



Laboratório de Bases de Dados

Prof. José Fernando Rodrigues Júnior

Aula 4 – Transações, Visões, e Visões Materializadas

Material original editado: Profa. Elaine Parros Machado de Sousa



Transações - Revisão

- **Transação**: Unidade lógica de trabalho
 - abrange um conjunto de operações de manipulação de dados que executam uma única tarefa

Conecta ao Banco de Dados

Começa transação

Operações de consulta/atualização

...

Finaliza transação

Começa transação

Operações de consulta/atualização

...

Finaliza transação

Desconecta



Transações – Propriedades ACID

- Atomacidade
- Consistência
- Isolamento
- Durabilidade



Transações – Propriedades ACID

- **Atomacidade:** todas as operações de uma transação devem ser efetivadas; ou, na ocorrência de uma falha, nada deve ser efetivado
 - “tudo ou nada” – não se admite parte de uma operação
- **Consistência:** transações preservam a consistência da base
 - Estado inicial consistente \Rightarrow Estado final consistente
- **Isolamento:** a maneira como várias transações em paralelo interagem (o que pode ser lido e o que pode ser escrito por cada uma) deve ser bem definido
- **Durabilidade:** uma vez consolidada (committed) a transação, suas alterações permanecem no banco até que outras transações aconteçam

Transações – Propriedades ACID

- **Atomicidade:** todas as operações de uma transação devem ser efetivadas; ou, nenhuma deve ser efetivado
 - “tudo ou nada” – não se admite parte de uma operação
- **Consistência:** transações preservam a consistência da base
 - Estado inicial consistente \Rightarrow Estado final consistente
- **Isolamento:** a maneira como várias transações em paralelo interagem (o que pode ser lido e o que pode ser escrito por cada uma) deve ser bem definido
- **Durabilidade:** uma vez consolidada (committed) a transação, suas alterações permanecem permanentes, mesmo que outras transações aconteçam

Recuperação de falhas
via log

Recuperação de falhas
via log

Transações – Propriedades ACID

- **Atomicidade:** todas as operações de uma transação devem ser efetivadas; ou, nenhuma deve ser efetivado
 - “tudo ou nada” – não se admite parte de uma operação
- **Consistência:** transações não devem deixar a base de dados em estado inconsistente
 - Estado inicial consistente
- **Isolamento:** a maneira como várias transações em paralelo interagem (o que cada uma pode ler e escrever) deve ser controlado de modo que cada uma possa ser escrita por uma transação sem que haja interferência com as outras
- **Durabilidade:** uma vez consolidada (committed) a transação, suas alterações devem permanecer permanentes, mesmo que outras transações aconteçam

Recuperação de falhas
via log

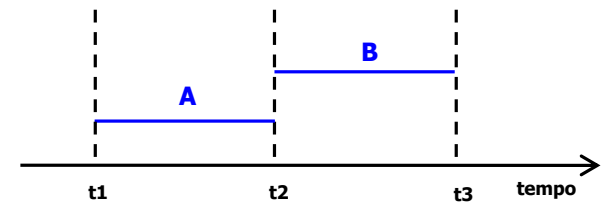
Controle de Concorrência
via Locks

Controle de Concorrência
via Locks

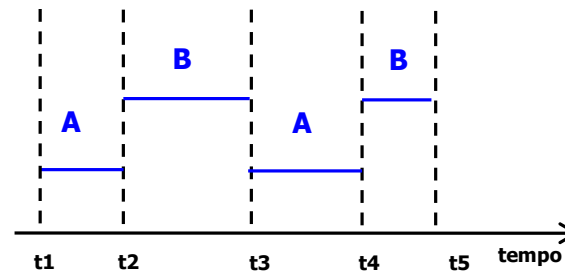
Recuperação de falhas
via log

Controle de Concorrência

- **Execução Serial (sequencial):**



- **Execução Intercalada:**





Controle de Concorrência

- **Execução Serial (sequencial):** diversas transações executadas em sequência
 - deixa a base de dados em estado correto e consistente
- **Execução Intercalada:** comandos de diversas transações são intercalados
 - pode levar a inconsistências

	Isolamento	Concorrência	Chance de inconsistências
Serial	↑	↓	↓
Intercalada	↓	↑	↑



Execução Serial X Intercalada

- **Execução serial**

- estado inicial correto e consistente \Rightarrow estado final correto e consistente



