**SCC0241 - Laboratório de Bases de Dados**

**Relatório de execução da aula prática 4**

**Nome Nº USP**

**Hamilton Gonçalves Jr 1053763**

**Rafael Souza 7239301**

**Exercício 1**

**Item a**

**➔ Comentário**

Nesse exercício criamos duas conexões as quais chamamos SESSÃO1 e SESSÃO2 e executamos os comandos segundo o roteiro apresentado na prática.

Iniciamos usando isolamento READ COMMITTED e depois usando isolamento SERIALIZABLE.

* **para nível de isolamento READ COMMITTED**

Iniciamos as sessões como recomendado no roteiro (itens i e ii). Na sessão 2 incluímos a transação (item iii):

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

Nessa mesma sessão, fazemos a consulta (item iv):

SELECT JOGADOR FROM F04 INNER JOIN F05

ON F04.TTIME = F05.TTIMEJ;

e selecionamos um jogador pra fazer o experimento, a saber, o jogador de registro:

90000001 Rogério Ceni GOLEIRO São Paulo

Agora na sessão 1 fizemos as seguinte alteração (item v):

UPDATE F05 SET JOGADOR='Rogerio Mito' WHERE CPFJ=90000001;

alterando o nome do goleiro Rogério Ceni para seu "apelido". Repetimos emtão a consulta (item vi) e notamos que a alteração não consta na tabela JOGADOR. Por quê? Porque a transação é READ COMMITTED e foi feita em sessões distintas (em transações distintas portanto). Nesse nível de isolamento, somente dados consolidados são exibidos. Logo, somente após ocorrer COMMIT na sessão 1 é que a sessão 2 "enxergará" a mudança.

Dando COMMIT na sessão 1, consolidando assim a alteração (item vii):

COMMIT;

Repetindo a consulta (item viii) vemos então a alteração feita na sessão 1. Dando COMMIT na sessão 1 (item ix):

COMMIT;

e fazendo novamente a consulta nada de importante acontece (item x).

* **para nível de isolamento SERIALIZABLE**

Agora refizemos todos os itens (de i a x). Iniciamos as sessões como recomendado no roteiro (itens i e ii).

Na sessão 2 incluímos a transação (item iii):

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

Nessa mesma sessão, fazemos a consulta (item iv):

SELECT JOGADOR FROM F04 INNER JOIN F05

ON F04.TTIME = F05.TTIMEJ;

e selecionamos um jogador pra fazer o experimento, a saber, o jogador de registro:

90000001 Rogério Ceni GOLEIRO São Paulo

Agora na sessão 1 fizemos as seguinte alteração (item v):

UPDATE F05 SET JOGADORr='Rogerio Mito' WHERE CPFJ=90000001;

alterando o nome do goleiro Rogério Ceni para seu "apelido". Repetimos emtão a consulta (item vi) e notamos que a alteração não consta na tabela JOGADOR.

Por quê? Porque a transação é READ COMMITTED e foi feita em sessões distintas (em transações distintas portanto). Nesse nível de isolamento, somente dados

consolidados são exibidos. Logo, somente após ocorrer COMMIT na sessão 1 é que a sessão 2 "enxergará" a mudança.

Dando COMMIT na sessão 1, consolidando assim a alteração (item vii):

COMMIT;

Repetindo a consulta (item viii) vemos então a alteração feita na sessão 1. Dando COMMIT na sessão 1 (item ix):

COMMIT;

e fazendo novamente a consulta nada de importante acontece (item x).

**Exercício 2**

**Ítem a**

**➔ Comentário**

O relatório solicitado pode ser obtido pela consulta mostrada na seção Código abaixo. Em seguida aparece a view criada.

**➔ Código**

-- relatório de jogadores que já receberam mais de 1 cartão vermelho

SELECT F05\_JOGADOR.JOGADOR, SUM(VERMELHO) "CARTÕES VERMELHOS"

FROM F15\_GOLS\_CARTOES JOIN F05\_JOGADOR

ON F15\_GOLS\_CARTOES.CPFJ = F05\_JOGADOR.CPFJ

GROUP BY F05\_JOGADOR.JOGADOR

HAVING SUM(VERMELHO)>1;

-- visão criada para gerar o relatório de cartões vermelhos

CREATE VIEW RELATOIO\_CARTOES AS

SELECT F05\_JOGADOR.JOGADOR, SUM(VERMELHO) "CARTÕES VERMELHOS"

FROM F15\_GOLS\_CARTOES JOIN F05\_JOGADOR

ON F15\_GOLS\_CARTOES.CPFJ = F05\_JOGADOR.CPFJ

GROUP BY F05\_JOGADOR.JOGADOR

HAVING SUM(VERMELHO)>1;

-- consulta usando a visão

SELECT \*FROM RELATOIO\_CARTOES;

**➔ Saída**

JOGADOR CARTÕES VERMELHOS

---------------------------------

Julio Cesar 2

Alvaro Arbeloa 2

Dante 2

Shinji Konno 2

View RELATOIO\_CARTOES criado.

JOGADOR CARTÕES VERMELHOS

---------------------------------

Julio Cesar 2

Alvaro Arbeloa 2

Dante 2

Shinji Konno 2

**Ítem b**

**➔ Comentário**

Fizemos junção dos dados da tabela F12\_PATROCINA com F06\_PATROCINADOR (pra obter o nome do patrocinador e não somente o IDPAT que aparece na tabela F12\_PATROCINA) e com F04\_TIME (para pegar a cidade e o Estado do time).

