

# Banco de Dados

## SQL Básica

# Introdução

- SQL:
  - *Structured Query Language* –  
Linguagem Estruturada de Consulta
  - Linguagem declarativa
  - Desenvolvida para os bancos de dados relacionais
  - Tornou-se um padrão para banco de dados relacionais
  - Impulsionou não apenas a disseminação dos SGBDR, mas também da própria

# Introdução

- História:
  - Final da década de 1960:
    - Edgar Frank Codd apresentou as primeiras idéias sobre banco de dados relacional
  - 1970:
    - Publicou o artigo “A relational model of data for large shared data banks – Um modelo relacional de dados para grandes bancos de dados compartilhados”
  - 1981:
    - Recebeu o prêmio ACM TURING AWARD

# Introdução

- História: *(cont.)*
  - 1973:
    - IBM criou o seu primeiro gerenciador de dados relacional
      - O SYSTEM R que utilizava a linguagem de consulta SEQUEL (*Structured English Query Language* - Linguagem Inglesa Estruturada de Consulta).
      - Por motivos legais, a sigla foi alterada para SQL
  - 1979:
    - ORACLE disponibiliza comercialmente o primeiro SGBDR

# Introdução

- História: *(cont.)*
  - 1986:
    - SQL-86 – Primeira versão padronizada da SQL
    - Inicialmente desenvolvida no âmbito da ANSI (*American National Standards Institute* – Instituto Nacional Americano de Padrões)
  - 1987:
    - Aprovada pela ISO (*International Organization for Standardization* – Organização Internacional para Padronização)

# Introdução

- História: *(cont.)*
  - 1989:
    - SQL-89 – Extensão do padrão SQL-86
  - 1992:
    - SQL-92 – Aprovada pela ISO
  - 1999:
    - SQL-99 ou SQL3 – Aprovada pela ISO
    - Definidos os usos de *triggers*, *stored procedures*, consultas recursivas, entre outros
    - Definiu regras para os SGBDOR (Sistema de

# Introdução

- História: *(cont.)*
  - 2003:
    - SQL-2003
    - Introduzindo características relacionadas ao XML (*eXtensible Markup Language* - Linguagem de Marcação Extensiva), sequências padronizadas e colunas com valores de auto-generalização.
  - 2008:
    - SQL-2008
    - Trouxe nas especificações formas para a SQL poder ser usada em conjunto com XML

# Introdução

- Grupos:
  - DDL:
    - *Data Definition Language* – Linguagem de Definição de Dados
    - Subconjunto utilizado para criar, alterar e excluir tabelas e elementos associados
  - DML:
    - *Data Manipulation Language* – Linguagem de Manipulação de Dados
    - Subconjunto dos comandos usado para inserir, atualizar e apagar dados



# Introdução

- Outros Grupos:
  - DCL:
    - *Data Control Language* – Linguagem de Controle de Dados
    - Subconjunto de comandos que controla o acesso dos usuários aos dados
  - DTL:
    - *Data Transaction Language* - Linguagem de Transação de Dados
    - Subconjuntos de comandos usados para iniciar e finalizar transações.

# Introdução

- Outros Grupos: *(cont.)*
  - DQL:
    - *Data Query Language* – Linguagem de Consulta de Dados
    - Tem apenas um único comando (*select*) e suas várias cláusulas e opções
    - Permite recuperar os dados de uma ou mais tabelas através de consultas elaboradas como uma descrição do resultado desejado

# DDL Básica

- Comando *CREATE TABLE*:

- Sintaxe:

```
create table Tabela (  
    Atributo Tipo_Dado [Restrição]  
    [ {, Atributo Tipo_Dado [Restrição] } ] );
```

- Exemplo:

```
create table Setor (  
    ID integer not null primary key,  
    Setor varchar(20)  
);
```

# DDL Básica

- Comando *CREATE TABLE*: (cont.)