Execução Serial X Intercalada

- **Execução Intercalada**

- Toda execução serial é consistente
- Mas uma execução intercalada só é consistente se for igual ao resultado de uma execução em sequência (em ordem conhecida)
 - esta execução é dita **serializável**



Problemas de Execução Intercalada

- Ocorrência de anomalias
 1. leitura inválida
 2. leitura não repetível
 3. leitura fantasma



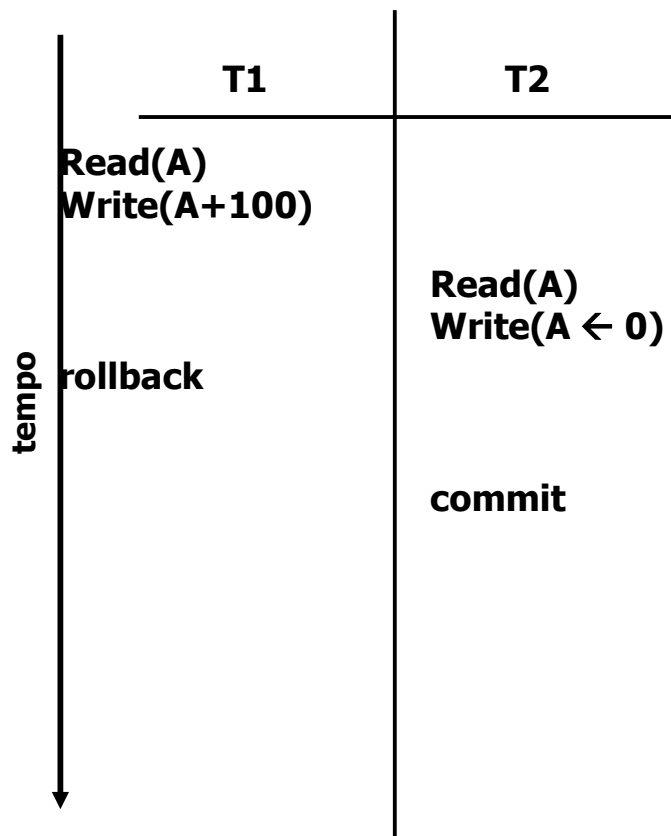
Problemas de Execução Intercalada

■ 1) Leitura inválida (*Dirty Read*):

- transação T' lê um dado modificado por uma transação T que ainda não terminou;
- permite que outras transações possam ver os dados que ainda não foram consolidados (committed), isto é, mudanças que podem ser descartadas em seguida, por causa de uma instrução ROLLBACK por exemplo.

Problemas de Execução Intercalada

■ Ex: Leitura inválida (*Dirty Read*):



Exemplo 1:

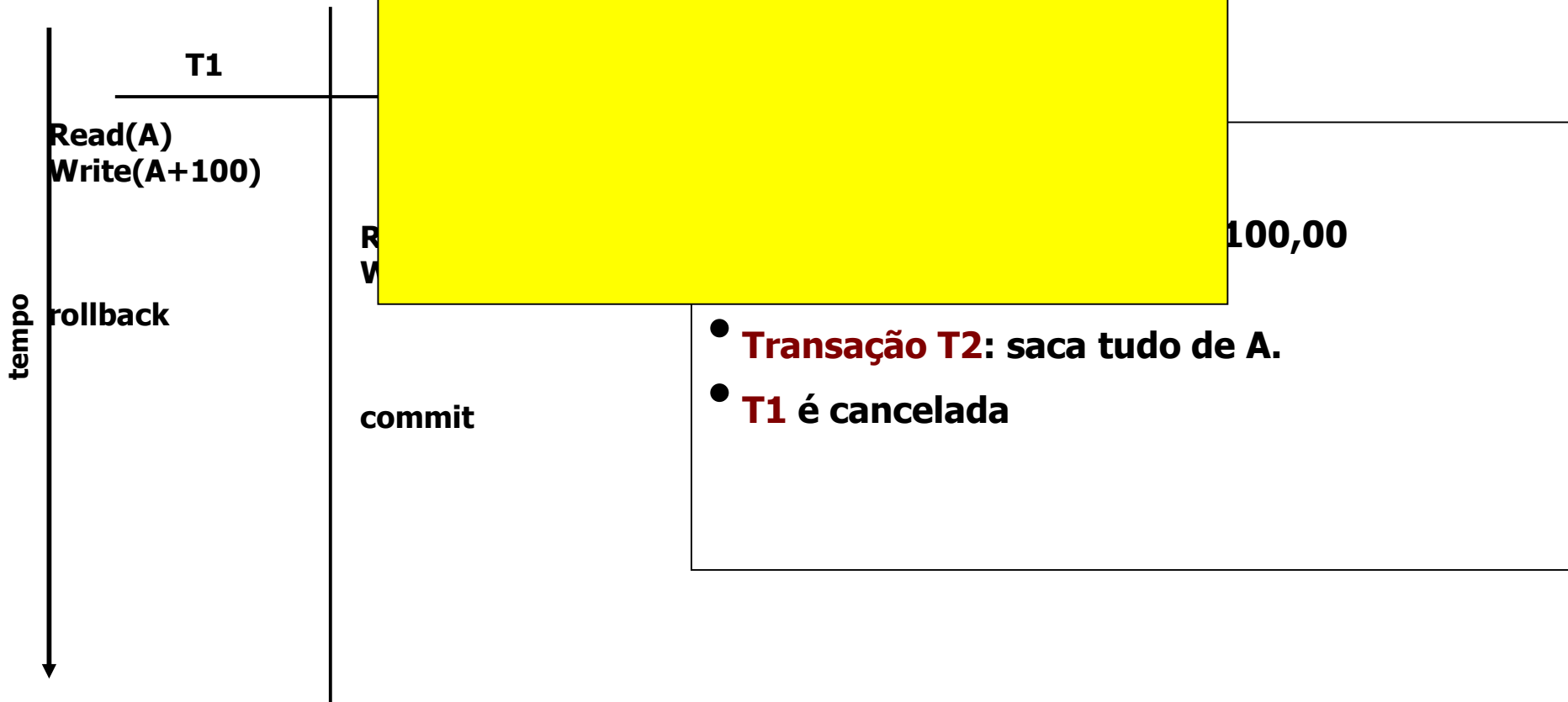
- **Transação T1:** deposita R\$100,00 na conta A.
- **Transação T2:** saca tudo de A.
- **T1** é cancelada

Problemas de Execução

Interpretação

■ Ex: Lei

Resultado: foi possível sacar R\$ 100,00 a mais.



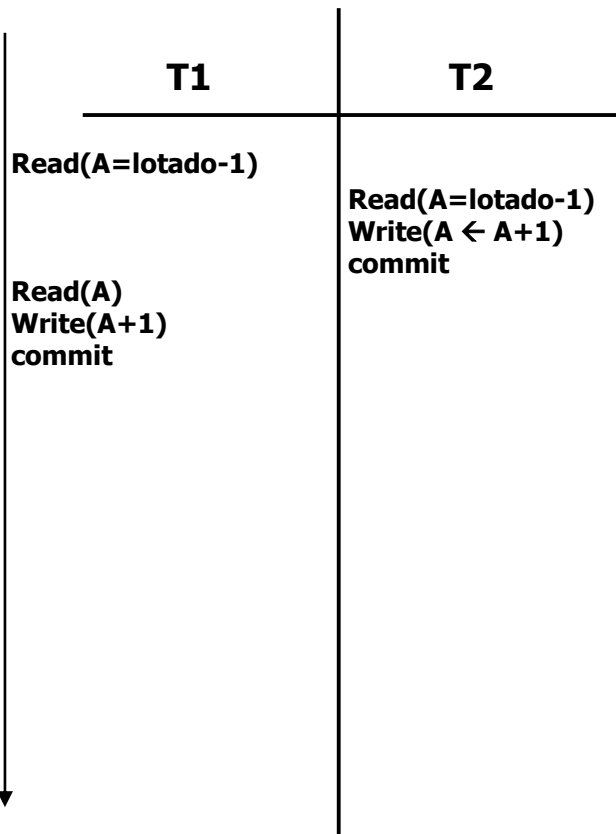


Problemas de Execução Intercalada

- **2) Leitura não repetível (*Nonrepeatable Read*):**
 - transação T lê um dado
 - esse dado é modificado por uma transação T' que começou depois de T
 - T é efetivada
 - se T' tentar reler o mesmo dado, obterá valores diferentes (*nonrepeatable read*)

