**SCC0241 – Laboratório de Bases de Dados**

**Relatório de execução da aula prática 4**

**Aluno NUSP**

Fábio Alves Martins Pereira 7987435

Matheus Cabral Manoel 9066470

**Exercício X**

**Item x**

**➔ Comentário**

# Este espaço destina-se às questões discursivas.

**➔ Código**

SELECT \*

FROM DUAL;

**➔ Saída**

Este espaço destina-se aos resultados apresentados no terminal da interface SQL – status de execução, mensagens de erro, dados (em forma tabular), e quaisquer outras informações pertinentes. Quando necessário, pode-se enfatizar o texto usando-se negrito ou itálico, mas não *underline*.

# **Observações:**

⚫ O código será copiado deste documento diretamente para execução, portanto, finalize cada linha com um ponto e vírgula; comentários de código também devem ser colocados junto com o código e iniciados pelos caracteres de comentário “--”;

⚫ O código deve ser enfatizado com a fonte Courier New, de preferência com uma cor diferente de preto;

⚫ Faça a cópia do exemplo acima para cada exercício, mantendo a aparência do relatório ao longo dos exercícios;

⚫ Inicie cada exercício (mas não cada item) em uma página nova usando CTRL+ENTER (Insert->Page break);

⚫ Itens que não valem nota, não precisam ser reportados;

⚫ Itens curtos podem ser agrupados quando conveniente;

⚫ Objetive sempre a fácil leitura do documento, a não observância do modelo implicará em perda de nota.

**Exercício 1**

**Item III**

**➔ Comentário**

Sem comentários para este item.

**➔ Código**

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; /\* Primeria execucao \*/

**➔ Saída**

Transaction ISOLATION succeeded.

**Exercício 1**

**Item IV**

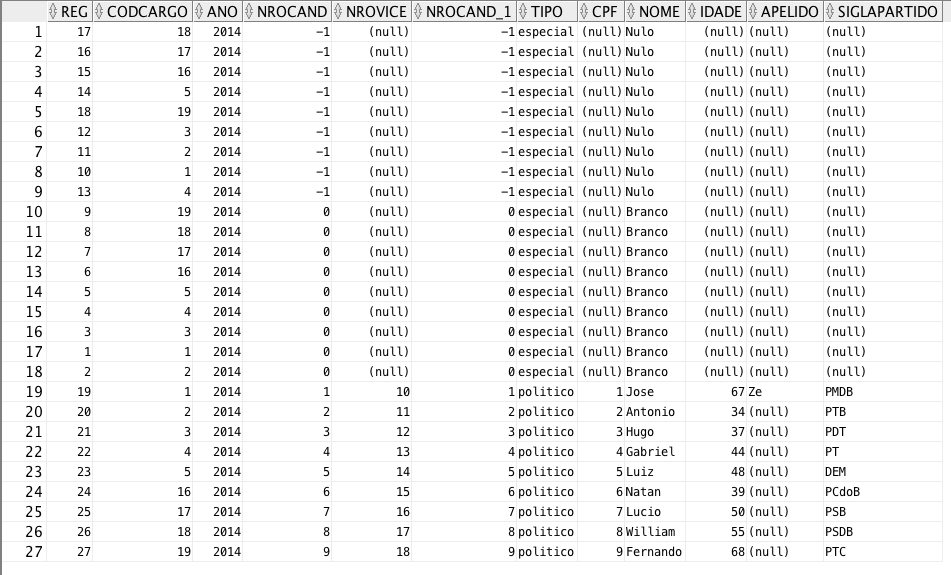
**➔ Comentário**

Sem comentários para este item.

**➔ Código**

SELECT \* FROM CANDIDATURA Cnd JOIN CANDIDATO Cdto ON Cnd.NroCand=Cdto.NroCand ORDER BY Cnd.NroCand;

**➔ Saída**



**Exercício 1**

**Item V**

**➔ Comentário**

Sem comentários para este item.

