BDs para alta escabilidade

Geomar A. Schreiner
gschreiner@uffs.edu.br

Roteiro

- Apresentação da disciplina
- Revisão SQL (início)

Apresentação da Disciplina

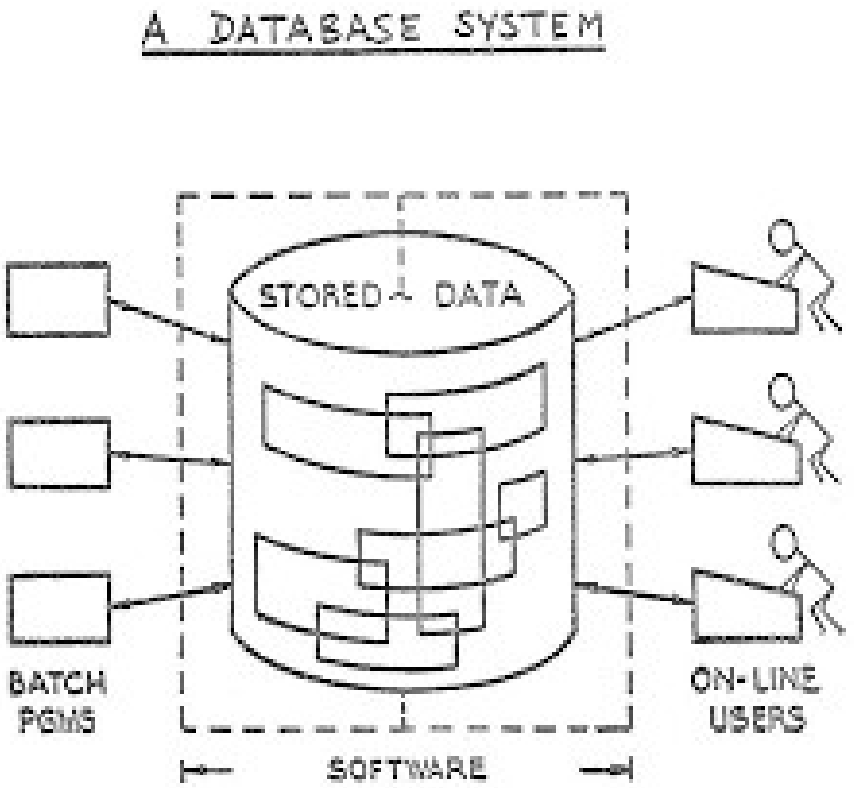
- Encontros SIGAA

Apresentação da Disciplina

Revisão SQL

Histórico

- 1970 foi publicado “A Relational Model of Data Large Shared Banks” que define o modelo relacional;



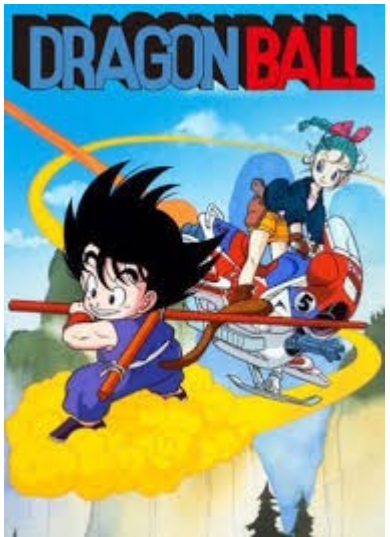
Histórico

- 1970 foi publicado “A Relational Model of Data Large Shared Banks” que define o modelo relacional;
- 1974 é publicada SEQUEL (Structured English Query Language)


```
SELECT      ITEM
FROM        SALES
WHERE       DEPT =
            SELECT      DEPT
            FROM        LOC
            WHERE       FLOOR = '2'
```

