

Banco de Dados I

Constraint (Integridade/Restrições)

Prof. Msc. Aparecido Vilela Junior
aparecido.vilela@unicesumar.edu.br

CREATE TABLE

- Para criar uma tabela, primeiramente é necessário ter privilégio para criá-la
- Deve especificar:
 - Nome da Tabela (Table Name)
 - Nome, tipo e tamanho das colunas

```
CREATE TABLE [schema.] table  
    (column datatype [DEFAULT expr] [...]);
```

A opção DEFAULT

- Especifica um valor default para uma coluna durante um INSERT
- Valores literais, expressões, ou funções SQL são valores aceitos
- Nome de uma outra coluna, ou pseudo-colunas são valores NÃO aceitos
- O tipo de dados default, deve combinar com o tipo de dados da coluna

ALTER TABLE

- Utilizamos o ALTER TABLE para:
 - Adicionar uma nova coluna
 - Modificar uma coluna existente;
 - Definir um valor default para uma nova coluna
 - Dropar (eliminar) uma coluna

ALTER TABLE - ADD

- Adicionando uma coluna:
 - Use a cláusula ADD para adicionar colunas
 - A nova coluna criada torna-se a última coluna da tabela.

```
ALTER TABLE DEPT  
ADD (Uf varchar2(02));
```

ALTER TABLE - MODIFY

- Modificando uma coluna:
 - Para alterar um tipo de dados da coluna, tamanhos e valores default.
 - Uma alteração para valores default, afetam somente inserções subsequentes na tabela
 - ALTER TABLE dept
 - MODIFY DNAME VARCHAR2(30);

ALTER TABLE - DROP

- Dropando uma coluna:
 - Utilize o DROP COLUMN para eliminar colunas que não são mais necessárias na tabela
 - Somente uma coluna de cada vez, pode ser eliminada
 - Uma coluna dropada, não pode ser recuperada

```
ALTER TABLE DEPT  
DROP COLUMN UF;
```

DROP TABLE

- Todos os dados e a estrutura da tabela são eliminados
- Algumas transações pendentes são commitadas
- Todos os índices são apagados
- O comando Rollback, não tem efeito, após um DROP TABLE

CONSTRAINT

- Utiliza-se CONSTRAINTS para impor uma ou mais das seguintes restrições sobre uma coluna ou grupos de colunas:
 - Requerer que uma coluna ou grupo de colunas contenham valores NOT NULL
 - Especificar que o valor de uma coluna seja único na tabela referida
 - Especificar colunas como CHAVE PRIMÁRIA (Primary Key)
 - Especificar uma restrição de CHAVE ESTRANGEIRA (Foreign Key)
 - Requerer que o valor da coluna seja conforme uma valor pré-determinado (CHECK)

CONSTRAINT

- Podem ser definidas para uma tabela e colunas e são especificadas com o parte dos comandos CREATE ou ALTER TABLE
- É um conjunto de restrições de valores para validação.
- Toda declaração INSERT, UPDATE e DELETE causam uma avaliação de constraint.

CONSTRAINT – NOT NULL

- Definida no nível da coluna:

```
CREATE TABLE CO_NULO  
(CODIGO NUMBER(6),  
  NOME  VARCHAR2(25) NOT NULL,  
  SALARIO NUMBER(8,2),  
  DT_ADM DATE CONSTRAINT ADMISSAO_NN NOT NULL)
```

CONSTRAINT – PRIMARY KEY

- Pode ser elaborada na coluna ou na tabela

```
CREATE TABLE CO_PRIMARIA  
(ID_DEPTO NUMBER(4),  
  NOME VARCHAR2(30) CONSTRAINT DP_NAME_NN  
  NOT NULL,  
  ID_LOC NUMBER(4),  
  CONSTRAINT CO_PRIMARIA_ID_PK PRIMARY  
  KEY(ID_DEPTO))
```

