

# Banco de Dados I

## Constraint ( Integridade/Restrições)

Prof. Msc. Aparecido Vilela Junior  
[aparecido.vilela@unicesumar.edu.br](mailto: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



# Alterando o nome do Objeto

- Para alterar o nome de uma tabela, view, sequence ou sinônimos execute o comando RENAME.
- RENAME DEPT TO DEPARTAMENTO;

# TRUNCATE

- Remove todas as linhas de uma tabela;
- O comando Rollback não consegue recuperar as linhas removidas, utilizando o TRUNCATE. Para isso deve-se utilizar o comando DELETE.
  - `TRUNCATE TABLE Nome_Tabela;`

# 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

- 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

```
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

<b>CONSTRAINT restrição_nome</b>	Especifica o nome da restrição. Esse parâmetro é opcional. Se omiti-lo o nome será atribuído com um nome padrão formado de SYS_Cnome, onde nome é um número que unicamente identifica a restrição
<b>NULL / NOT NULL</b>	Especifica que a coluna deve ser ou não preenchida. O padrão é NULL
<b>UNIQUE</b>	Especifica que cada linha da tabela deve ter um valor distinto para a coluna. Cada coluna deve ser declarada como NOT NULL e a coluna não deve ser chave primária.

# CONSTRAINT

<b>PRIMARY KEY</b>	A coluna é única e identifica cada linha. A coluna deve ser declarada como NOT NULL e não deve ter uma restrição de UNIQUE. Pode ser composta de várias colunas.
<b>FOREIGN KEY</b>	Identifica as colunas como as chaves estrangeira das tabelas. Deve referenciar a uma chave primária de uma tabela
<b>CHECK</b>	Especifica uma condição que uma coluna deve satisfazer para cada linha que existe na tabela. Pode unicamente referenciar-se com colunas da mesma tabela.



# 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 - 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 – 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

# 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)

# Visualizando CONSTRAINT

- Pode-se visualizar as definições e nomes utilizando uma pesquisa na tabela `USER_CONSTRAINTS`

```
SELECT constraint_name, constraint_type,  
       Search_condition  
FROM user_constraints  
WHERE table_name = 'nome_tabela'
```

# 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
```

# CONSTRAINT - DISABLE

- O comando DISABLE deve ser executado na cláusula ALTER TABLE, para desativar uma constraint
- Aplica-se a opção CASCADE para desabilitar constraint de integridades dependentes
  - ALTER TABLE EMP
  - DISABLE CONSTRAINT EMP\_PK CASCADE

# CONSTRAINT – ENABLE

- Para ativar uma constraint com o status de desabilitada numa tabela, utiliza-se o comando ENABLE.
  - ALTER TABLE EMP
  - ENABLE CONSTRAINT EMP\_PK;
- Uma UNIQUE ou PRIMARY KEY, é automaticamente criada quando habilitada

# CONSTRAINT - CASCADE

- A opção CASCADE é utilizada:
  - Com a cláusula DROP COLUMN
  - Elimina todas as integridades referenciais referentes a primary key e unique key, definidas nas colunas eliminadas
  - Elimina todas as Multi-Colunas Constraint, definidas na coluna eliminada

# CONSTRAINT - CASCADE

```
CREATE TABLE TESTE1 (  
  PK NUMBER PRIMARY KEY,  
  FK NUMBER,  
  COL1 NUMBER,  
  COL2 NUMBER,  
  CONSTRAINT FK_CONSTRAINT FOREIGN KEY (FK) REFERENCES  
    TESTE1,  
  CONSTRAINT CK1 CHECK (PK > 0 AND COL1 > 0),  
  CONSTRAINT CK2 CHECK (COL2 > 0))
```

- Visualize as colunas associadas com os nomes das constraints na view USER\_CONS\_COLUMNS

```
SELECT CONSTRAINT_NAME, COLUMN_NAME  
FROM USER_CONS_COLUMNS  
WHERE TABLE_NAME = 'EMP'
```



# Prática

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

# Exercício 01 (mini-mundo histórico), Apresente o Esquema (tabela / campos / Integridade)

Nome: Pedro Álvares Cabral			Matricula: 1500	
Filiação: Joaquim Álvares Cabral e Maria Álvares Cabral				
Logradouro: Rua Pastel de Belém			Bairro: Novo Recheio	
Nascimento: 05/05/1460		Cidade: Lisboa		País: Portugal
Identidade: 11223344		CPF: 2222222222		
Curso: Navegação pelo Mundo Data de Autorização: em 25/06/1492 N° autorização 1225				
Data Inscrição no Curso: 22/08/1498				
Currículo Pleno				
Disciplina	Situação	Carga Horária	Faltas	Notas
Navegação Básica	Aprovado	80	4	8
Artilharia de Canhão I	Aprovado	80	6	9
Artilharia de Canhão II	Reprovado	80	0	10