- Exemplo:

```
create table Funcion (
    ID integer not null,
    Nome varchar(40),
    ID_Setor integer,
    primary key(ID),
    foreign key(ID_Setor) references Setor (ID) on
    delete cascade
);
```

# DDL Básica

- Comando *ALTER TABLE*:

- Sintaxe:

```
alter table Tabela  
add Atributo Tipo_Dado [Restrição]  
[ {, add Atributo Tipo_Dado [Restrição] } ];
```

- Exemplo:

```
alter table Funcion  
add Salario numeric(7, 2),  
add Dt_Nasc date;
```

# DDL Básica

- Comando *ALTER TABLE*: (cont.)

- Sintaxe:

- alter table Tabela
    - alter Atributo type Tipo\_Dado
    - [ {, alter Atributo type Tipo\_Dado } ];

- Exemplo:

- alter table Funcion
    - alter Dt\_Nasc type varchar(10);

- MySQL:

- alter table Funcion
    - modify dt\_nasc varchar(10);

# DDL Básica

- Comando *ALTER TABLE*: (cont.)
  - Sintaxe:

```
alter table Tabela  
drop Atributo  
[ {, drop Atributo } ];
```
  - Exemplo:

```
alter table Funcion  
drop Dt_Nasc;
```

# DDL Básica

- Comando *ALTER TABLE*: (cont.)

- Sintaxe:

- alter table Tabela add primary key( Atributo [, Atributo] ) ;

- Exemplo:

- alter table Funcion add primary key(ID);

- Sintaxe:

- alter table Tabela add foreign key(Atributo) references Tabela\_FK (Atributo\_FK) );

- Exemplo:

- alter table Funcion add foreign key(ID\_Setor) references Setor (ID) ;