**➔ Código**

UPDATE CANDIDATURA SET NroVice=10 WHERE NroCand = -1; /\* Serializable \*/

**➔ Saída**

9 rows updated.

**Exercício 1**

**Item VI**

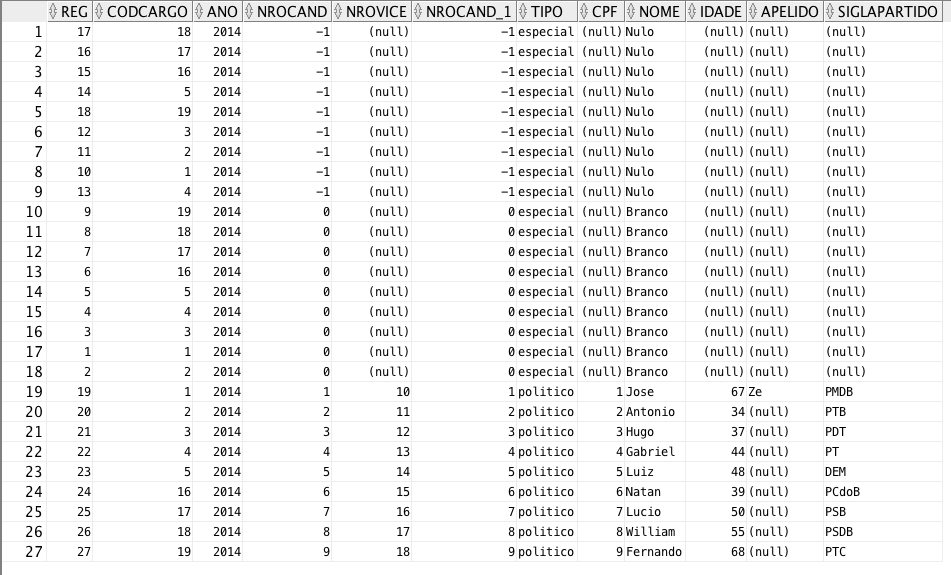
**➔ Comentário**

A alteração feita no item V não foi visualizada pela transação. Os campos NroVice para o candidato -1 deveriam conter o valor 10, porém estão como (null).

**➔ Código**

SELECT \* FROM CANDIDATURA Cnd JOIN CANDIDATO Cdto ON Cnd.NroCand=Cdto.NroCand ORDER BY Cnd.NroCand;

**➔ Saída**

****

**Exercício 1**

**Item VIII**

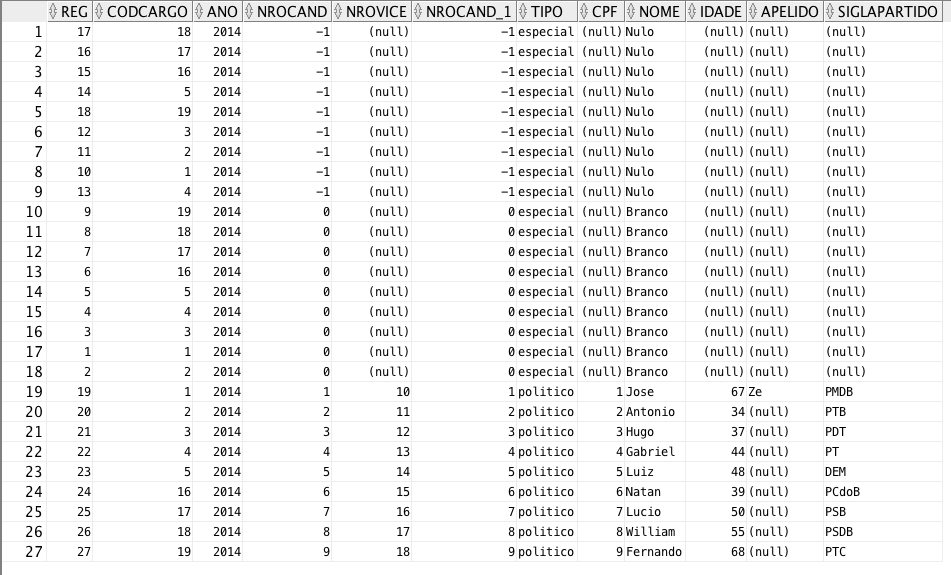
**➔ Comentário**

A alteração feita no item V, mesmo após o commit, não foi visualizada pela transação. Os campos NroVice para o candidato -1 deveriam conter o valor 10, porém estão como (null)