Histórico

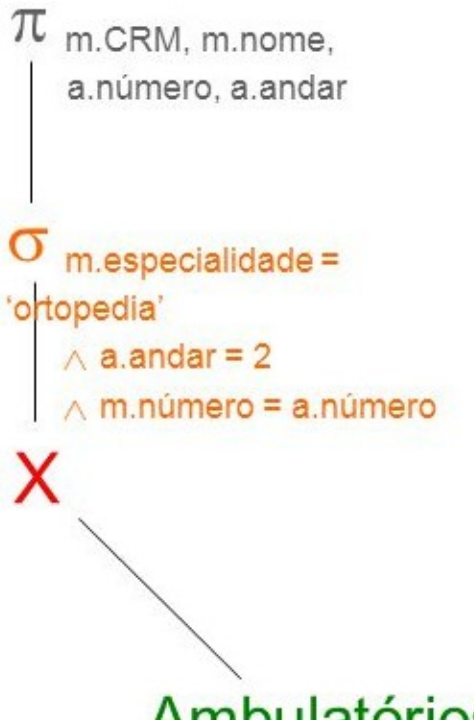
- 1970 foi publicado “A Relational Model of Data Large Shared Banks” que define o modelo relacional;
- 1974 é publicada SEQUEL (Structured English Query Language)
- 1986 SQL vira ISO
  - SQL-2(92); SQL-3(99)



Visão geral

- Possui base pautada na álgebra relacional e no cálculo relacional de tupla

```
select m.CRM, m.nome, a.número, a.andar
from Médicos m, Ambulatórios a
where m.especialidade = 'ortopedia'
and a.andar = 2
and m.número = a.número
```



Visão geral

- Possui base pautada na álgebra relacional e no cálculo relacional de tupla
- Principais funcionalidades:
  - DDL – definição de dados
  - DML – manipulação de dados
  - Restrições de integridade
  - Transações
  - Store Procedures

DDL

- Criação de um BD
  - SQL não define essa operação
  - Alguns BDs (maioria dos comerciais) possuem o comando de criação:
    - **CREATE DATABASE** nome_db
  - Remover um BD
    - **DROP DATABASE** nome_db

DDL

- Comandos para definição do esquema
  - CREATE TABLE
    - Cria uma nova tabela, com suas definições estruturais e de integridade
  - ALTER TABLE
    - Modifica as definições da tabela
      - Atributos chave não podem ser removidos
      - Atributos NOT NULL não podem ser inseridos
  - DROP TABLE
    - Remove uma tabela

DDL

- CREATE TABLE
  - **CREATE TABLE** tabela (

    atributo_1 tipo_1,

    [atributo_n tipo_n,]

    [**PRIMARY KEY** (atributo_, [atributo_n])]

    [**FOREIGN KEY** (nome_atributo) **REFERENCES** nome_tabela]

    );
  - Principais tipos:
    - Int, smallint, numeric (definição e tamanho), char, varchar (definição de tamanho), date, datetime
    - Date → ‘YYYY-MM-DD’

DDL

Name	Aliases	Description
bigint	int8	signed eight-byte integer
bigserial	serial8	autoincrementing eight-byte integer
bit [(*n*)]		fixed-length bit string
bit varying [(*n*)]	varbit	variable-length bit string
boolean	bool	logical Boolean (true/false)
character [(*n*)]	char [(*n*)]	fixed-length character string
character varying [(*n*)]	varchar [(*n*)]	variable-length character string
date		calendar date (year, month, day)
double precision	float8	double precision floating-point number (8 bytes)
integer	int, int4	signed four-byte integer
numeric [(*p*, *s*)]	decimal [(*p*, *s*)]	exact numeric of selectable precision
smallint	int2	signed two-byte integer
text		variable-length character string
time [(*p*)] with time zone	timetz	time of day, including time zone
timestamp [(*p*)] with time zone	timestamptz	date and time, including time zone

DDL

- CREATE TABLE
  - Exemplo
    - CREATE TABLE alunos (
      matricula NUMERIC(10),
      nome VARCHAR(120),
      sexo SMALLINT,
      PRIMARY KEY (matricula)
      );

DDL - Restrições

- Restrições (constraints) são construções previstas no SQL utilizados para definir um nível de controle mais apurado sobre os dados e seu domínio
- Geralmente, restrições podem ser definidas de duas maneiras
  - Através de DDL na construção de tabelas
    - Unique, not null, check
  - Adicionados posteriormente sob demanda
    - constraint

