

Banco de Dados I

Constraint (Integridade/Restrições)

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

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.

Linguagem de Manipulação de Dados (DML)

- Seleccionando todas as colunas de toda a tabela:
 - ```
SELECT *
FROM emp
```
- Seleccionando colunas especificadas:
  - ```
SELECT empno,  
       ename  
FROM emp
```


- Criar expressões com números, datas usando operadores aritméticos

Operadores	Descrição
+	Somar
-	Subtrair
*	Multiplicar
/	Dividir

- SELECT ename,
 sal,
 10 * sal + 500
FROM emp
- Precedência dos Operadores:
 - Multiplicação e Divisão tem prioridade sobre adição e subtração
 - Operadores da mesma prioridade, são avaliados da esquerda para direita
 - Parênteses são utilizados para forçar a priorização

- Renomeia o título de uma coluna
- Utilizado com colunas calculadas
 - `SELECT name AS nome,
 sal AS salario
FROM emp`
- Ou
- `SELECT name “nome”,
 sal “Salario”
FROM emp`

- Um operador de concatenação:
- Concatena colunas ou caracteres strings para outras colunas
- É representado por duas barras verticais (||)
- Cria uma coluna resultante, que é uma expressão caracter.
 - SELECT ename
|| job
FROM emp

- Ao menos que seja indicado de outra maneira uma consulta numa base de dados poderá trazer linhas duplicadas, para evitar isso utilizamos o comando DISTINCT
- ```
SELECT DISTINCT deptno
FROM emp
```

- As restrições são estabelecidas com o uso da cláusula WHERE.
- Determina que linhas devem ser relacionadas.
- `SELECT * | {DISTINCT} coluna | expressão [alias], .... } FROM table;`
- WHERE condição;
- `SELECT * FROM EMP WHERE SAL > 3000`

- Caracteres e Datas são incluídos entre aspas simples.
- Caracteres são case-sensitive e datas são format sensitive
- O formato default para data é 'DD-MON-RR'.

- Condições de Comparação:

| Operador | Significado      |
|----------|------------------|
| =        | Igual            |
| >        | Maior que        |
| >=       | Maior ou igual a |
| <        | Menor que        |
| <=       | Menor ou igual a |
| <>       | Diferente        |



- Outras Condições de Comparação:

| Operador               | Significado                               |
|------------------------|-------------------------------------------|
| BETWEEN<br>... AND ... | Entre dois valores<br>(inclusive)         |
| IN (set)               | Selecionando uma lista<br>de valores      |
| LIKE                   | Texto dentro de uma<br>determinada coluna |
| IS NULL                | Se o valor é nulo                         |

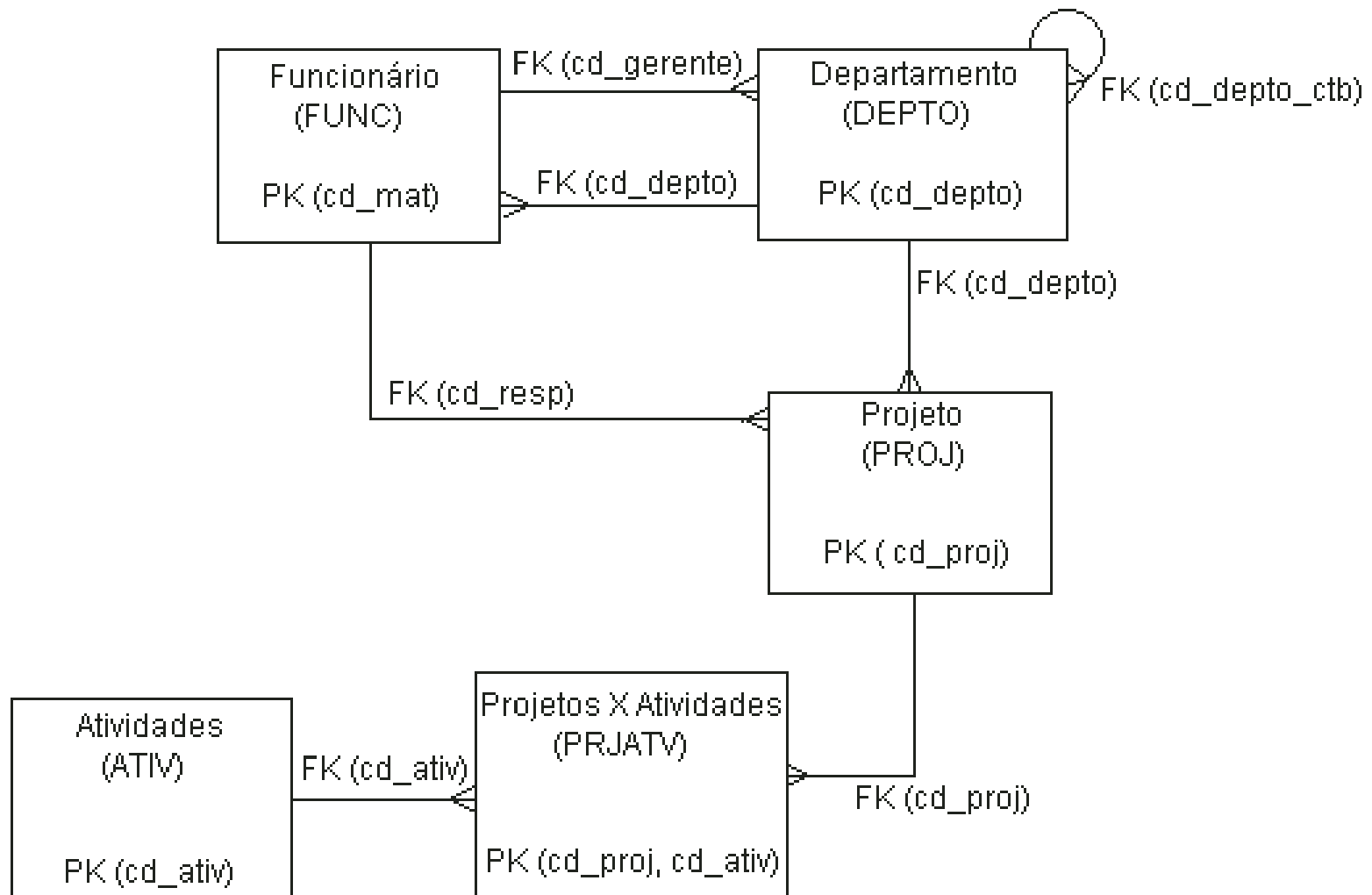
- Condições Lógicas:

| Operador | Significado                                                  |
|----------|--------------------------------------------------------------|
| AND      | Retorna verdadeiro se ambas condições forem verdadeiras      |
| OR       | Retorna verdadeiro se pelo menos uma condição for verdadeira |
| NOT      | Retorna verdadeiro se a condição é falsa.                    |

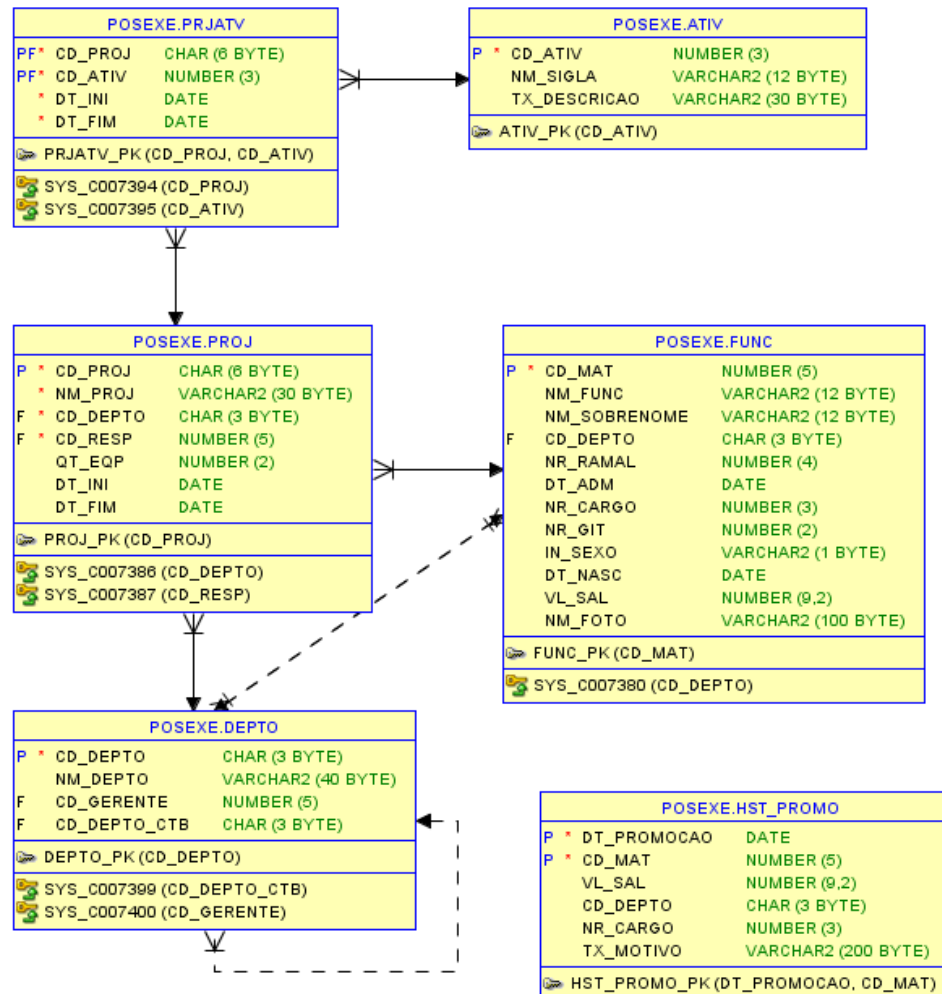
|   |                               |
|---|-------------------------------|
| 1 | Operadores Aritméticos        |
| 2 | Concatenação                  |
| 3 | Condições de Comparação       |
| 4 | IS [NOT] NULL, LIKE, [NOT] IN |
| 5 | [NOT] BETWEEN                 |
| 6 | NOT                           |
| 7 | AND                           |
| 8 | OR                            |

