SCC0241 - Laboratório de Bases de Dados

Relatório de execução da aula prática X

Alunos NUSP

Matheus Cabral Manoel 9066470

Fabio Alves 7987435

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 deste exemplo 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;
* 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 2

Item a

**➔** Comentário

Este espaço destina-se às questões discursivas.

**➔** Código

/\*

Faz: Sequência que gera os números identificadores (ids)

para a tabela ZonaEleitoral

\*/

CREATE SEQUENCE numeroZonaEleitoral

START WITH 1

INCREMENT BY 1

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

/\*

Faz: Sequência que gera os números identificadores (ids)

para a tabela SessaoEleitoral

\*/

CREATE SEQUENCE idSessaoEleitoral

START WITH 1

INCREMENT BY 1

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

/\*

Faz: Sequência que gera os números identificadores (ids)

para a tabela Urna

\*/

CREATE SEQUENCE codigoUrna

START WITH 1

INCREMENT BY 1

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

/\*

Faz: Sequência que gera os números identificadores (ids)

para a tabela Filiacao

\*/

CREATE SEQUENCE idFiliacao

START WITH 1

INCREMENT BY 1

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

/\*

Faz: Sequência que gera os números identificadores (ids)

para a tabela IntencaoVoto

\*/

CREATE SEQUENCE idIntencaoVoto

START WITH 1

INCREMENT BY 1

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

/\*

Sequence: ID Voto

Faz: Sequencia que gera os numeros identificadores (ids) para a tabela Voto

\*/

CREATE SEQUENCE idVoto

START WITH 1

INCREMENT BY 1

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

/\*

Sequence: ID Cargo

Faz: Sequencia que gera os numeros identificadores (ids) para a tabela Cargo

\*/

CREATE SEQUENCE idCargo

START WITH 1

INCREMENT BY 1

NOMAXVALUE

MINVALUE 1

NOCYCLE

NOCACHE;

**➔** Saída

Sequence NUMEROZONAELEITORAL criado.

Sequence IDSESSAOELEITORAL criado.

Sequence CODIGOURNA criado.

Sequence IDFILIACAO criado.

Sequence IDINTENCAOVOTO criado.

Sequence IDVOTO criado.

Sequence IDCARGO criado.

Item b

**➔** Comentário

Este espaço destina-se às questões discursivas.

**➔** Código

/\*

Cria a tabela de Cidades

- nome: nome da cidade

- estado: estado da cidade (SP, RJ, BA etc)

\*/

CREATE TABLE Cidade (

nome VARCHAR2(30) NOT NULL,

estado CHAR(2) NOT NULL,

CONSTRAINT PK\_Cidade PRIMARY KEY (nome, estado)

);

/\*

Cria a tabela de Bairros

- nome: nome do bairro

- cidade: nome da cidade onde o bairro se encontra

- estado: estado do bairro (SP, RJ, BA etc)

\*/

CREATE TABLE Bairro (

nome VARCHAR2(30) NOT NULL,

cidade VARCHAR2(30) NOT NULL,

estado CHAR(2) NOT NULL,

CONSTRAINT PK\_Bairro PRIMARY KEY (nome, cidade, estado),

CONSTRAINT FK\_Bairro\_Cidade FOREIGN KEY (cidade, estado)

REFERENCES Cidade (nome, estado) ON DELETE CASCADE

);

/\*

Cria a tabela de Zona Eleitoral

- numero: numero identificador da zona

- tamanho: numero de votantes na zona

\*/

CREATE TABLE ZonaEleitoral (

numero INTEGER NOT NULL,

tamanho INTEGER NOT NULL,

CONSTRAINT PK\_ZonaEleitoral PRIMARY KEY (numero)

);

/\*

Cria a tabela de Sessao Eleitoral

- id: id identificador da sessao

- numero: numero de votantes da sessao

- zona: zona onde está a sessao

- bairro: bairro da sessao

- estado: estado da sessao

- cidade: cidade da sessao

\*/

CREATE TABLE SessaoEleitoral (

id INTEGER NOT NULL,

numero INTEGER,

zona INTEGER,

bairro VARCHAR2(30),

cidade VARCHAR2(30),

estado CHAR(2),

CONSTRAINT PK\_SessaoEleitoral PRIMARY KEY (id),

CONSTRAINT UQ\_SessaoEleitoral UNIQUE (numero, zona, bairro, cidade, estado),

CONSTRAINT FK\_SessaoEleitoral FOREIGN KEY (bairro, cidade, estado) REFERENCES Bairro(nome, cidade, estado),