**➔ Código**

SELECT \* FROM CANDIDATURA Cnd JOIN CANDIDATO Cdto ON Cnd.NroCand=Cdto.NroCand ORDER BY Cnd.NroCand;

**➔ Saída**

****

**Exercício 1**

**Item X**

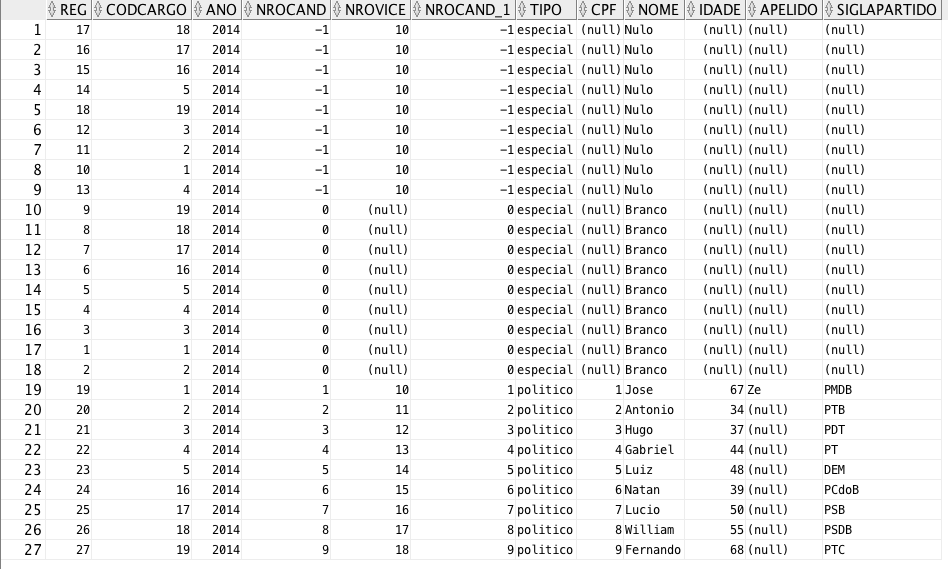
**➔ Comentário**

Após o commit na SESSÂO 2 as alterações feitas no item V foram visualizadas. Os campos NroVice para o candidato -1 agora aparecem com o valor 10.

**➔ Código**

SELECT \* FROM CANDIDATURA Cnd JOIN CANDIDATO Cdto ON Cnd.NroCand=Cdto.NroCand ORDER BY Cnd.NroCand;

**➔ Saída**

****

**Exercício 1**

**Item III**

**➔ Comentário**

Sem comentários para este item.

**➔ Código**

SET TRANSACTION ISOLATION LEVEL READ COMMITTED; /\* Segunda execucao \*/

**➔ Saída**

Transaction ISOLATION succeeded.