Problemas de Execução Intercalada

■ Ex: Leitura não repetível (*Nonrepeatable Read*):

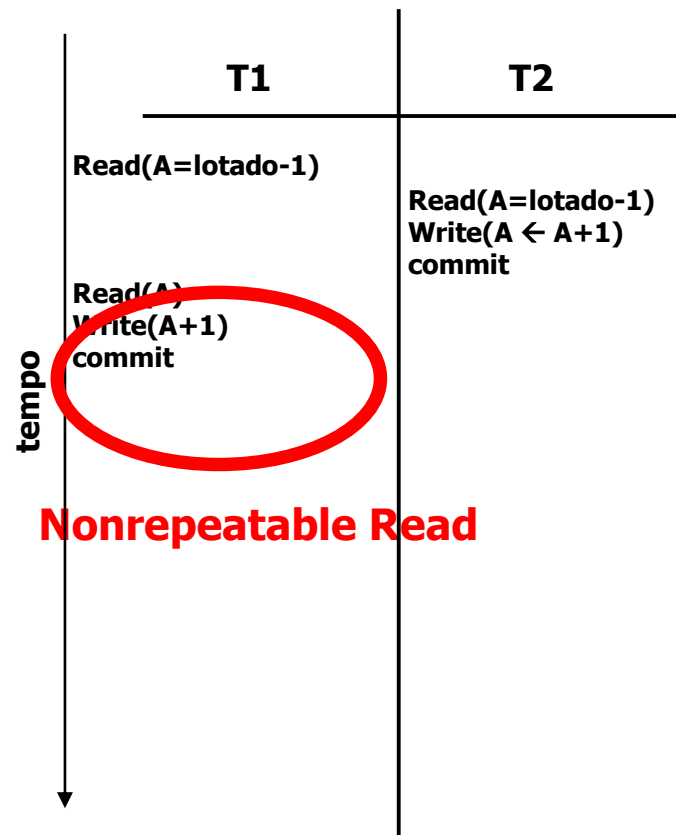


Exemplo:

- **Transação T1:** lê reservas de um voo e verifica que há apenas um lugar disponível.
- **Transação T2:** lê a mesma coisa
- **T2** reserva o último lugar e é efetivada.
- **T2** tenta reservar o lugar e ocorre um erro.

Problemas de Execução Intercalada

■ Ex: Leitura não repetível (*Nonrepeatable Read*):



Exemplo:

- **Transação T1:** lê reservas de um voo e verifica que há apenas um lugar disponível.
- **Transação T2:** lê a mesma coisa
- **T2** reserva o último lugar e é efetivada.
- **T1** tenta reservar o lugar e ocorre um erro.

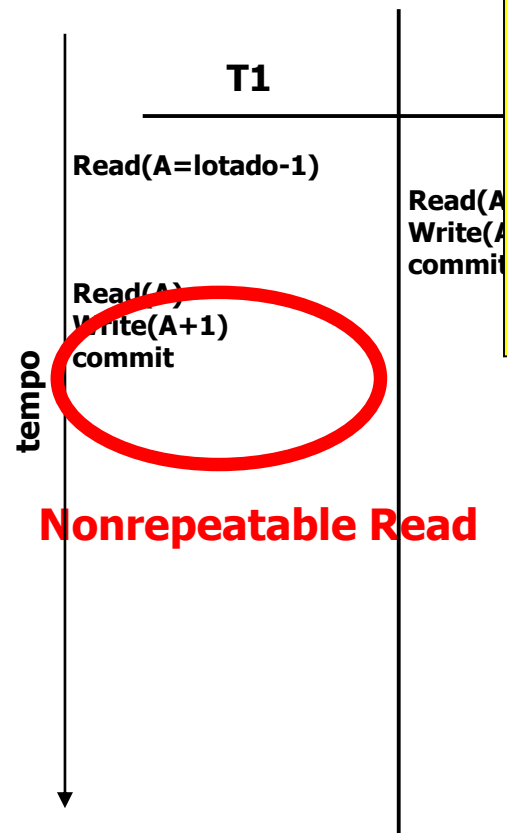
Problemas de Execução

Interacção 1

■ Ex: Leitura

Resultado: um usuário foi informado de que ainda havia lugares. Após preencher o cadastro, clicou confirma e recebeu um erro de que o voo estava lotado.