**OBS:** ficamos em dúvida se o patrocinador deveria aparecer na tabela apenas uma vez, mesmo patrocinando vários times. Mas não imaginamos um meio de fazer isso e fizemos da maneira mais óbvia.

**➔ Código**

CREATE VIEW DADOS\_PATROCINIO AS

SELECT F06\_PATROCINADOR.PATROCINADOR,

F12\_PATROCINA.TTIME,

F12\_PATROCINA.CONTRATO,

F12\_PATROCINA.INICIO\_VIGENCIA,

F12\_PATROCINA.TEMPO\_VIGENCIA,

F04\_TIME.CIDADE,

F04\_TIME.ESTADO

FROM F12\_PATROCINA JOIN F06\_PATROCINADOR

ON F12\_PATROCINA.IDPAT = F06\_PATROCINADOR.IDPAT

JOIN F04\_TIME

ON F04\_TIME.TTIME = F12\_PATROCINA.TTIME;

**➔ Saída**

View DADOS\_PATROCINIO criado.

**Exercício 3**

**Ítem a**

**➔ Comentário**

Não conseguimos achar o erro.

**➔ Código**

/\* drops

DROP MATERIALIZED VIEW LOG ON F08\_CAMPEONATO;

DROP MATERIALIZED VIEW LOG ON F11\_PARTIDA;

DROP MATERIALIZED VIEW LOG ON F09\_ESTADIO;

DROP MATERIALIZED VIEW LOG ON F13\_APITA;

DROP MATERIALIZED VIEW LOG ON F07\_ARBITRO;

\*/

-- visão materializada CAMPEONATOS

CREATE MATERIALIZED VIEW LOG ON F08\_CAMPEONATO WITH ROWID;

CREATE MATERIALIZED VIEW LOG ON F11\_PARTIDA WITH ROWID;

CREATE MATERIALIZED VIEW LOG ON F09\_ESTADIO WITH ROWID;

CREATE MATERIALIZED VIEW LOG ON F13\_APITA WITH ROWID;

CREATE MATERIALIZED VIEW LOG ON F07\_ARBITRO WITH ROWID;

CREATE MATERIALIZED VIEW CAMPEONATOS

BUILD IMMEDIATE REFRESH FAST ON COMMIT AS

SELECT F08\_CAMPEONATO.CAMPEONATO,

F11\_PARTIDA.TTIME1,

F11\_PARTIDA.GOLS\_TIME1,

F11\_PARTIDA.TTIME2,

F11\_PARTIDA.GOLS\_TIME2,

F11\_PARTIDA.DATAHORA,

F09\_ESTADIO.ESTADIO,

F07\_ARBITRO.ARBITRO

FROM F08\_CAMPEONATO,

F11\_PARTIDA,

F09\_ESTADIO,

F13\_APITA,

F07\_ARBITRO

WHERE F08\_CAMPEONATO.IDCAMPEONATO = F11\_PARTIDA.IDCAMPEONATOP AND

F11\_PARTIDA.IDESTADIOP = F09\_ESTADIO.IDESTADIO AND

F13\_APITA.IDPARTIDA = F11\_PARTIDA.IDPARTIDA AND

F07\_ARBITRO.CPFA = F13\_APITA.CPFA;

**➔ Saída**

Materialized view LOG eliminado.

Materialized view LOG eliminado.

Materialized view LOG eliminado.

Materialized view LOG eliminado.

Materialized view LOG criado.

Materialized view LOG criado.

Materialized view LOG criado.

Materialized view LOG criado.

Materialized view LOG criado.

Erro a partir da linha : 16 no comando -

CREATE MATERIALIZED VIEW CAMPEONATOS

BUILD IMMEDIATE REFRESH FAST ON DEMAND AS

SELECT F08\_CAMPEONATO.CAMPEONATO,

F11\_PARTIDA.TTIME1,

F11\_PARTIDA.GOLS\_TIME1,

F11\_PARTIDA.TTIME2,

F11\_PARTIDA.GOLS\_TIME2,

F11\_PARTIDA.DATAHORA,

F09\_ESTADIO.ESTADIO,

F07\_ARBITRO.ARBITRO

FROM F08\_CAMPEONATO,

F11\_PARTIDA,

F09\_ESTADIO,

F13\_APITA,

F07\_ARBITRO

WHERE F08\_CAMPEONATO.IDCAMPEONATO = F11\_PARTIDA.IDCAMPEONATOP AND

F11\_PARTIDA.IDESTADIOP = F09\_ESTADIO.IDESTADIO AND

F13\_APITA.IDPARTIDA = F11\_PARTIDA.IDPARTIDA AND

F07\_ARBITRO.CPFA = F13\_APITA.CPFA

Relatório de erros -

Erro de SQL: ORA-12052: não é possível executar a atualização rápida da view materializada L1053763.CAMPEONATOS

12052. 00000 - "cannot fast refresh materialized view %s.%s"

\*Cause: Either ROWIDs of certain tables were missing in the definition or

the inner table of an outer join did not have UNIQUE constraints on

join columns.

\*Action: Specify the FORCE or COMPLETE option. If this error is got

during creation, the materialized view definition may have be

changed. Refer to the documentation on materialized views.