CONSTRAINT – FOREIGN KEY

- Pode ser elaborada na coluna ou na tabela.

```
CREATE TABLE CO_SECUNDARIA  
(EMPRESA NUMBER(06),  
  NOME   VARCHAR2(25) NOT NULL,  
  DEPARTAMENTO_ID NUMBER(04),  
  CONSTRAINT CO_SECUNDARIA_DP_FK FOREIGN KEY  
    (DEPARTAMENTO_ID)  
  REFERENCES CO_PRIMARIA (ID_DEPTO))
```

CONSTRAINT – FOREIGN KEY

- Palavras chaves:
 - Foreign Key: Define a coluna ligada a uma outra tabela
 - References: Identifica a tabela e coluna na tabela relacionada
 - On Delete Cascade: Apaga as linhas dependentes na tabela filha quando a linha é apagada na tabela pai
 - On Delete Set Null: Converte os valores da foreign key para NULL

- Uma manipulação de dados é executada quando:
 - Adiciona-se informações (linhas) na tabela
 - Modifica-se informações existentes na tabela
 - Remove-se informações existentes na tabela.

INSERT

- O comando INSERT é utilizado para inserir linhas em tabelas.
- A inclusão pode ser feita linha a linha ou os valores podem ser obtidos de outras tabelas.
- Podemos definir quais colunas serão preenchidas, informando valores para somente estas colunas.

INSERT - Sintaxe

- Adicionando novas linhas para uma tabela:
 - INSERT INTO table [(column [,column...])]
 - VALUES (value [,value...]);
- Somente uma linha por vez é inserida com esta sintaxe

INSERT

- A inserção de novas linhas (informações) poderá ser feita:
 - Inserindo uma nova linha contendo valores para cada coluna

```
INSERT INTO DEPT (DEPTNO, DNAME, LOC)  
VALUES (60, 'POS BANCO', 'LONDRINA')
```

- Associando valores a cada uma das colunas, não informando as colunas da tabela.

```
INSERT INTO DEPT VALUES (65, 'ORACLE', 'MARINGA', 'PR')
```

INSERT

- Inserindo linhas com o comando SELECT, para efetuar a inclusão de diversas linhas simultaneamente.
 - insert into dept (deptno, dname)
 - select 99, 'teste' from dual
- Strings e Datas dever ser escritas entre aspas simples.

INSERT

- Inserindo linhas com valores nulos.
 - Método Implícito: Omitir a coluna da lista.

```
INSERT INTO DEPT (DEPTNO, DNAME)  
VALUES (100,'RH');
```

Método Explícito: Especifica a palavra NULL nos valores.

```
INSERT INTO DEPT  
VALUES (110,'CONTABIL',NULL)
```

INSERT

- Inserindo valores especiais: Data Corrente

```
INSERT INTO EMP (EMPNO, ENAME, DEPTNO, HIREDATE)
VALUES
(1234,'NOME TESTE',10,SYSDATE)
```

Inserindo valores específicos da data.

```
INSERT INTO EMP (EMPNO, ENAME, DEPTNO, HIREDATE)
VALUES
(1234,'NOME TESTE',10,TO_DATE('JAN 3, 1969', 'MON DD, YYYY' ))
```

UPDATE

- O comando UPDATE tem a finalidade de alterar informações já gravadas na base de dados.
- Possui uma cláusula WHERE, que determinará quais linhas serão modificadas.
- O Oracle faz um select implícito no banco de dados, para determinar as linhas que atendem a cláusula WHERE.

UPDATE

- Alterando linhas em uma tabela:

UPDATE TABLE

SET column = value [, column = value,...]