Read):



- **T2** reserva o último lugar e é efetivada.
- **T1** tenta reservar o lugar e ocorre um erro.

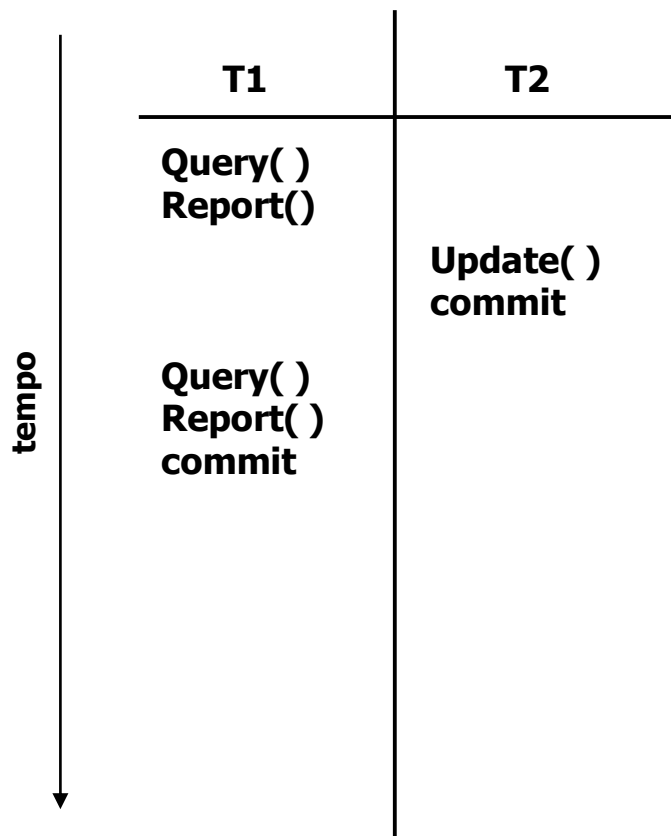


Problemas de Execução Intercalada

- 3) **Leitura fantasma (*Phantom Read*):**
 - transação T lê um conjunto de tuplas que atendam a uma condição de consulta
 - transação T' **insere/remove/atualiza** uma tupla que atenderia a essa condição e é efetivada
 - se T refizer a mesma consulta, obterá um conjunto diferente de tuplas (*phantom read*)

Problemas de Execução Intercalada

■ Ex: Leitura fantasma (*Phantom Read*):



Exemplo:

- **Transação T1:** faz uma consulta que retorna a média geral dos alunos que têm média ponderada acima de 5.0, e gera um relatório
 - **Transação T2:** atualiza as notas de alguns alunos e é efetivada
 - **T1** refaz a consulta para gerar relatório com nro de alunos por faixa de média
- ⇒ **relatórios inconsistentes.**

Problemas de Execução

Repeatable read vs Phantom read

→ *Repeatable read*: lê valores diferentes de um mesmo dado que ainda está lá, mas foi alterado.

→ *Phantom read*: lê conjuntos de dados diferentes, sendo que um dos conjuntos possui dados que não existem no(s) outro(s) conjunto(s) – fantasmas.

→ Phantom reads estão intimamente ligados a predicados que determinam conjuntos de tuplas; é caracterizado pelo surgimento/desaparecimento de tuplas.

tempo

Qu
Re

Qu
Re
co



Problemas de Execução Intercalada → Isolamento

- Ocorrência de anomalias

1. **leitura inválida:** leitura de um dado que não foi consolidado, cujo valor pode ser alterado causando inconsistência;
2. **leitura não repetível:** leitura de um dado consolidado, mas cujo valor foi alterado e consolidado ao longo da transação;
3. **leitura fantasma:** leitura de um conjunto (definido por um predicado) de tuplas consolidadas cujos elementos não se repetem ao longo da transação – tuplas novas surgem e tuplas existentes desaparecem.