**Exercício 1**

**Item IV**

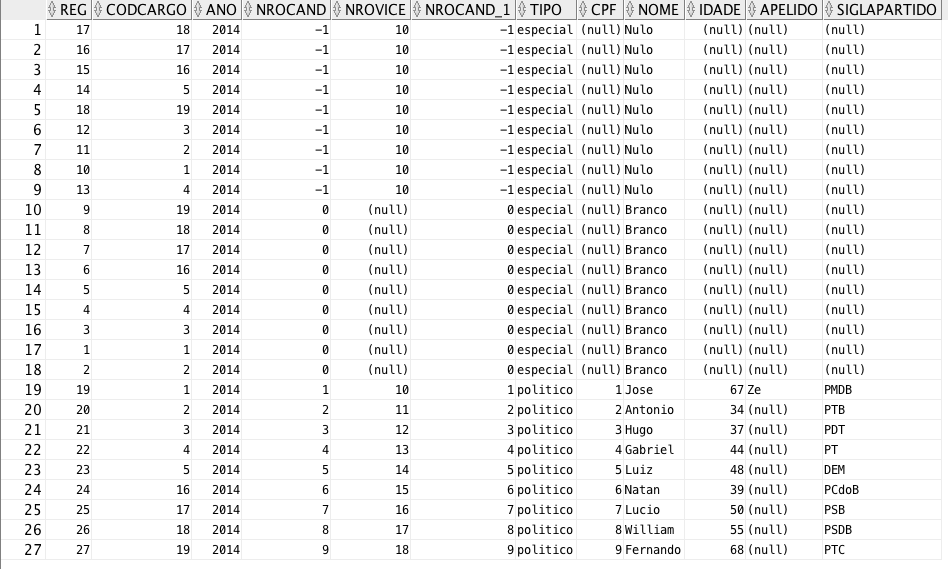
**➔ Comentário**

Sem comentários para este item.

**➔ Código**

SELECT \* FROM CANDIDATURA Cnd JOIN CANDIDATO Cdto ON Cnd.NroCand=Cdto.NroCand ORDER BY Cnd.NroCand;

**➔ Saída**



**Exercício 1**

**Item V**

**➔ Comentário**

Sem comentários para este item.

**➔ Código**

UPDATE CANDIDATURA SET NroVice=12 WHERE NroCand = -1; /\* Read Committed \*/

**➔ Saída**

**9 rows updated.**

**Exercício 1**

**Item VI**

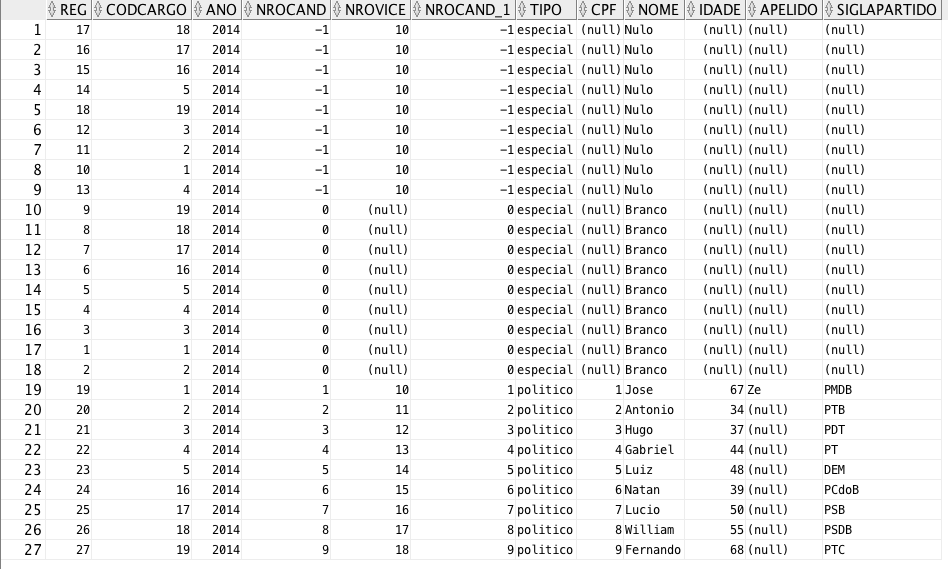
**➔ Comentário**

A alteração feita no item V não foi visualizada pela transação. Os campos NroVice para o candidato -1 deveriam conter o valor 12, porém estão como 10.