CONSTRAINT FK\_SessaoEl\_ZonaEl FOREIGN KEY (zona)

REFERENCES ZonaEleitoral (numero) ON DELETE SET NULL

);

/\*

Cria a tabela de Urna

- codigo: codigo identificador da urna

- fabricante: fabricante da urna

- numRegistro: numero de registro da urna

- zona: zona da urna

- sessao: sessao onde está a urna

\*/

CREATE TABLE Urna (

codigo INTEGER NOT NULL,

fabricante VARCHAR2(20) NOT NULL,

numRegistro INTEGER NOT NULL,

zona INTEGER NOT NULL,

sessao INTEGER NOT NULL,

CONSTRAINT PK\_Urna PRIMARY KEY (codigo),

CONSTRAINT FK\_Urna\_Zona FOREIGN KEY (zona)

REFERENCES ZonaEleitoral (numero) ON DELETE SET NULL,

CONSTRAINT FK\_Urna\_SessaoEleitoral FOREIGN KEY (sessao)

REFERENCES SessaoEleitoral (id)

);

/\*

Cria a tabela de Eleitor

- CPF: cpf do Eleitor

- RG: rg do eleitor

- tituloEleitoral: titulo eleitoral do eleitor

- nome: nome do eleitor

\*/

CREATE TABLE Eleitor (

CPF VARCHAR2(11) NOT NULL,

RG VARCHAR2(15),

tituloEleitoral VARCHAR2(15),

nome VARCHAR2(30),

CONSTRAINT PK\_Eleitor PRIMARY KEY (CPF),

CONSTRAINT UQ\_Eleitor\_RG UNIQUE (RG),

CONSTRAINT UQ\_Eleitor\_tituloEleitoral UNIQUE (tituloEleitoral)

);

/\*

Cria a tabela de VotoBrancoNulo

- eleitor: nome do eleitor do voto

- urna: urna do voto

- data: data do voto

- isBranco: t para true, f para falso

- isNulo: t para true, f para falso

\*/

CREATE TABLE VotoBrancoNulo (

eleitor VARCHAR2(11) NOT NULL,

urna INTEGER NOT NULL,

data DATE NOT NULL,

isBranco CHAR(1),

isNulo CHAR(1),

CONSTRAINT PK\_VotoBrancoNulo PRIMARY KEY (eleitor, urna, data),

CONSTRAINT FK\_VotoBrancoNulo\_Eleitor FOREIGN KEY (eleitor)

REFERENCES Eleitor (CPF) ON DELETE CASCADE,

CONSTRAINT FK\_VotoBrancoNulo\_Urna FOREIGN KEY (urna)

REFERENCES Urna (codigo)

);

/\*

Cria a tabela de Candidato

- CPF: cpf do candidato

- RG: rg do candidato

- sexo: M para masculino, F para feminino

- dataNascimento: data do nascimento do candidato

- naturalidade: naturalidade do candidato

- endereco: endereço do candidato

\*/

CREATE TABLE Candidato (

CPF VARCHAR2(11) NOT NULL,

RG VARCHAR2(15),

nome VARCHAR2(30) NOT NULL,

sexo CHAR(1) NOT NULL,

dataNascimento DATE NOT NULL,

naturalidade VARCHAR2(30) NOT NULL,

endereco VARCHAR2(50) NOT NULL,

CONSTRAINT PK\_Candidato PRIMARY KEY (CPF),

CONSTRAINT UQ\_Candidato UNIQUE (RG)

);

/\*

Cria a tabela de PartidoPolitico

- numeroEleitoral: numero eleitoral do partido

- nome: nome do partido

- sigla: sigla do partido

- dataCriacao: data de criação do partido

- dataRegistro: data de registro do partido

\*/

CREATE TABLE PartidoPolitico (

numeroEleitoral INTEGER NOT NULL,

nome VARCHAR2(50) NOT NULL,

sigla VARCHAR2(5) NOT NULL,

dataCriacao DATE NOT NULL,

dataRegistro DATE NOT NULL,

CONSTRAINT PK\_PartidoPolitico PRIMARY KEY (numeroEleitoral)

);