[WHERE condition];

UPDATE

- A linha ou linhas que serão alteradas são especificadas na cláusula WHERE.
- UPDATE EMP
 SET DEPTNO = 20
 WHERE EMPNO = 1234
- Todas as linhas serão alteradas se for omitida a cláusula WHERE
- UPDATE EMP
 SET DEPTNO = 20

UPDATE

- Atualizando Linhas: Erro de Constraint de Integridade:

```
UPDATE EMP  
SET DEPTNO = 55  
WHERE DEPTNO = 110
```

- O departamento 55 não existe

DELETE

- Excluir linhas cadastradas no banco de dados.
- Da mesma forma que o UPDATE, a cláusula WHERE será responsável por determinar que linhas poderão ser removidas.
- Caso essa cláusula não seja informada, o comando tentará remover todas as linhas da tabela informada

DELETE

- Removendo linhas de uma tabela:

```
DELETE [FROM] table  
[WHERE condition]
```

Obs.: Se nenhuma linha for excluída, a mensagem '0 rows deleted' é retornada

DELETE

- Especificando linhas a serem excluídas pela cláusula WHERE:

```
DELETE FROM DEPT  
WHERE DNAME = 'Finance'
```

- Todas as linhas da tabela serão excluídas se for omitida a cláusula WHERE:

```
DELETE FROM DEPT
```

DELETE

- Removendo Linhas: Erro de Constraint de Integridade:
 - Não pode ser excluída uma linha que contenha uma primary key, que é utilizada como foreign key em outra tabela.

CONSTRAINT

```
CREATE TABLE [schema.] table  
  (column datatype [DEFAULT expr]  
  [column constraint]  
  ...  
  [table_constraint] [...]);
```

CONSTRAINT

- Existem dois tipos de restrições (constraints), de tabelas e colunas.
- As restrições de colunas refere-se a uma única coluna;
- A restrição de tabela pode referenciar-se a uma ou mais colunas.

CONSTRAINT - UNIQUE

- Definida no nível da coluna ou da tabela.

```
CREATE TABLE CO_UNICA  
(CODIGO NUMBER(6),  
  NOME  VARCHAR2(25) NOT NULL,  
  EMAIL VARCHAR2(25),  
  SALARIO NUMBER(8,2),  
  DT_ADMISSAO DATE CONSTRAINT ADMISSAO2_NN NOT  
  NULL,  
  CONSTRAINT CO_UNICA_EMAIL_UN UNIQUE (EMAIL))
```


CONSTRAINT - CHECK

- Define uma condição para ser satisfeita
- Uma coluna pode ter múltiplos CHECK constraints.
- Não há limites para o número de CHECK, a serem definidos para uma coluna.
 -
 - SALARIO NUMBER(7,2) CONSTRAINT
EMP_SALARIO_MINIMO CHECK (SALARIO > 0)

Adicionando uma CONSTRAINT

- Pode-se utilizar o ALTER TABLE, para:
 - Adicionar ou dropar uma constraint, mas não modificar sua estrutura;
 - Habilitar ou desabilitar constraints;
 - Adicionar uma constraint NULL, pode ser feita usando a cláusula MODIFY
 - ALTER TABLE *table*
 - ADD [CONSTRAINT *constraint*] type (column)

Adicionando uma Constraint

- Adicionando uma Foreign Key, por exemplo para indicar um auto-relacionamento na tabela EMP, indicando que o gerente somente deve ser cadastrado se houver um código de funcionário válido na tabela

```
ALTER TABLE EMP
```

```
ADD CONSTRAINT EMP_MGR_FK FOREIGN KEY  
(MGR) REFERENCES EMP (EMPNO)
```

Eliminando uma CONSTRAINT

- Para eliminar uma constraint, utiliza-se a cláusula DROP

- CONSTRAINT.

```
ALTER TABLE Nome_Tabela
```

```
DROP CONSTRAINT Nome_Constraint;
```

- Para desativar uma CONSTRAINT, pode-se utilizar a cláusula
- **DISABLE CONSTRAINT**

Eliminando uma Constraint

- Removendo a Constraint da coluna MGR, da tabela EMP, criada anteriormente:

```
ALTER TABLE EMP  
DROP CONSTRAINT EMP_MGR_FK;
```

Removendo a constraint PRIMARY KEY da tabela DEPT e eliminando as FOREIGN KEY, associadas.

```
ALTER TABLE DEPT  
DROP PRIMARY CASCADE
```

Prática

- Construa as Constraints de Integridade/Restrições para o exercício resolvido em sala de aula