**➔ Código**

SELECT \* FROM CANDIDATURA Cnd JOIN CANDIDATO Cdto ON Cnd.NroCand=Cdto.NroCand ORDER BY Cnd.NroCand;

**➔ Saída**

****

**Exercício 1**

**Item VIII**

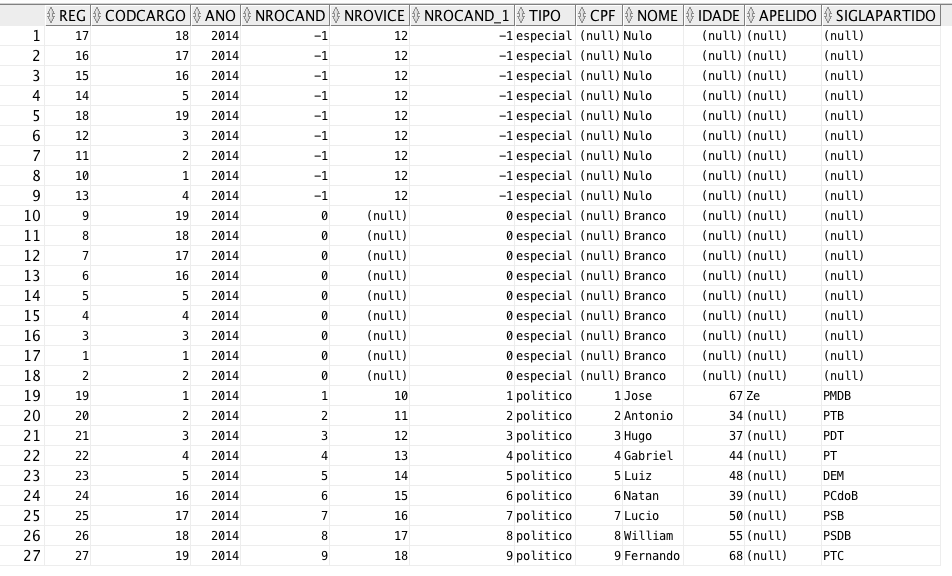
**➔ Comentário**

Após o commit na SESSÂO 1, a alteração feita no item V foi visualizada. Os campos NroVice para o candidato -1 agora aparecem com o valor 12.

**➔ Código**

SELECT \* FROM CANDIDATURA Cnd JOIN CANDIDATO Cdto ON Cnd.NroCand=Cdto.NroCand ORDER BY Cnd.NroCand;

**➔ Saída**

****

**Exercício 1**

**Item X**

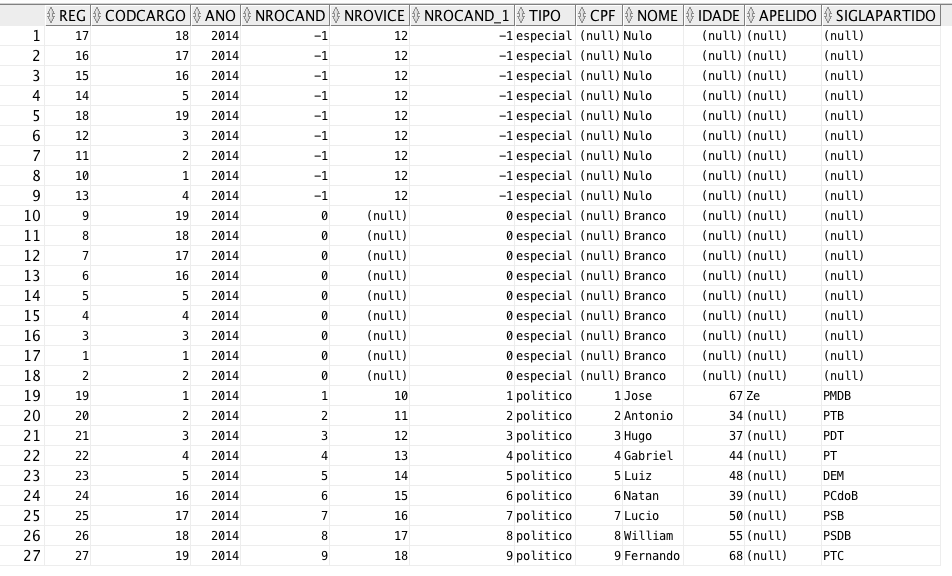
**➔ Comentário**

As alterações feitas já estavam sendo vizualizadas. Após o commit na SESSÂO 2 as alterações feitas no item V continuaram sendo visualizadas. Os campos NroVice para o candidato -1 continuam aparecendo com o valor 12.