Transações – Propriedades ACID

- **Atomacidade:** todas as operações de uma transação devem ser efetivadas; ou, na ocorrência de uma falha, nada deve ser efetivado
 - “tudo ou nada” – não se admite parte de uma operação
- **Consistência:** transações preservam a consistência da base
 - Estado inicial consistente \Rightarrow Estado final consistente
- **Isolamento:** a maneira como várias transações em paralelo interagem (o que pode ser lido e o que pode ser escrito por cada uma) deve ser bem definido
- **Durabilidade:** uma vez consolidada (committed) a transação, suas alterações permanecem no banco até que outras transações aconteçam



Problemas de Execução Intercalada → Isolamento

- Ocorrência de anomalias
 1. leitura inválida
 2. leitura não repetível
 3. leitura fantasma

- Solução via isolamento em diferentes graus
 - Read uncommitted
 - Read committed
 - Repeatable read
 - Serializable



Interpretação

Nível de isolamento	Anomalias que PODEM ocorrer		
	1) Leitura inválida	2) Leitura não repetível	3) Leitura fantasma
Leitura mesmo do que NÃO FOI committed	Sim	Sim	Sim
Leitura apenas do que FOI committed	Não	Sim	Sim
Leitura apenas se a leitura repetida for garantida	Não	Não	Sim
Torna a execução equivalente à execução em série	Não	Não	Não



Níveis de Isolamento em SQL99

Nível de isolamento	Anomalias que PODEM ocorrer		
	1) Leitura inválida	2) Leitura não repetível	3) Leitura fantasma
Read uncommitted	Sim	Sim	Sim
Read committed	Não	Sim	Sim
Repeatable read	Não	Não	Sim
Serializable	Não	Não	Não



Níveis de Isolamento em SQL99

Nível de isolamento	Anomalias que PODEM ocorrer		
	1) Leitura inválida	2) Leitura não repetível	3) Leitura fantasma
Read uncommitted	NÃO EXISTE EM ORACLE Sim	NÃO EXISTE EM ORACLE Sim	Sim
Read committed	MÍNIMO ACEITO EM ORACLE Não	MÍNIMO ACEITO EM ORACLE Sim	Sim
Repeatable read	NÃO EXISTE EM ORACLE Não	NÃO EXISTE EM ORACLE Não	Sim
Serializable	Não	Não	Não



Níveis de Isolamento

- Tanto o PostgreSQL quanto o Oracle não implementam todas os quatro níveis de isolamento previstos pelo padrão SQL
- Em Oracle, apenas os níveis READ COMMITTED e SERIALIZABLE são aceitos; em PostgreSQL, todos os quatro níveis podem ser enunciados, no entanto, o READ UNCOMMITTED opera da mesma maneira que o READ COMMITTED e o REPEATABLE READ opera com as mesmas restrições do SERIALIZABLE.



Transações em ORACLE

- Modo de isolamento: para transações com atualizações
 - READ COMMITTED (padrão):
 - LEITURA: a transação “vê” apenas dados consolidados (committed) antes do início de uma dada operação
 - ESCRITA: antes de uma operação, a transação **aguarda** até que quaisquer tuplas sendo atualizadas sejam liberadas e prossegue
 - SERIALIZABLE:
 - LEITURA: a transação “vê” apenas dados modificados pela própria transação e dados efetivados antes do início da transação
 - ESCRITA: caso uma tupla seja alterada (por outra transação) após o início da transação serializable, **caso a transação serializable tente alterar esta tupla, ela receberá a exceção:**

ORA-08177: Can't serialize access for this transaction.

ou seja, o Oracle informa que não é capaz de tornar a concorrência semelhante a um processamento em série



Transações em ORACLE

- Modo de isolamento: para transações com atualizações

- READ COMMITTED (padrão):

- LEITURA: a transação “vê” apenas dados consolidados (committed) antes do

- **Ideia de Snapshot dos dados antes de uma operação – vários snapshots do banco.**

- SERIALIZABLE:

- LEITURA: a transação “vê” apenas dados modificados pela própria transação e dados efetivados antes do início da transação

- **Ideia de Snapshot dos dados antes de uma transação inteira – um único snapshot.**

ou seja, o Oracle informa que não é capaz de tornar a concorrência semelhante a um processamento em série



Transação em ORACLE

- Comando **SET TRANSACTION**

SET TRANSACTION

READ ONLY | READ WRITE | ISOLATION LEVEL
{SERIALIZABLE | READ COMMITTED} |
NAME 'nome_da_transacao';



Transação em ORACLE

- Comando **COMMIT**
 - termina a transação
 - torna permanente as ações da transação
 - libera os recursos bloqueados