/\*

Cria a tabela de Filiacao

- id: identificador da filiacao

- numeroFiliacao: numero da filiacao

- candidato: cpf do candidato da filiacao

- partido: numero id do partido da filiacao

\*/

CREATE TABLE Filiacao (

id INTEGER NOT NULL,

numeroFiliacao INTEGER,

candidato VARCHAR2(11),

partido INTEGER,

CONSTRAINT PK\_Filiacao PRIMARY KEY (id),

CONSTRAINT UQ\_Filiacao UNIQUE (numeroFiliacao, candidato, partido),

CONSTRAINT FK\_Filiacao\_Candidato FOREIGN KEY (candidato)

REFERENCES Candidato(CPF),

CONSTRAINT FK\_Filiacao\_PartidoPolitico FOREIGN KEY (partido)

REFERENCES PartidoPolitico(numeroEleitoral)

);

/\*

Table: Cargo

O que armazena: O numero identificador do cargo e a esfera a qual ele pertence

As possiveis esferas sao: Municipal, Estadual e Federal

\*/

CREATE TABLE Cargo (

id INTEGER NOT NULL,

esfera VARCHAR2(9) NOT NULL, /\* Contem 9 caracteres pois a maior palavra armazenada e 'Municipal' \*/

CONSTRAINT PK\_CARGO PRIMARY KEY (id),

CONSTRAINT CHECK\_ESFERA CHECK (UPPER(esfera) in ('MUNICIPAL','ESTADUAL','FEDERAL'))

);

/\*

Cria a tabela de Concorrente

- filiacao: numero identificador da filiacao do Concorrente

- idCargo: id do cargo do Concorrente

\*/

CREATE TABLE Concorrente (

filiacao INTEGER NOT NULL,

idCargo INTEGER NOT NULL,

CONSTRAINT PK\_Concorrente PRIMARY KEY (filiacao, idCargo),

CONSTRAINT FK\_Concorrente\_Filiacao FOREIGN KEY (filiacao)

REFERENCES Filiacao(id),

CONSTRAINT FK\_Concorrente\_Cargo FOREIGN KEY (idCargo)

REFERENCES Cargo(id)

);

/\*

Table: Voto

O que armazena: Os votos de eleitores em uma urna, para um determinado candidato

e em uma determinada data

\*/

CREATE TABLE Voto (

id INTEGER NOT NULL,

eleitor VARCHAR2(11) NOT NULL,

urna INTEGER,

filiacaoConcorrente INTEGER,

cargoConcorrente INTEGER,

data DATE NOT NULL,

CONSTRAINT PK\_VOTO PRIMARY KEY (id),

CONSTRAINT SK\_VOTO UNIQUE (eleitor, urna, filiacaoConcorrente, cargoConcorrente, data),

CONSTRAINT TK\_VOTO UNIQUE (eleitor, data),

CONSTRAINT FK\_VOTO\_ELEITOR FOREIGN KEY (eleitor) REFERENCES Eleitor(CPF),

CONSTRAINT FK\_VOTO\_URNA FOREIGN KEY (urna) REFERENCES Urna(codigo) ON DELETE SET NULL,

CONSTRAINT FK\_VOTO\_CONCORRENTE FOREIGN KEY (filiacaoConcorrente,cargoConcorrente) REFERENCES Concorrente(filiacao,idCargo) ON DELETE SET NULL

);

/\*

Cria a tabela de IntencaoVoto

- id: numero identificador da intencao de voto

- eleitor: eleitor da intencao de voto

- filiacaoConcorrente: filiacao concorrente da intencao de voto

- cargoConcorrente: cargo concorrente da intencao de voto

- data: data da intencao de voto

\*/

CREATE TABLE IntencaoVoto (

id INTEGER NOT NULL,

eleitor VARCHAR2(11),

filiacaoConcorrente INTEGER,

cargoConcorrente INTEGER,

data DATE NOT NULL,

CONSTRAINT PK\_IntencaoVoto PRIMARY KEY (id),

CONSTRAINT UQ\_IntencaoVoto UNIQUE (eleitor, filiacaoConcorrente,

cargoConcorrente),

CONSTRAINT UQ\_IntencaoVoto1 UNIQUE (eleitor, data),

CONSTRAINT FK\_IntencaoVoto\_Concorrente FOREIGN KEY (filiacaoConcorrente,

cargoConcorrente)

REFERENCES Concorrente(filiacao, idCargo)

);

/\*

Table: Municipal

O que armazena: O numero identificador de um cargo municipal e o tipo do cargo municipal.