- Permite que as linhas resultantes da consulta sejam apresentadas em uma determinada ordem:
  - ASC – Ordem Ascendente (default).
  - DESC – Ordem Descendente.
- SELECT ename,  
    sal  
FROM emp  
ORDER BY sal

- A tabela Funcionario tem como primary key (PK) a coluna cd\_mat e como foreign key (FK) a coluna cd\_depto, que estabelece relacionamento com a tabela Depto.
- A tabela Departamento tem como primary key (PK) a coluna cd\_depto e como foreign key (FK) a coluna cd\_gerente, que estabelece relacionamento com a tabela Funcionario, e a coluna cd\_depto\_ctb, que estabelece um autorelacionamento.
- A tabela Projeto tem como primary key (PK) a coluna cd\_proj e como foreign key (FK) a coluna cd\_resp, que estabelece relacionamento com a tabela Funcionario, e a coluna cd\_depto, que estabelece relacionamento com a tabela Depto.
- A tabela Atividade tem como primary key (PK) a coluna cd\_ativ e não possui relacionamentos com outras tabelas.
- Finalmente, a tabela Projetos-Atividades (PrjAtv) tem como primary key (PK) as colunas cd\_proj e cd\_ativ, que simultaneamente agem como foreign key (FK) e estabelecem relacionamento com as tabelas Projeto e Atividade, respectivamente.



# Modelo de Dados



- 1) Deseja-se uma lista que contenha número da matrícula, primeiro e último nome e cargo de todos os funcionários do sexo feminino que tenham cargo superior a 55.
- 2) Deseja-se uma lista que contenha número de matrícula, nome e sobrenome do funcionário em cujo sobrenome não seja encontrada a letra “E”.
- 3) Deseja-se uma lista contendo código e nome do projeto para todos os projetos que possuam o texto “AMA” em algum lugar de seu nome. Como restrição adicional, temos que somente os projetos em cujo código existam os números 21 como terceiro e quarto caracter devem ser pesquisados.



- 4) Deseja-se uma lista contendo os códigos de departamento para os departamentos que possuam funcionários. Na listagem-resposta, não deve haver repetição de código de departamento.

- ```
SELECT cd_mat,  
       nm_func,  
       nm_sobrenome,  
       nr_cargo  
FROM   funcionario  
WHERE  in_sexo = 'F'  
       AND nr_cargo > 55;
```

- ```
SELECT cd_mat,
 nm_func,
 nm_sobrenome
FROM funcionario
WHERE nm_sobrenome NOT LIKE '%E%';
```

- ```
SELECT cd_proj,
       nm_proj
FROM   projeto
WHERE  nm_proj LIKE '%AMA%'
       AND cd_proj LIKE '__21%';
```

- ```
SELECT DISTINCT cd_depto
FROM funcionario
WHERE cd_depto IS NOT NULL;
```