DDL - Restrições

- Checagem (check)
  - Permitem a especificação de valores válidos ou não para o campo
  - Podem referenciar outros campos
  - Avaliam uma expressão boleana

DDL - Restrições

- Exemplo

```
create table Empregados (

codEmp integer,
nome  varchar(40) not null,
CPF  numeric(11) not null unique,
idade integer check (idade between 16 and 90),
estadoCivil char(10) check (estadoCivil in
  ('solteiro', 'casado', 'viúvo', 'desquitado',
  'divorciado')),
salario numeric(8,2) check (salario > 0),
tempoServiço integer,
codGer  integer,
codDepto  integer,
...
```

DDL - Restrições

- Exemplo

```
...
constraint EmpPk primary key (codEmp),
constraint EmpCodg foreign key (codGer)
  references Empregados on update cascade
                           on delete set null,
constraint EmpDept foreign key (codDepto)
  references Departamentos on update cascade
                             on delete no action,
constraint IdadeTS check(tempoServico < idade),
constraint EmpGer check(codEmp < > codGer),
constraint SalarioGerente
     check(salario < (select salario
                      from Empregados e
                      where e.codEmp = codGer))
 );
```

DDL - Restrições

- Exemplo

```
...
constraint EmpPk primary key (codEmp),
constraint EmpCodg foreign key (codGer)
  references Empregados on update cascade
                           on delete set null,
constraint EmpDept foreign key (codDepto)
  references Departamentos on update cascade
                              on delete no action,
constraint IdadeTS check(tempoServico < idade),
constraint EmpGer check(codEmp < > codGer),
constraint SalarioGerente
     check(salario < (select salario
                      from Empregados e
                      where e.codEmp = codGer))

);
```

DDL - Restrições

- Exemplo

```
create table Departamentos (

codDepto integer,
nome varchar(20) check (nome  in ('Vendas',
  'Pessoal', 'Finanças', 'Administrativo')),
Andar integer check (andar between 1 and 10),
orçamento numeric(20,2),

constraint DeptPk primary key (codDepto),
constraint ControleOrç
       check( orçamento > =
                (select sum(salário)
                 from Empregados e
                 where e.codDepto = codDepto))
);
```

DDL - Restrições

- Exemplo

```
create table Departamentos (

codDepto integer,
nome varchar(20) check (nome  in ('Vendas',
  'Pessoal', 'Finanças', 'Administrativo')),
Andar integer check (andar between 1 and 10),
orçamento numeric(20,2),

constraint DeptPk primary key (codDepto),
constraint ControleOrç
        check( orçamento > =
                (select sum(salário)
                 from Empregados e
                 where e.codDepto = codDepto))

);
```

DDL

- DROP TABLE
  - **DROP TABLE** nome_tabela;
  - Exemplo
    - **DROP TABLE** alunos;

DDL

- ALTER TABLE
  - **ALTER TABLE** nome_tabela **ADD**

    COLUMN nome_atributo tipo [, OPP nome_atributo_n tipo_n];
  - **ALTER TABLE** nome_tabela **DROP**

    COLUMN nome_atributo;
  - **ALTER TABLE** nome_tabela **MODIFY**

    COLUMN nome_atributo tipo [, OPP nome_atributo_n tipo_n];
  - **ALTER TABLE** nome_tabela **[ADD | DROP]**

    CONSTRAINT nome_ref tipo_ref;
  - **[ADD|DROP] [PRIMARY KEY ...|FOREIGN KEY ...]**

DDL

- ALTER TABLE
  - Exemplo
    - **ALTER TABLE** Ambulatorios **ADD** nome **VARCHAR(30)**
    - **ALTER TABLE** Medicos **DROP PRIMARY KEY**
    - **ALTER TABLE** Pacientes **DROP COLUMN** doenca, **DROP COLUMN** cidade
    - **ALTER TABLE** Funcionarios **ADD FOREIGN KEY**(nroa) **REFERENCES** Ambulatorios

DDL

- Exemplo (Inclusão de restrições gerais)

  ALTER TABLE departamentos **ADD CONSTRAINT** nome_departamento **CHECK (**nome **IN** (‘Vendas’, ‘Pessoal’, ‘Financas’**))**