**➔ Código**

SELECT \* FROM CANDIDATURA Cnd JOIN CANDIDATO Cdto ON Cnd.NroCand=Cdto.NroCand ORDER BY Cnd.NroCand;

**➔ Saída**

****

**Exercício 2**

**Item a**

**➔ Comentário**

Sem comentários para este exercício.

**➔ Código**

CREATE VIEW MaisDeUmaCandidatura (Nome, CPF, Apelido, NroCandidaturas)

AS

(SELECT Cand.Nome, Cand.CPF, Cand.Apelido, NroCandidaturas FROM Candidato Cand

JOIN

(SELECT NroCand AS Candidato , NroCandidaturas FROM (SELECT Candt.NroCand, COUNT(Candt.Ano) AS NroCandidaturas FROM CANDIDATURA Candt GROUP BY Candt.NroCand) WHERE NroCandidaturas > 1)

ON

Cand.NroCand=Candidato);

**➔ Saída**

View MAISDEUMACANDIDATURA created.

**Exercício 2**

**Item b**

**➔ Comentário**

Atributos escolhidos: Para o Candidato: CPF, Nome, Idade e Apelido do Candidato. Para o Cargo: Nome, Esfera, Cidade e Estado.

Esta visão é atualizável? Por quê?

Não. De acordo com a teoria lida no Slide, a junção de ambas as tabelas, com os atributos que escolhemos, fez com que esta junção não fosse key-preserved. As chaves primárias de ambas tabelas que fazem parte da View não estão contidas na View.

Caso tentemos dar update em atributos que vieram tanto da tabela Candidato, quanto da tabela Cargo obtemos os erros abaixo.

**➔ Código**

CREATE VIEW DadosCandidatura (CPF, Nome, Idade, Apelido, Cargo, Esfera, Cidade, Estado)

AS(

SELECT Cand.CPF, Cand.Nome, Cand.Idade, Cand.Apelido, Crg.NomeDescritivo, Crg.Esfera, Crg.NomeCidade, Crg.SiglaEstado

FROM CANDIDATO Cand, CARGO Crg, CANDIDATURA Cdt WHERE Cand.Nome NOT IN ('Branco', 'Nulo') AND Cand.NroCand=Cdt.NroCand AND Cdt.CodCargo=Crg.CodCargo

);

UPDATE DadosCandidatura SET Apelido = 'Zeh' WHERE CPF = 1;

UPDATE DadosCandidatura SET Cargo = 'Presidentex' WHERE Cargo = 'Presidente';

**➔ Saída**

View DADOSCANDIDATURA created.

Error starting at line : 111 in command -

UPDATE DadosCandidatura SET Apelido = 'Zeh' WHERE CPF = 1

Error at Command Line : 111 Column : 29

Error report -

SQL Error: ORA-01779: cannot modify a column which maps to a non key-preserved table

01779. 00000 - "cannot modify a column which maps to a non key-preserved table"

\*Cause: An attempt was made to insert or update columns of a join view which

map to a non-key-preserved table.

\*Action: Modify the underlying base tables directly.

Error starting at line : 112 in command -

UPDATE DadosCandidatura SET Cargo = 'Presidents' WHERE Cargo = 'Presidente'

Error at Command Line : 112 Column : 56

Error report -

SQL Error: ORA-01779: cannot modify a column which maps to a non key-preserved table

01779. 00000 - "cannot modify a column which maps to a non key-preserved table"

\*Cause: An attempt was made to insert or update columns of a join view which

map to a non-key-preserved table.