Os possiveis cargos sao: Vereador e Prefeito

\*/

CREATE TABLE Municipal (

cargo INTEGER NOT NULL,

tipo CHAR(8) NOT NULL, /\* Contem exatamente 8 caracteres pois as duas palavras armazenadas: Vereador e Prefeito tem o mesmo tamanho \*/

CONSTRAINT PK\_MUNICIPAL PRIMARY KEY (cargo),

CONSTRAINT FK\_MUNICIPAL\_CARGO FOREIGN KEY (cargo) REFERENCES Cargo(id) ON DELETE CASCADE,

CONSTRAINT CHECK\_TIPO\_MUNICIPAL CHECK(UPPER(tipo) in ('VEREADOR', 'PREFEITO'))

);

/\*

Table: Estadual

O que armazena: O numero identificador de um cargo estadual e o tipo do cargo estadual.

Os possiveis cargos sao: Deputado Estadual e Governador

\*/

CREATE TABLE Estadual (

cargo INTEGER NOT NULL,

tipo VARCHAR2(17) NOT NULL, /\* VARCHAR(17) pois a palavra armazenada de maior tamanho e DEPUTADO ESTADUAL que tem 17 caracteres \*/

CONSTRAINT PK\_ESTADUAL PRIMARY KEY (cargo),

CONSTRAINT FK\_ESTADUAL\_CARGO FOREIGN KEY (cargo) REFERENCES Cargo(id) ON DELETE CASCADE,

CONSTRAINT CHECK\_TIPO\_ESTADUAL CHECK(UPPER(tipo) in ('DEPUTADO ESTADUAL', 'GOVERNADOR'))

);

/\*

Table: Federal

O que armazena: O numero identificador de um cargo federal e o tipo do cargo federal.

Os possiveis cargos sao: Deputado Federal, Senador e Presidente

\*/

CREATE TABLE Federal (

cargo INTEGER NOT NULL,

tipo VARCHAR2(16) NOT NULL, /\* VARCHAR(16) pois a palavra armazenada de maior tamanho e DEPUTADO FEDERAL que tem 17 caracteres \*/

CONSTRAINT PK\_FEDERAL PRIMARY KEY (cargo),

CONSTRAINT FK\_FEDERAL\_CARGO FOREIGN KEY (cargo) REFERENCES Cargo(id) ON DELETE CASCADE,

CONSTRAINT CHECK\_TIPO\_FEDERAL CHECK(UPPER(tipo) in ('DEPUTADO FEDERAL', 'SENADOR', 'PRESIDENTE'))

);

/\*

Table: Vereador

O que armazena: O numero identificador do cargo de vereador, o nome do cargo,

numero da cadeira do vereador, quantos anos de mandato tem o cargo e o ano da

ultima eleicao para verador no Brasil

\*/

CREATE TABLE Vereador (

cargo INTEGER NOT NULL,

nome CHAR(8) NOT NULL,

numCadeira INTEGER NOT NULL,

anosMandato INTEGER NOT NULL,

anoBase DATE,

CONSTRAINT PK\_VEREADOR PRIMARY KEY (cargo),

CONSTRAINT SK\_VEREADOR UNIQUE (nome, numCadeira),

CONSTRAINT FK\_VEREADOR\_MUNICIPAL FOREIGN KEY (cargo) REFERENCES Municipal(cargo) ON DELETE CASCADE

);

/\*

Table: Prefeito

O que armazena: O numero identificador do cargo de prefeito, o nome do cargo,

numero da cadeira do prefeito, quantos anos de mandato tem o cargo e o ano da

ultima eleicao para prefeito no Brasil

\*/

CREATE TABLE Prefeito (

cargo INTEGER NOT NULL,

nome CHAR(8) NOT NULL,

numCadeira INTEGER NOT NULL,

anosMandato INTEGER NOT NULL,

anoBase DATE,

vice CHAR(1) NOT NULL, /\* Diz de o cargo em questao e de prefeito 'F' ou vice-prefeito 'T' \*/