  ALTER TABLE fun_dep **ADD CONSTRAINT** unicidade_departamento **UNIQUE** (id_departamento, id_funcionario)

DML

- Utilizada para a manipulação dos dados
- Comandos
  - **INSERT**
  - **UPDATE**
  - **SELECT**
  - **DELETE**
- São instruções declarativas
  - Manipulação de conjuntos
  - Baseia-se no que deve ser feito

DML

- Inserção dos dados
  - **INSERT INTO** nome_tabela **{**(lista_atributos)**}**

    VALUES (lista_valores | (**SELECT ...**));
  - EXEMPLO
    - **INSERT INTO** Ambulatorios **VALUES** (1, 1, 30)
    - **INSERT INTO** Medicos (codm, nome, idade, especialidade, CPF, cidade) **VALUES** (4, ‘Carlos’, 28,’ortopedia’, 11000110000, ‘Joinville’);

DML

- Alteração de dados
  - **UPDATE** nome_tabela SET nome_atributo_1 = Valor_1 [{, nome_atributo_n = Valor_n}]

    [**WHERE** condição]
  - Condição
    - Att (>,<, >=, <=, <>, =) [att | calculo | valor]
    - Condição1 AND condição2 OR condição3 ...

DML

- Alteração de dados
  - **UPDATE** nome_tabela SET nome_atributo_1 = Valor_1 [{, nome_atributo_n = Valor_n}]

    [**WHERE** condição]
  - EXEMPLO
    - **UPDATE** Medicos **SET** cidade = 'Florianopolis'
    - **UPDATE** Ambulatorios **SET** capacidade = capacidade + 5, andar = 3 **WHERE** nroa = 2

DML

- Exclusão de dados
  - **DELETE FROM** nome_tabela

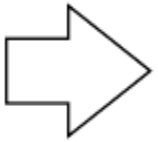
    [**WHERE** condição]
  - EXEMPLO
    - **DELETE** Medicos **WHERE** cidade = 'Florianopolis'

DML - SELECT

- Consulta de dados
  - **SELECT** lista_atributos **FROM** tabelas
    [**WHERE** condição]

- Mapeamento para Álgebra Relacional

select a_1, ..., a_n
from t
where c



$\pi_{a1, ..., an} (\sigma_{c} (t))$

- Ex:
  - **SELECT** * **FROM** Medicos

Exercícios

```
Tipos_Veiculos (codTipo, descricao);
Habilitacoes (codH, tipo, idade_min, descricao);
Veiculos (matricula, nome, modelo, comprimento, potMotor,
vlDiaria, codTipo (TiposVeiculos));
Funcionarios (codF, nome, telefone, endereco, idade, salario)
Veiculos_Habilitacoes (codTipo (Tipos_Veiculos), codH
(Habilitacoes));
Clientes (CPF, nome, endereco, estado_civil, num_filhos,
data_nasc, telefone, codH (Habilitacoes));
Locacoes (codLoc, valor, inicio, fim, obs, matricula
(Veiculos), codF (Funcionarios), CPF (Cliente));
```

Criar o BD conforme as regras disponíveis no portal;
Popular o BD;

Exercícios Part 1

- Popule o BD com os dados da pasta compartilhada
- Faça as seguintes atualizações
  - Um funcionário cadastrou o nome do cliente errado, atualize o nome do cliente com cpf 68745120480 para “João”
  - Mariana (cpf 23548754210) se divorciou, atualize a base de dados
  - O veículo código 103 está com o comprimento errado, o valor correto é 18 metros.
  - Todos os barcos devem sofrer uma alteração em suas diárias. Reajuste em 12.4% todos os valores de diárias
- O funcionário Marquito foi demitido, exclua ele da base.

Exercícios Part 2

1) Listar o nome e o estado civil e a data de nascimento de todos os clientes
2) Listar o nome, idade e telefone de todos os funcionarios
3) Liste as habilitações que necessitam que o usuário possua mais de 25 anos
4) Listar o nome dos veiculos que tem comprimento maior que 10 e com potencia superio a 120
5) Listar o nome e o comprimento de todos os bascos cuja potencia fique entre 50 e 300