

1. O que é a SQL e qual sua vantagem?

A Structured Query Language (SQL) ou Linguagem de Consulta Estruturada, é uma linguagem de consulta para banco de dados relacionais e uma vantagem em relação a outras linguagens é sua forma declarativa, ou seja, uma consulta SQL especifica qual o resultado esperado e não o caminho para atingi-lo.

2. Cite e explique os grupos de instruções da SQL.

- Data Definition Language (DDL): Permite a criação de bancos de dados, de tabelas de banco de dados e outros elementos.
- Data Manipulation Language (DML): Permite inserir, alterar, apagar e consultar dados.
- Data Transaction Language (DTL): Permite o controle de transações em banco de dados.
- Data Control Language (DCL): Permite modificar as permissões dos usuários sobre o acesso aos dados.

3. Descreva como criar e apagar bancos de dados através de instruções DDL.

para criarmos o banco de dados, devemos usar a instrução:

```
CREATE DATABASE
```

E para removermos este banco de dados do SGBD devemos usar a instrução:

```
DROP DATABASE
```

4. Descreva como criar e apagar tabelas através de instruções DDL.

Usamos os comandos:

```
CREATE TABLE nometabela(  
  campo_1 tipo [opções],  
  ...  
  campo_n tipo [opções],  
  [restrição_1,  
  ...  
  restrição_n]);
```

E para apagar:

```
DROP TABLE nometabela;
```

5. Cite e explique os principais tipos de dados presentes na maioria dos SGBD.

| Nome | Tamanho | Descrição | Intervalo |
|------|---------|-----------|-----------|
|------|---------|-----------|-----------|

| | | | |
|---------------|--------------------------|---|--------------------------|
| integer, int, | com o tamanho de 4 bytes | num intervalo de -2, 14x10 ⁹ | a +2, 14x10 ⁹ |
|---------------|--------------------------|---|--------------------------|

real, float4 com o tamanho de 4 bytes num intervalo de -1×10^{37} a 1×10^{37} com 6 dígitos de precisão

double precision, float8 com 8 bytes e -1×10^{307} a 1×10^{308} com 15 dígitos de precisão

boolean com 1 byte

char(n), character(n) Variável com o número de caracteres Tamanho fixo
varchar(n), character varying(n) Variável com o número de caracteres Variável com tamanho máximo

date com 4 bytes e um intervalo de Data 4.713 A.C. a 5.874.897 D.C.

time 8 bytes Hora 00:00:00 a 24:00:00

6. Além do tipo, quais informações podem ser inseridas a respeito de campos na criação de tabelas.

NOT NULL O campo não pode conter valores nulos, ou seja, o valor do campo de ser informado

DEFAULT valor_padrão Valor padrão para o campo, sempre que for inserido um novo registro o valor inicial para o campo será valor_padrão

CHECK verificação Impõe uma verificação sobre os valores do campo, não permitindo valores que não passem pela verificação

7. Sobre as restrições estudadas, explique-as e mostre como criá-las através de instruções DDL.

Para criamos uma chave primária utilizamos as instruções

```
CONSTRAINT nome_chave PRIMARY KEY (c_1, ... c_n)
```

Onde nome_chave é o nome da chave primária e c_1, ... c_n são os campos que formam a chave primária

8. Cite e exemplifique as possíveis ações adicionais que podem ser criadas juntamente com chaves estrangeiras.

ON DELETE (para exclusões) e ON UPDATE (para alterações) inseridas juntamente com a chave estrangeira

ON DELETE e ON UPDATE devem ser seguidas pelas ação a ser executada. As possíveis ações são:

SET NULL Quando o dado referenciado for excluído ou apagado, o dado dependente assumi valor nulo

SET DEFAULT Quando o dado referenciado for excluído ou apagado, o dado dependente assumi seu valor padrão

CASCADE Quando o dado referenciado for excluído ou apagado, o dado dependente a exclusão ou alteração será propagada para o dado dependente

9. Considere as instruções DDL a seguir:

```
CREATE TABLE a(  
  id_a INT NOT NULL,  
  den_a VARCHAR(30) NOT NULL,  
  CONSTRAINT a_pk PRIMARY KEY (id_a)  
);  
CREATE TABLE b(  
  id_b INT NOT NULL,  
  id_a INT NOT NULL,  
  CONSTRAINT b_pk PRIMARY KEY (id_b),  
  CONSTRAINT b_fk_a FOREIGN KEY (id_a) REFERENCES a(id_a) ON DELETE SET NULL  
);
```

Identifique um problema que pode acontecer quando houver manipulação de dados nestas tabelas.

Na Tabela 1 e na Tabela 2 tem os mesmos nomes de identificação (id_a), e pode dar problema depois.