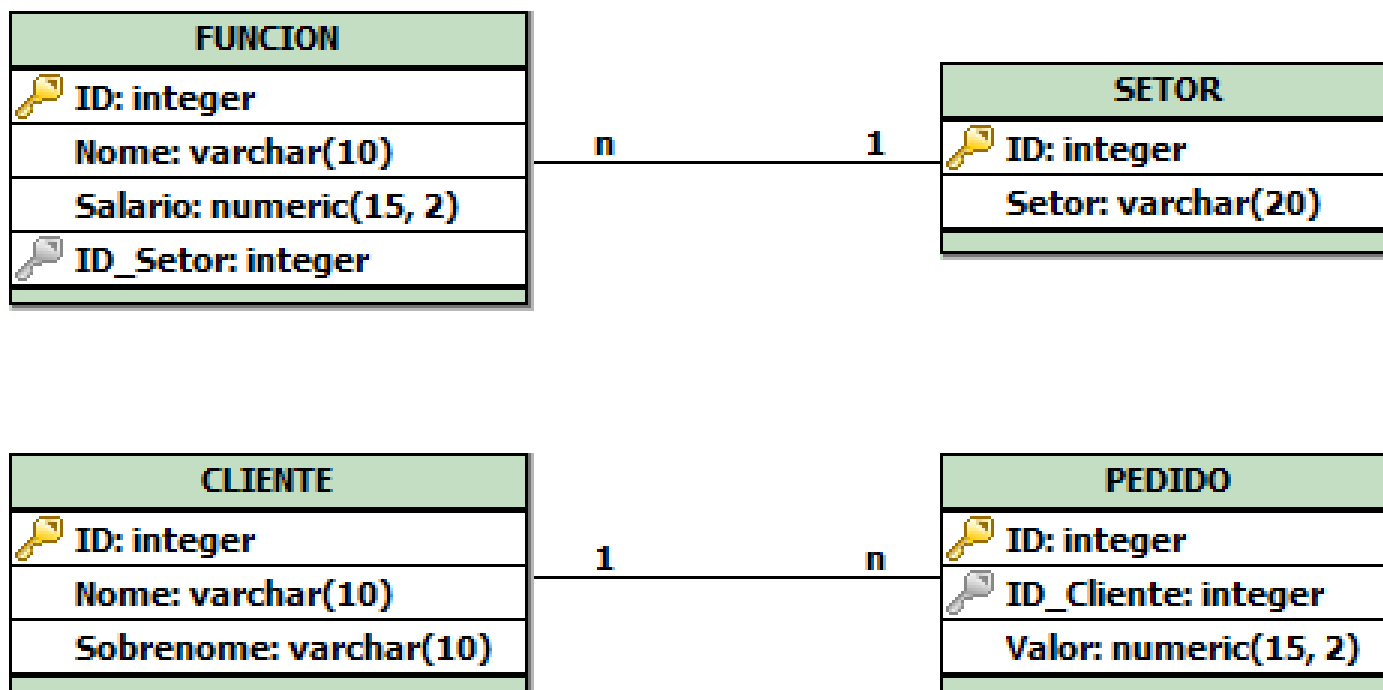


# DDL Básica

- Comando *DROP TABLE*:
  - Sintaxe:  
drop table Tabela;
  - Exemplo:  
drop table Funcion;

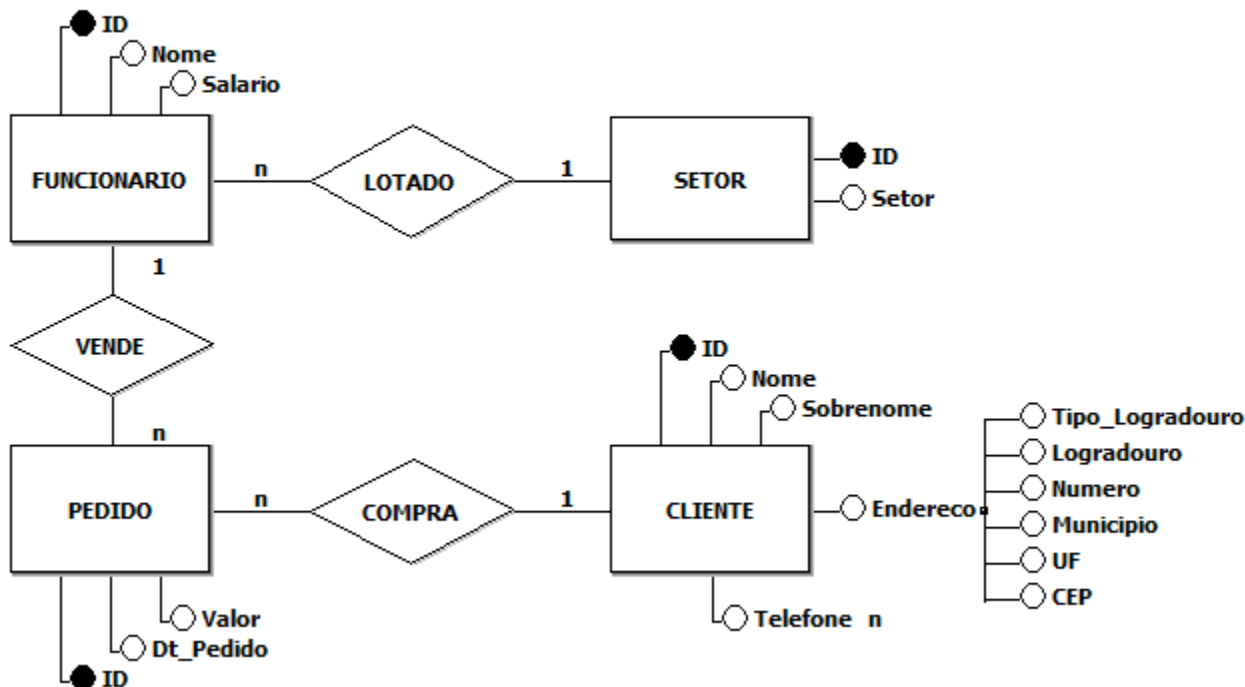
# Exercício #1

- Criar o modelo físico do BD lógico abaixo:



# Exercício #2

- Baseado no exercício #1, alterar o modelo lógico (normalizando) e modelo físico do BD abaixo:



# DML Básica

- Comando *INSERT*:

- Sintaxe:

- insert into Tabela [ (Atributo [ {, Atributo} ] ) ]  
values (Valor [ {, Valor } ] );

- Exemplos:

- insert into Setor (ID, Setor)  
values (1, 'Filho');

- insert into Setor  
values (2, 'Mãe');

# Exercício #3

- Povoar as tabelas criadas no Exercício #2 com os dados apresentados:

Tabela Setor	
ID	Setor
1	Filho
2	Mãe
3	Pai
4	Apostolo
5	Profeta

Tabela Funcionario			
ID	Nome	Salario	ID_Setor
1	José	1.000,00	3
2	Maria	1.500,00	2
3	Jesus	2.000,00	1
4	Lucas	500,00	4
5	Jacob	500,00	4
6	Adriana	250,00	null

Tabela Telefone		
ID	Telefone	ID_Cliente
1	1111-1111	1
2	1111-2222	1
3	2222-3333	2
4	5555-4444	5

Tabela Pedido				
ID	Dt_Pedido	Valor	ID_Funcionario	ID_Cliente
1	01/01/2011	1.000,00	1	1
2	02/02/2011	2.000,00	3	3
3	03/03/2011	3.000,00	6	5
4	04/04/2011	1.500,00	1	3

Tabela Cliente								
ID	Nome	Sobrenome	Tipo_Logradouro	Logradouro	Numero	Municipio	UF	CEP
1	Joana	Silva	Rua	José Marrocos	1010	Juazeiro	CE	63.050-210
2	Fernando	Oliveira	Av.	Carlos Gomes	2020	Crato	CE	63.200-000
3	Marcio	Fernandes	Rua	João Sampaio	30	Trindade	PE	null
4	Josefa	Sampaio	Av.	Leão Sampaio	404	Cajazeiras	PB	null
5	Sarah	Ferreira	Rua	Todos os Santos	50	Juazeiro	CE	null

# DML Básica

- Comando *UPDATE* :

- Sintaxe:

```
update Tabela set Atributo = Valor [ {, Atributo =  
    Valor} ]  
[ where Condição ];
```

- Exemplos:

```
update Funcion set Salario = 1200  
where ID_Setor = 4;
```

```
update Funcion set Salario = Salario * 1.5  
where ID_Setor = 4;
```

# DML Básica

- Comando *DELETE*:

- Sintaxe:

- delete from Tabela  
[ where Condição ];

- Exemplo:

- delete from Setor  
where ID >= 4;

# DML Básica

- Comando *COMMIT*:
  - Sintaxe:  
`commit;`
  - Exemplo:  
`commit;`
- Comando *ROLLBACK*:
  - Sintaxe:  
`rollback;`
  - Exemplo:  
`rollback;`



# Exercício #4

- Atualizar as tabelas povoadas no Exercício #3 com os dados apresentados:

Tabela Função			
ID	Nome	Salario	ID_Setor
1	Tadeu	1500,00	1
2	Ylane	1200,00	2
3	Julian	1000,00	1
4	Ewerton	1000,00	1
5	João	800,00	2
6	Celestino	1500,00	3
7	Maria	500,00	null
8	Joana	1000,00	4
9	Fernanda	1000,00	4

Tabela Setor	
ID	Setor
1	Desenvolvedor
2	Manutenção
3	Financeiro

Tabela Cliente		
ID	Nome	Sobrenome
1	Francisco	Silva
2	José	Lima
3	Maria	Silva
4	Adriana	Ferreira
5	João	Oliveira
6	Eduardo	Souza