\*Action: Modify the underlying base tables directly.

**Exercício 3**

**Item único**

**➔ Comentário**

A junção entre Sessão e Bairro onde o número de zona da sessão é igual à do bairro nos dá todas as informações requisitadas no enunciado do exercício: Considere uma visão materializada com os dados das sessões eleitorais e todos seus dados correlatos, incluindo zona, urna, bairro, cidade, estado, e correspondentes atributos.

**➔ Código**

CREATE MATERIALIZED VIEW LOG ON LE06SESSAO WITH ROWID;

CREATE MATERIALIZED VIEW LOG ON LE04BAIRRO WITH ROWID;

CREATE MATERIALIZED VIEW DadosSessoes

REFRESH FAST AS

SELECT S.ROWID "Sessao\_rid", B.ROWID "Bairro\_rid",

S.NroSessao, S.NroZona, S.Nserial AS Urna, B.Nome, B.NomeCidade, B.SiglaEstado, B.CEP FROM SESSAO S, BAIRRO B WHERE S.NroZona=B.NroZona;

**➔ Saída**

Materialized view LOG created.

Materialized view LOG created.

Materialized view DADOSSESSOES created.

**Exercício 4**

**Item a**

**➔ Comentário**

Quais são os privilégios necessários para que um usuário possa:

* **Realizar operações de DML em suas visões:**
  + - Caso ele seja o OWNER das Tabelas que compõem as visões, ele já possui todos os privilégios necessários para realizar operações de DML.
    - Caso ele não seja o OWNER das tabelas, é preciso que o OWNER delas dê ao usuário as permissões de DELETE, INSERT, UPDATE nos campos das Tabelas que são utilizados pelas visões.
* **Criar visões no próprio esquema:**
  + - Se ele for o OWNER do esquema, ele já possui os privilégios necessários para criar as visões no seu Esquema.
    - Caso contrário, é necessário que o OWNER do esquema dê a ele o priviégio de sistema CREATE VIEW, ou se o OWNER for admin, pode dar também o privilégio CREATE ANY VIEW.
* **Criar visões materializadas no próprio esquema:**
  + Se ele for o OWNER do esquema, ele já possui os privilégios necessários para criar as visões materializadas no seu Esquema.
  + Caso contrário, é necessário que o OWNER do esquema dê a ele o priviégio de sistema CREATE MATERIALIZED VIEW, ou se o OWNER for admin, pode dar também o privilégio CREATE ANY MATERIALIZED VIEW.
* **Criar visões em seu próprio esquema usando tabelas de outros esquemas:**
  + É preciso que os OWNER de outros esquema dê ao usuário a permissão de CREATE VIEW para que ele possa criar as visões utilizando as tabelas que não pertencem ao esquema dele.

**➔ Código**

**➔ Saída**

**Exercício 4**

**Item b**

**➔ Comentário**

USER1: a9066470

USER2: l8937308

**➔ Código**

GRANT SELECT, UPDATE, INSERT ON CANDIDATO TO a9066470 WITH GRANT OPTION;

GRANT SELECT, UPDATE, INSERT ON PARTIDO TO a9066470 WITH GRANT OPTION;

/\* Executado por a9066470 \*/

CREATE VIEW CandidatoPartido (Nome, SiglaPartido, NomePartido)

AS

(SELECT C.Nome, P.Sigla, P.Nome FROM a7987435.CANDIDATO C, a7987435.PARTIDO P WHERE C.SiglaPartido=P.Sigla);

GRANT SELECT ON CandidatoPartido TO l8937308;

/\* Executado por l8937308 \*/

SELECT \* FROM a9066470.CandidatoPartido WHERE SIGLAPARTIDO='PCB';

**➔ Saída**

Materialized view LOG created.

Materialized view LOG created.

Materialized view DADOSSESSOES created.

View CANDIDATOPARTIDO created.

Grant succeeded.

