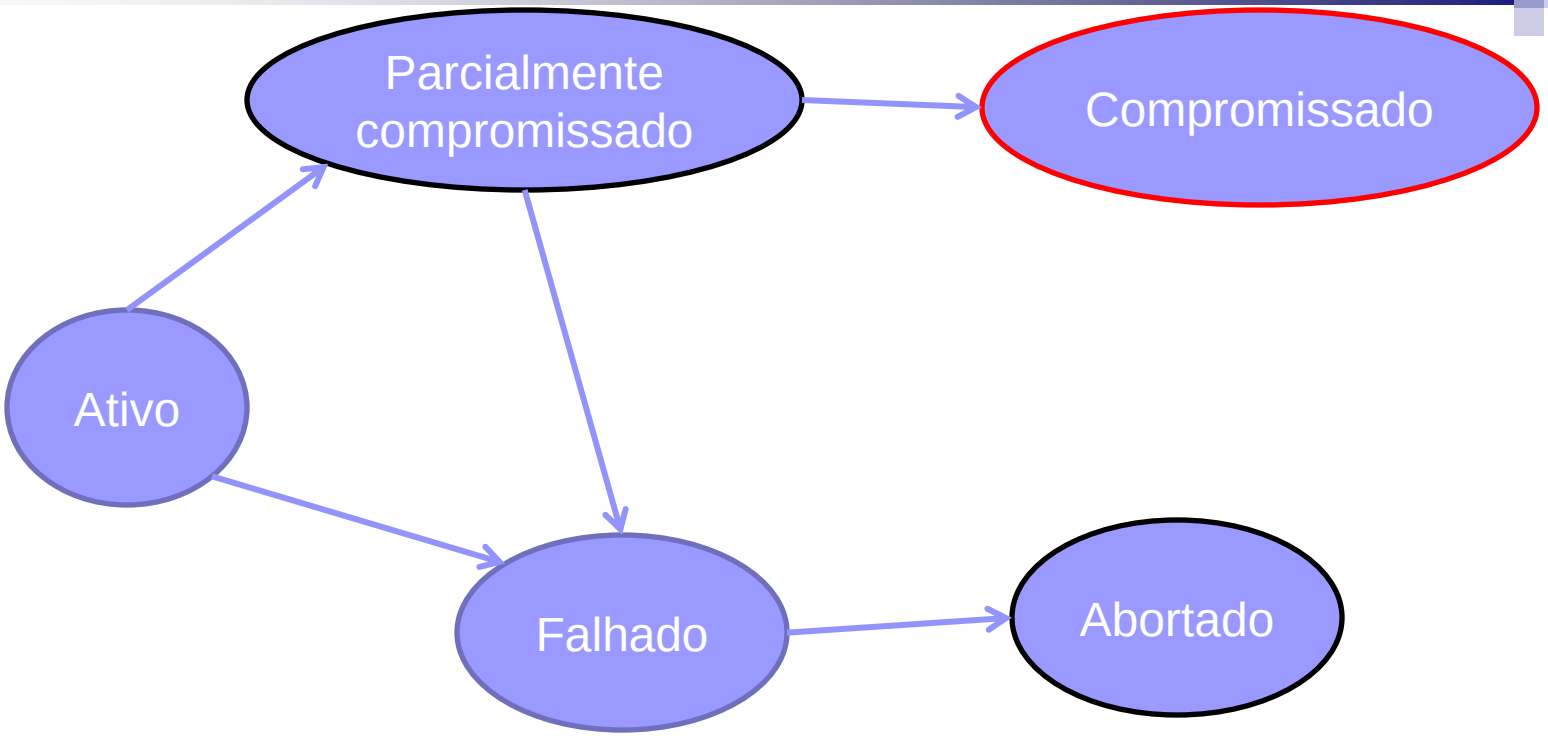


Estado parcialmente compromissado

- Uma transação entra no estado parcialmente compromissado quando alcança sua última instrução.

- Nesse ponto, a transação completou sua execução, mas ainda é possível que ela venha a ser abortada.

Estados de Transações



Estado Compromissado

- ‘Commited’

- Uma transação entra no estado compromissado se já estiver parcialmente compromissada e for garantido que nunca será abortada, garantindo sua atomicidade e durabilidade.

Commit

- Operação de confirmação de que correu tudo bem e que todos os comandos que fazem parte da transação foram executados com sucesso e o banco de dados encontra-se em um estado consistente;
- Salvar a(s) operação(ões) realizadas até o momento;
- As alterações feitas na transação atual tornam-se permanentes e visíveis a outros usuários.

Transações Implícitas

- São aquelas que têm um COMMIT interno e já tornam quaisquer modificações nos dados permanentes no(s) arquivo(s) de dados em disco.

Transações Explícitas

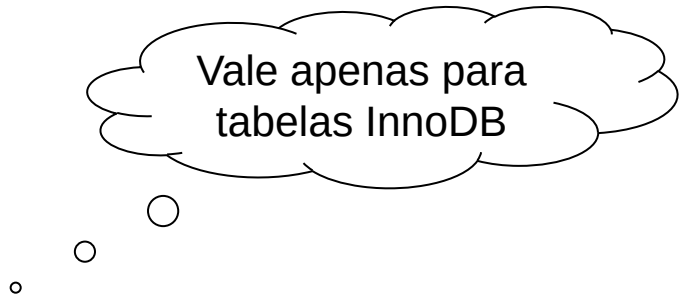
- São aquelas que têm um início e fim explicitado pelo desenvolvedor.