CONSTRAINT PK\_PREFEITO PRIMARY KEY (cargo),

CONSTRAINT SK\_PREFEITO UNIQUE (nome, numCadeira),

CONSTRAINT FK\_PREFEITO\_MUNICIPAL FOREIGN KEY (cargo) REFERENCES Municipal(cargo) ON DELETE CASCADE

);

/\*

Table: Deputado Estadual

O que armazena: O numero identificador do cargo de Deputado Estadual, o nome do cargo,

numero da cadeira do Deputado Estadual, quantos anos de mandato tem o cargo e o ano da

ultima eleicao para Deputado Estadual no Brasil

\*/

CREATE TABLE DeputadoEstadual (

cargo INTEGER NOT NULL,

nome CHAR(17) NOT NULL,

numCadeira INTEGER NOT NULL,

anosMandato INTEGER NOT NULL,

anoBase DATE,

CONSTRAINT PK\_DEPUTADO\_ESTADUAL PRIMARY KEY (cargo),

CONSTRAINT SK\_DEPUTADO\_ESTADUAL UNIQUE (nome, numCadeira),

CONSTRAINT FK\_DEPUTADO\_ESTADUAL\_ESTADUAL FOREIGN KEY (cargo) REFERENCES Estadual(cargo) ON DELETE CASCADE

);

/\*

Table: Governador

O que armazena: O numero identificador do cargo de Governador, o nome do cargo,

numero da cadeira do Governador, quantos anos de mandato tem o cargo e o ano da

ultima eleicao para Governador no Brasil

\*/

CREATE TABLE Governador (

cargo INTEGER NOT NULL,

nome CHAR(10) NOT NULL,

numCadeira INTEGER NOT NULL,

anosMandato INTEGER NOT NULL,

anoBase DATE,

vice CHAR(1) NOT NULL, /\* Se o cargo for de Governador recebe 'F' se for Vice-Governador recebe 'T' \*/

CONSTRAINT PK\_GOVERNADOR PRIMARY KEY (cargo),

CONSTRAINT SK\_GOVERNADOR UNIQUE (nome, numCadeira),

CONSTRAINT FK\_GOVERNADOR\_ESTADUAL FOREIGN KEY (cargo) REFERENCES Estadual(cargo) ON DELETE CASCADE

);

/\*

Table: Deputado Federal

O que armazena: O numero identificador do cargo de Deputado Federal, o nome do cargo,

numero da cadeira do Deputado Federal, quantos anos de mandato tem o cargo e o ano da

ultima eleicao para Deputado Federal no Brasil

\*/

CREATE TABLE DeputadoFederal (

cargo INTEGER NOT NULL,

nome CHAR(16) NOT NULL,

numCadeira INTEGER NOT NULL,

anosMandato INTEGER NOT NULL,

anoBase DATE,

CONSTRAINT PK\_DEPUTADO\_FEDERAL PRIMARY KEY (cargo),

CONSTRAINT SK\_DEPUTADO\_FEDERAL UNIQUE (nome, numCadeira),

CONSTRAINT FK\_DEPUTADO\_FEDERAL\_FEDERAL FOREIGN KEY (cargo) REFERENCES Federal(cargo) ON DELETE CASCADE

);

/\*

Table: Senador

O que armazena: O numero identificador do cargo de Senador, o nome do cargo,

numero da cadeira do Senador, quantos anos de mandato tem o cargo e o ano da

ultima eleicao para Senador no Brasil

\*/

CREATE TABLE Senador (

cargo INTEGER NOT NULL,

nome CHAR(7) NOT NULL,

numCadeira INTEGER NOT NULL,

anosMandato INTEGER NOT NULL,

anoBase DATE,

CONSTRAINT PK\_SENADOR PRIMARY KEY (cargo),

CONSTRAINT SK\_SENADOR UNIQUE (nome, numCadeira),

CONSTRAINT FK\_SENADOR\_FEDERAL FOREIGN KEY (cargo) REFERENCES Federal(cargo) ON DELETE CASCADE

);

/\*

Table: Presidente

O que armazena: O numero identificador do cargo de Presidente, o nome do cargo,

numero da cadeira do Presidente, quantos anos de mandato tem o cargo e o ano da

ultima eleicao para Presidente no Brasil

\*/

CREATE TABLE Presidente (

cargo INTEGER NOT NULL,

nome CHAR(10) NOT NULL,

numCadeira INTEGER NOT NULL,

anosMandato INTEGER NOT NULL,

anoBase DATE,

vice CHAR(1) NOT NULL, /\* Se o cargo for de Presidente recebe 'F' se for Vice-Presidente recebe 'T' \*/

CONSTRAINT PK\_PRESIDENTE PRIMARY KEY (cargo),

CONSTRAINT SK\_PRESIDENTE UNIQUE (nome, numCadeira),

CONSTRAINT FK\_PRESIDENTE\_FEDERAL FOREIGN KEY (cargo) REFERENCES Federal(cargo) ON DELETE CASCADE

);

**➔** Saída

Table CIDADE criado.