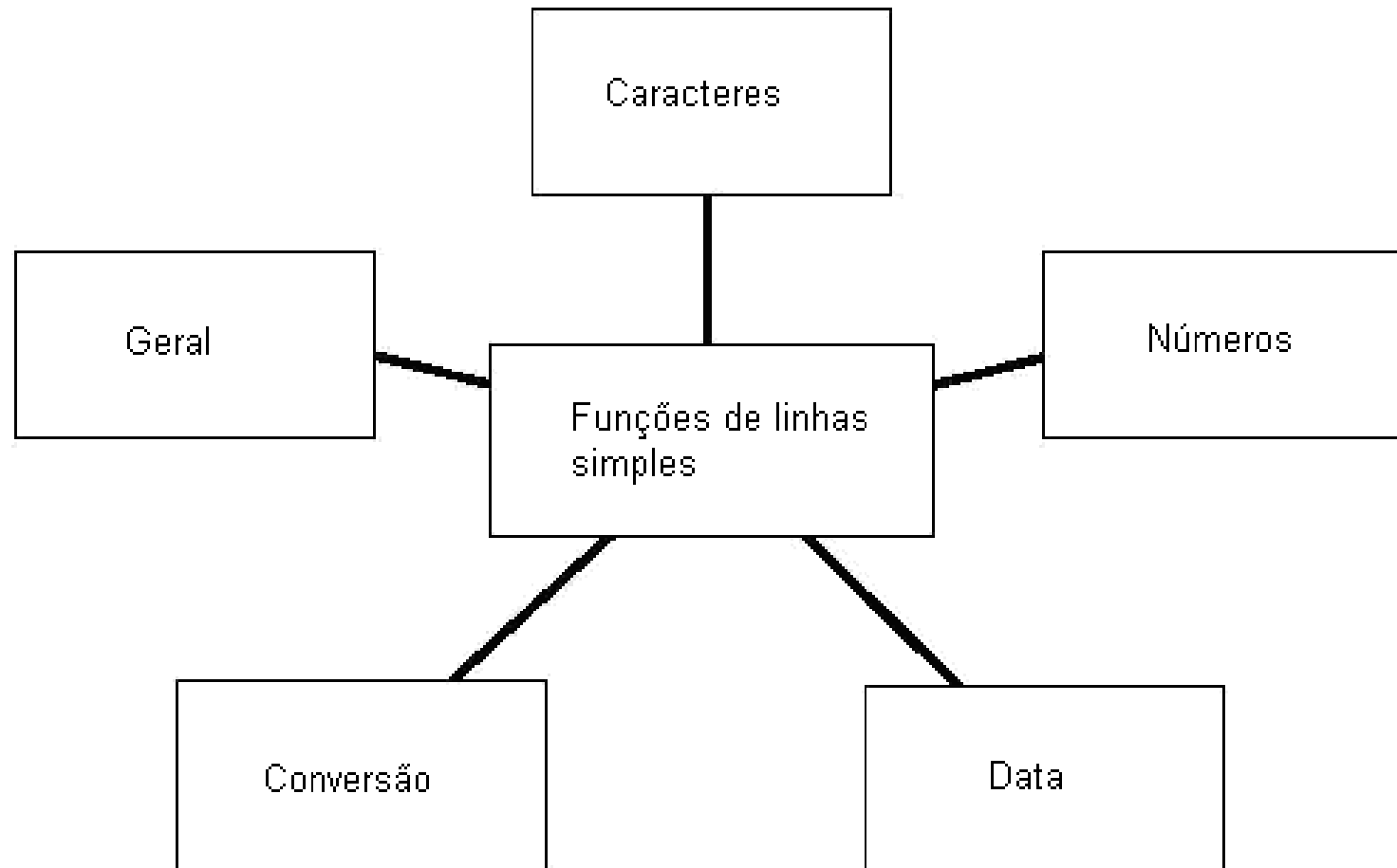
- Programas que têm a finalidade de efetuar cálculos sobre um conjunto de linhas e retornam um valor.
- O argumento para estas funções pode ser o nome de uma coluna ou uma expressão (combinação de nomes de colunas, constantes ou outras funções).

- As funções são uma característica muito poderosa e podem ser usadas para:
  - Executar cálculos em dados;
  - Manipular saída de grupos de linhas
  - Formatar datas e números para exibir
  - Converter os tipos de dados das colunas

- Funções de Manipulação:

| Função              | Resultado |
|---------------------|-----------|
| LOWER ('Curso SQL') | curso sql |
| UPPER ('Curso SQL') | CURSO SQL |
|                     |           |





- Funções de manipulação de strings:

| Função                     | Resultado  |
|----------------------------|------------|
| CONCAT ('Pos Banco')       | PosBanco   |
| SUBSTR('PosBanco',1,3)     | Pos        |
| LENGTH('PosBanco')         | 8          |
| INSTR ('PosBanco','B')     | 4          |
| LPAD (SAL,10,'*')          | *****24000 |
| RPAD (SAL,10, '*')         | 24000***** |
| TRIM ('P' FROM 'PosBanco') | osBanco    |

- Funções Numéricas:

| Funções           | Resultado |
|-------------------|-----------|
| ROUND(45.926 , 2) | 45.93     |
| TRUNC (45.926 ,2) | 45.92     |
| MOD (1600,300)    | 100       |

|       |                          |
|-------|--------------------------|
| YYYY  | Ano numérico completo    |
| YEAR  | Ano soletrado            |
| MM    | 2 dígitos do mês         |
| MONTH | Nome do Mês              |
| MON   | Abreviação do mês        |
| DY    | Abreviação dia da semana |
| DAY   | Nome do dia da semana    |
| DD    | Dia do mês numérico      |