Tabela Pedido		
ID	ID_Cliente	Valor
1	2	1000,00
2	4	2000,00
3	2	1500,00
4	5	2500,00
5	2	1000,00

# *SELECT* Básico

- Comando *SELECT* simples:
  - Sintaxe:  
select Atributo [ {, Atributo} ] from Tabela;
  - Exemplos:  
select ID, Nome from Funcion;  
  
select \* from Funcion;

# *SELECT* Básico

- Cláusula *WHERE* com Condições Simples:

- Sintaxe:

- select Atributo [ {, Atributo} ] from Tabela  
[where Condição];

- Operador Comparativo “=”

- Exemplos:

- select ID, Nome from Funcion  
where Nome = 'Tadeu';

# *SELECT* Básico

- Cláusula *WHERE* com Condições Simples: (*cont.*)
  - Operador Comparativo “<>”
    - Exemplos:

```
select ID, Nome from Funcion  
where Nome <> 'Tadeu';
```
  - Operador Comparativo “>”
    - Exemplos:

```
select ID, Nome from Funcion  
where Nome > 'Tadeu';
```

# *SELECT* Básico

- Cláusula *WHERE* com Condições Simples: (*cont.*)
  - Operador Comparativo “>=”
    - Exemplos:  
select ID, Nome from Funcion  
where Nome >= ‘Tadeu’;
  - Operador Comparativo “<”
    - Exemplos:  
select ID, Nome from Funcion  
where Nome < ‘Tadeu’;

# *SELECT* Básico

- Cláusula *WHERE* com Condições Simples: (*cont.*)
  - Operador Comparativo “<=”
    - Exemplos:

```
select ID, Nome from Funcion
where Nome <= 'Tadeu';
```
  - Operador Comparativo “*BETWEEN ... AND ...*”
    - Exemplos:

```
select ID, Nome from Funcion
where Nome between 'João' and 'Tadeu';
```

# *SELECT* Básico

- Cláusula *WHERE* com Condições Simples: (cont.)

- Operador Comparativo “*NOT BETWEEN ... AND ...*”

- Exemplos:

- ```
select ID, Nome from Funcion  
where Nome not between 'João' and 'Tadeu';
```

- Operador Comparativo “*IN (LISTA)*”

- Exemplos:

- ```
select ID, Nome from Funcion  
where Nome in ('João','Tadeu');
```

# *SELECT* Básico

- Cláusula *WHERE* com Condições Simples: (*cont.*)
  - Operador Comparativo “*NOT IN (LISTA)*”
    - Exemplos:

```
select ID, Nome from Funcion
where Nome not in ('João','Tadeu');
```
  - Operador Comparativo “*LIKE*”
    - Exemplos:

```
select ID, Nome from Funcion
where Nome like 'J%';
```



# *SELECT* Básico

- Cláusula *WHERE* com Condições Simples: (cont.)
  - Operador Comparativo “*NOT LIKE*”
    - Exemplos:

```
select ID, Nome from Funcion  
where Nome not like 'J%';
```
  - Operador Comparativo “*IS NULL*”
    - Exemplos:

```
select ID, Nome from Funcion  
where Setor is null;
```

# *SELECT* Básico

- Cláusula *WHERE* com Condições Simples: (*cont.*)
  - Operador Comparativo “*IS NOT NULL*”
    - Exemplos:

```
select ID, Nome from Funcion  
where Setor is not null;
```

# Exercício #5

- Base nas tabelas povoadas no Exercício #4, mostrar os...
  - a) Funcionários que não estão alocado em nenhum setor.
  - b) Clientes que têm a letra “a” em qualquer parte do sobrenome.
  - c) Pedidos com valor superior a R\$ 1.500,00.
  - d) Setores entre “1” e “3”.
  - e) Pedidos emitidos antes do dia 02/02/2011 e depois do dia 03/02/2011.