Autocommit

- Por padrão, os SGBDs executam em modo autocommit.
  - Variável de Ambiente
  - No psql -> \echo :AUTOCOMMIT
- Isto significa que assim que você executa uma instrução que modifica uma tabela, a mesma é feita no disco.

Autocommit

- Autocommit = on;
  - as transações implícitas sempre terão um COMMIT interno.



- Autocommit = off;
  - o SGBD esperará por um COMMIT explícito para tornar permanente as últimas manipulações nos dados.

Rollback

- Desfaz as operações até o início da transação, caso tenha havido problema com algum comando dentro de uma transação;
- A transação é finalizada sem sucesso.

Autocommit

- \set AUTOCOMMIT off
  - Insert into departamento VALUES (2, “Recursos Humanos”, 1, ‘2007-01-01’);
  - Select * from departamento;
  - Rollback;
  - Select * from departamento;
- \set AUTOCOMMIT on
  - Insert into departamento VALUES (3, “TI”, 6, ‘2008-01-01’);
  - Select * from departamento;
  - Rollback;
  - Select * from departamento;

START TRANSACTION

- Desabilitar o modo autocommit para uma única série de instruções (autocommit = off);
- O autocommit volta a estar habilitado quando se termina a transação;

START TRANSACTION ou BEGIN

- Inicia uma transação multi-instrução;

START TRANSACTION;

Comando SQL;

Comando SQL;

Comando SQL;

...

COMMIT;

START TRANSACTION ou BEGIN

START TRANSACTION;

INSERT INTO departamento VALUES (3, “TI”, 6, ‘2008-01-01’);

SELECT * FROM departamento;

UPDATE empregado SET salario = (salario+(salario*0.3));

SELECT * FROM empregado;

ROLLBACK;

SELECT * FROM departamento;

SELECT * FROM empregado;

START TRANSACTION ou BEGIN

START TRANSACTION;

INSERT INTO departamento VALUES (3, “TI”, 6, ‘2008-01-01’);

SELECT * FROM departamento;

UPDATE empregado SET salario = (salario+(salario*0.3));

SELECT * FROM empregado;

COMMIT;

SELECT * FROM departamento;

SELECT * FROM empregado;

Triggers e Transações

```
-- QUESTÃO 3
CREATE TRIGGER atualizaEstoque BEFORE INSERT OR UPDATE ON vendas.ItensVenda
    FOR EACH ROW EXECUTE PROCEDURE validaQuantidade();
--
CREATE FUNCTION validaQuantidade() RETURNS trigger AS $emp_stamp$
    DECLARE
        estoque numeric;
    BEGIN
        estoque = (SELECT quantEstoque FROM vendas.Produtos AS pr WHERE pr.codigoProduto = new.codigoProduto);
        estoque = estoque - new.quantidade;
        IF (estoque >= 0) THEN
            UPDATE vendas.Produtos SET quantEstoque = estoque WHERE codigoProduto = new.codigoProduto;
        ELSE
            RAISE EXCEPTION '% Quantidade de estoque insuficiente', NEW.codigoProduto;
        END IF;
        RETURN NULL;
    END;
$emp_stamp$ LANGUAGE plpgsql;
```

SAVEPOINT

- Permite criar "pontos de salvamento" em meio a uma transação, possibilitando que esta seja recuperada até determinado ponto.

- Pode ser feito em transações implícitas e em meio a uma transação explícita, dentro de procedimentos, independente do valor de AUTOCOMMIT.

SAVEPOINT

Bloco Transacional

```
START TRANSACTION;
INSERT INTO cursos VALUES (5, 'EEL', 6, 2);
SAVEPOINT p1;
UPDATE cursos SET coordenador = 1 WHERE codigo=5;
ROLLBACK TO SAVEPOINT p1;
UPDATE cursos SET coordenador = 1 WHERE codigo=5;
COMMIT;
```

Rollback não salva tudo

- O rollback não desfaz comandos da DDL, como create table, create database, alter table...

- Esses comandos devem ser evitados dentro de transações

Exercício

- Verificar a saída de cada select no código abaixo:

```
START TRANSACTION;
INSERT INTO northwind.categories
    VALUES(9, 'Higiene', 'Higiene e Perfumaria');
SELECT * FROM northwind.categories;
ROLLBACK;
SELECT * FROM northwind.categories;
```

Exercício

- Verificar a saída de cada select no código abaixo:

```
START TRANSACTION;
INSERT INTO northwind.categories
    VALUES(9, 'Higiene', 'Higiene e Perfumaria');
SELECT * FROM northwind.categories;
COMMIT;
SELECT * FROM northwind.categories;
```

Exercício

- Verificar a saída de cada select no código abaixo:

```
START TRANSACTION;
SAVEPOINT P1;
DELETE FROM northwind.categories WHERE categoryid < 5;
SELECT * FROM northwind.categories;
ROLLBACK TO SAVEPOINT P1;
SELECT * FROM northwind.categories;
DELETE FROM northwind.categories WHERE categoryid = 9;
SELECT * FROM northwind.categories;
COMMIT;
SELECT * FROM northwind.categories;
```

Para Casa

- Livro : Sistemas de Banco de Dados – Projeto, Implementação e Administração
  - Peter Rob e Carlos Coronel
  - 8ª Edição
  - Cengage Learning

- Estudar o Capítulo 10, até o tema ‘Gerenciamento de Transações SQL’