Transação em ORACLE

- Comando **ROLLBACK**
 - desfaz todas as operações da transação
 - libera todos os recursos bloqueados
 - termina transação



Transações e Controle de Concorrência

■ Referências

- *Oracle Database Concepts*
- *OracleSQL Reference*
- Elmasri e Navathe. *Fundamentals of Database Systems*

Views



Exemplo

```
CREATE VIEW PARTIDA_RESUMO(TIME_DA_CASA, GOLS, GOLS_,  
    TIME_DE_FORA, EM) AS  
SELECT  
    TTIME1,GOLS_TIME1,GOLS_TIME2,TTIME2,TO_CHAR(DATAHORA,'DD  
    /MONTH/YYYY')  
FROM F11_PARTIDA
```



```
SELECT *  
FROM PARTIDA_RESUMO
```



Visão (*View*)

- Representação de dados contidos em outras tabelas (**tabelas base**) ou mesmo em outras visões
- Trata resultado de uma consulta como uma tabela
 - **consulta armazenada**
 - **tabela virtual**
- Espaço de armazenamento (no dicionário de dados) **apenas para a consulta (select) que define a visão**
- Consulta é executada cada vez que a visão é acessada



Visão (*View*)

- Utilidade:
 - **segurança** - restrição de acesso a tuplas e colunas
 - armazenamento de consultas complexas ou executadas com muita frequência
 - **simplicidade** para usuário
 - **abstração**
 - **apresentação** dos dados com menor complexidade ou em diferentes perspectivas
 - **isolamento de aplicações** em relação a alterações de esquema



Views no ORACLE

CREATE OR REPLACE VIEW *nome*

[(*NomeColuna* [, *NomeColuna* ...])]

AS <*select*>

[WITH CHECK OPTION | READ ONLY] ;



Visão (*View*)

- **Operações sobre visões**

- **Visões não atualizáveis** (*read-only*)

- seleção

- **Visões atualizáveis** (*updatable*)

- seleção
 - inserção
 - remoção
 - atualização

- **Privilégios**

- *owner* da *view* : operações requerem **privilégios adequados nas tabelas base**
 - outros usuários: requerem os privilégios para a *view*
 - *owner* da *view* pode conceder privilégios de acesso:
 - se for dono das tabelas base
 - se tiver recebido os privilégios com *grant option*



Visão (*View*)

- Visões **inerentemente** NÃO atualizáveis, contêm:
 - operadores de conjunto (UNION, INTERSECT, MINUS...)
 - operador DISTINCT
 - GROUP BY (como parte da visão)
 - ORDER BY..
 - subconsulta na lista da cláusula SELECT
 - *stored procedures*
 - alguns casos de junções

Disciplina = {Sigla, Nome, NCred, Professor, Livro}



Views no ORACLE

- Exemplo simples, todas as tuplas de uma única tabela → **visão atualizável**

```
create or replace view view_disciplina as
select nome, sigla
from disciplina;
```

```
select * from view_disciplina;
```

```
update view_disciplina set nome = 'Alg1.1'
where sigla = 'SCC181';
```

Disciplina = {Sigla, Nome, NCred, Professor, Livro}



Views no ORACLE

- Exemplo simples, todas as tuplas de uma única tabela

```
create or replace view
```

```
select
```

```
from disciplina;
```

A atualização ocorre na tabela base e, consequentemente, se reflete na view.

as

```
select * from view_disciplina;
```

```
update view_disciplina set nome = 'Alg1.1'
where sigla = 'SCC181';
```



Views no ORACLE

- Opção para tornar a *view* *read-only*

```
create view view_disciplina as  
    select nome, sigla  
    from disciplina  
WITH READ ONLY;
```

Professor = {Nome, NFunc, Idade, Titulação}



Views no ORACLE

■ Exemplo

```
create view view_professor_doutor as  
  select * from professor  
  where titulacao = 'DOUTOR'
```

FUNCIONA!!!

```
insert into view_professor_doutor values  
  ('Rogerio', 999, 40, 'TITULAR' );
```



Views no ORACLE

```
create view view_professor_doutor as  
  select * from professor  
  where titulacao = 'DOUTOR'
```

```
insert into view_professor_doutor values  
  ('Rogerio', 999, 40, 'TITULAR' );
```

Qual das consultas seguintes exibe 'Rogério'?

```
select * from professor;  
select * from view_professor_doutor;
```