Table BAIRRO criado.

Table ZONAELEITORAL criado.

Table SESSAOELEITORAL criado.

Table URNA criado.

Table ELEITOR criado.

Table VOTOBRANCONULO criado.

Table CANDIDATO criado.

Table PARTIDOPOLITICO criado.

Table FILIACAO criado.

Table CARGO criado.

Table CONCORRENTE criado.

Table VOTO criado.

Table INTENCAOVOTO criado.

Table MUNICIPAL criado.

Table ESTADUAL criado.

Table FEDERAL criado.

Table VEREADOR criado.

Table PREFEITO criado.

Table DEPUTADOESTADUAL criado.

Table GOVERNADOR criado.

Table DEPUTADOFEDERAL criado.

Table SENADOR criado.

Table PRESIDENTE criado.

Exercício 3

Item a

**➔** 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*.

Item b

**➔** 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*.

Exercício 4

Item a

**➔** Comentário

Os valores das tuplas adicionadas anteriormente recebem nessa coluna o valor default, isso se deve ao "NOT NULL". Caso o NOT NULL fosse omitido, todos as tuplas previamente adicionadas não sofreriam alteração alguma.

**➔** Código

ALTER TABLE Cidade

ADD isCapital CHAR(1) DEFAULT 'f' NOT NULL

CONSTRAINT CK\_Cidade CHECK (UPPER(isCapital) in ('T', 'F'));

**➔** Saída

Table CIDADE alterado.

Item b

**➔** Comentário

Este espaço destina-se às questões discursivas.

**➔** Código

ALTER TABLE PartidoPolitico

ADD CONSTRAINT UQ\_PartidoPolitico UNIQUE(nome);

**➔** Saída

Table PARTIDOPOLITICO alterado.

Item c

**➔** Comentário

Sem cascade a saída é um erro de impossibilidade de realizar a ação. Com CASCADE CONSTRAINTS a coluna na tabela em que havia a PK é removida em todas as tuplas já existentes, já na tabela em que havia a FK os valores da coluna continuam lá, mas a CONSTRAINT FOREIGN KEY de referência é removida.

**➔** Código

ALTER TABLE SessaoEleitoral DROP COLUMN numero;

ALTER TABLE SessaoEleitoral DROP COLUMN numero CASCADE CONSTRAINTS;

**➔** Saída

ALTER TABLE SessaoEleitoral DROP COLUMN numero

Relatório de erros -

Erro de SQL: ORA-12991: a coluna é indicada em uma restrição de várias colunas

12991. 00000 - "column is referenced in a multi-column constraint"

\*Cause: An attempt was made to drop a column referenced by some

constraints.

\*Action: Drop all constraints referencing the dropped column or

specify CASCADE CONSTRAINTS in statement.

Table SESSAOELEITORAL alterado.

Item d

**➔** Comentário

A ativação do check não funciona pois a tabela já possui uma tupla com valores conflitantes com o check.

**➔** Código

ALTER TABLE Cargo

DISABLE CONSTRAINT CHECK\_ESFERA;

INSERT INTO Cargo VALUES(idCargo.NextVal, 'MUNDIAL');

ALTER TABLE Cargo

ENABLE CONSTRAINT CHECK\_ESFERA;

**➔** Saída

Table CARGO alterado.

1 linha inserido.

Erro a partir da linha : 42 no comando -

ALTER TABLE Cargo

ENABLE CONSTRAINT CHECK\_ESFERA

Relatório de erros -

Erro de SQL: ORA-02293: não é possível validar (A9066470.CHECK\_ESFERA) - restrição de verificação violada

02293. 00000 - "cannot validate (%s.%s) - check constraint violated"

\*Cause: an alter table operation tried to validate a check constraint to

populated table that had nocomplying values.

\*Action: Obvious