Views no ORACLE

■ **WITH CHECK OPTION**

- em visões atualizáveis, **WITH CHECK OPTION** não permite operações que violem a condição de seleção que define a visão



Views no ORACLE

■ Exemplo

```
create view view_professor_doutor as  
  select * from professor  
  where titulacao = 'DOUTOR'  
  WITH CHECK OPTION;
```

ERRO!!!

```
insert into view_professor_doutor values  
  ('Rogerio', 999, 40, 'TITULAR');
```




Views no ORACLE

- **CREATE VIEW**
- **ALTER VIEW**
- **DROP VIEW**



Visão Materializada

(materialized view)

- Visões **armazenadas** como tabelas
 - dados provenientes de *master tables (tabelas base)*
- Utilidade
 - replicação de dados
 - desempenho
 - *snapshot* local de dados remotos
 - armazenamento de resultados de consultas complexas e custosas
 - armazenamento de informações sumarizadas
 - **distribuição** de dados



Visão Materializada (*materialized view*)

- Comuns em *data warehousing*, sistemas distribuídos, computação móvel....
- Principais desvantagens:
 - ocupa espaço de armazenamento
 - exige *refresh* quando as *master tables* são modificadas



Visão Materializada no ORACLE

- Visões materializadas
 - por *default*: *read-only*
- Recursos Oracle de *Advanced Replication*
 - permitem que as visões materializadas sejam atualizáveis



Visão Materializada no ORACLE

- Tipos:
 - Visões materializadas com agregações
 - Visões materializadas apenas com junções
 - Visões materializadas aninhadas

Disciplina = {Sigla, Nome, NCred, Professor, Livro}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}



Visão Materializada no ORACLE

Exemplo:

```
SELECT D.Sigla, count(M.Sigla) as
           Nro_Matriculados
FROM Disciplina D, Matricula M
WHERE D.Sigla=M.Sigla
GROUP BY D.Sigla;
```



Visão Materializada no ORACLE

-- logs nas tabelas *master* para o *refresh fast*

-- criados antes da visão

```
CREATE MATERIALIZED VIEW LOG ON  
  Disciplina with ROWID;
```

```
CREATE MATERIALIZED VIEW LOG ON  
  Matricula with ROWID;
```

Disciplina = {Sigla, Nome, NCred, Professor, Livro}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}



Visão Materializada no ORACLE

```
CREATE MATERIALIZED VIEW view_matriculados
BUILD IMMEDIATE
REFRESH FAST ON COMMIT
AS SELECT D.Sigla, count(M.Sigla) as
           Nro_Matriculados
   FROM Disciplina D, Matricula M
  WHERE D.Sigla=M.Sigla
 GROUP BY D.Sigla;
```

➔ Checar a documentação Oracle para a sintaxe completa (rowids)



Visão Materializada no ORACLE

Refresh automático, exemplos:

/*Atualização automática completa de hora em hora*/

```
REFRESH COMPLETE
```

```
START WITH Sysdate NEXT SysDate + 1/24
```

/*Atualização automática incremental uma vez por dia*/

```
REFRESH FAST
```

```
START WITH Sysdate NEXT SysDate + 1
```

Refresh manual, exemplo:

```
EXECUTE DBMS_MVIEWS.REFRESH('nome_view', 'c');
```

```
c - complete
```

```
f - fast
```



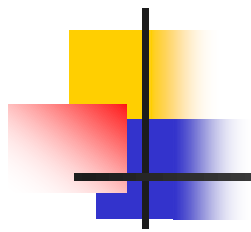
Dicas

- Para consultar informações do dicionário de dados, ou seja, consultar *views* do dicionário:
 - tabelas
 - **SELECT * FROM user_tables**
 - visões, atributos e colunas atualizáveis
 - **SELECT * FROM user_views**
 - **SELECT * FROM user_updatable_columns**
- Documentação sobre dicionário de dados em *Oracle 11g Database Reference*



Onde consultar *Views*

- Documentos em *list of books* no site da Oracle
 - *SQL Reference*
 - *Administrator's Guide*
 - *Concepts*
 - *Data Warehousing Guide*
 - tipos de visões materializadas e restrições de cada tipo para *refresh*
 - *Advanced Replication*
 - uso de visões materializadas em ambientes distribuídos – replicação de dados



Prática 4