10. Especifique as instruções DDL necessárias para criar um banco de dados para o seguinte esquema relacional:

```
cidade(id_cidade: integer, nome_cidade varchar(80), uf char(2))  
  
funcionario(id_funcionario integer, nome_funcionario varchar(60),  
           endereco varchar(80), bairro varchar(50),  
           *id_cidade integer, salario real, sexo char(1),  
           nascimento date)  
  *funcionario.id_cidade: cidade.id cidade  
  
dependente(*id_funcionario integer, id_dependente integer,  
           nome_dependente varchar(60), sexo char(1), nascimento date,  
           parentesco varchar(40))  
  
departamento(id_departamento integer, nome_departamento varchar(40),  
             *id_gerente integer)  
  *departamento.id_gerente: funcionario.id_funcionario  
  
projeto(id_projeto integer, nome_projeto varchar(40),  
       *id_supervisor integer, *id_departamento integer)  
  *projeto.id_supervisor: funcionario.id_funcionario  
  *projeto.id_departamento: departamento.id_departamento  
  
projeto_funcionario(*id_projeto integer, *id_funcionario integer,  
                   horas_trabalhadas real)  
  *projeto_funcionario.id_funcionario: funcionario.id_funcionario  
  *projeto_funcionario.id_projeto: projeto.id_projeto
```

```

CREATE DATABASE exercício10;
CREATE TABLE cidade (
    Id_cidade INT NOT NULL,
    Nome_cidade VARCHAR(80) NOT NULL,
    Uf CHAR(2) NOT NULL
);
CREATE TABLE funcionario (
    Id_funcionario INT NOT NULL,
    Nome_funcionario VARCHAR(60) NOT NULL,
    endereco VARCHAR(80) NOT NULL,
    bairro VARCHAR(50) NOT NULL,
    id_cidade INTEGER NOT NULL,
    salario REAL NOT NULL,
    sexo CHAR(1) NOT NULL,
    nascimento DATE NOT NULL
);
CREATE TABLE dependente (
    Id_funcionario INT NOT NULL,
    Id_dependente INT NOT NULL,
    Nome_dependente VARCHAR(60) NOT NULL,
    sexo CHAR(1) NOT NULL,
    nascimento DATE NOT NULL,
    parentesco VARCHAR(40) NOT NULL
);
CREATE TABLE departamento (
    Id_departamento INT NOT NULL,
    Nome_departamento VARCHAR(40) NOT NULL,
    Id_gerente INT NOT NULL,
);
CREATE TABLE projeto (
    Id_projeto INT NOT NULL,
    Nome_projeto VARCHAR(40) NOT NULL,
    Id_supervisor INT NOT NULL,
    Id_departamento INT NOT NULL,
);
CREATE TABLE projeto_funcionario (
    Id_projeto INT NOT NULL,
    Id_funcionario INT NOT NULL,
    Horas_trabalhadas REAL NOT NULL
);

```

11. Especifique as instruções DDL necessárias para criar um banco de dados para o seguinte esquema relacional

```

aluno(matricula_aluno integer, nome_aluno varchar(60),
      endereco varchar(80), nascimento date, media real)

professor(id_professor integer, nome_professor varchar(60),
          area varchar(40), titulacao varchar(40))

curso(id_curso integer, nome_curso varchar(60), duracao integer,
      area varchar(40), *id_coordenador integer)
*curso.id_coordenador: professor.id_professor

disciplina(id_disciplina integer, nome_disciplina varchar(60),
           carga_horaria integer)

matricula_aluno(*matricula_aluno integer, *id_disciplina integer,
               nota real)
*matricula_aluno.matricula_aluno: aluno.matricula_aluno
*matricula_aluno.id_disciplina: disciplina.id_disciplina

ministra(*id_disciplina integer, *id_professor integer,
         dia_semana char(3), horario time)
*ministra.id_disciplina: disciplina.id_disciplina
*ministra.id_professor: professor.id_professor

curso_disciplina(*id_curso integer, *id_disciplina integer)
*curso_disciplina.id_curso: curso.id_curso
*curso_disciplina.id_disciplina: disciplina.id_disciplina

```

```

CREATE DATABASE exercicio11;
CREATE TABLE aluno (
    Matricula_aluno INT NOT NULL,
    Nome_aluno VARCHAR(80) NOT NULL,
    endereco VARCHAR(80) NOT NULL,
    nascimento DATE NOT NULL,
    media REAL NOT NULL
);
CREATE TABLE professor (
    Id_professor INT NOT NULL,
    Nome_professor VARCHAR(60) NOT NULL,
    area VARCHAR(40) NOT NULL,
    titulacao VARCHAR(40) NOT NULL,
);
CREATE TABLE curso (
    Id_curso INT NOT NULL,
    Nome_curso VARCHAR(60) NOT NULL,
    duracao INT NOT NULL,
    area VARCHAR(40) NOT NULL,
    Id_cordenador INT NOT NULL,
);
CREATE TABLE disciplina (
    Id_disciplina INT NOT NULL,
    Nome_disciplina VARCHAR(40) NOT NULL,
    Carga_horaria INT NOT NULL,
);
CREATE TABLE matricula_aluno (

```

```

        Matricula_aluno INT NOT NULL,
        Id_disciplina INT NOT NULL,
        Nota REAL NOT NULL
    );
    CREATE TABLE ministra (
        Id_disciplina INT NOT NULL,
        Id_professor INT NOT NULL,
        Dia_semana CHAR(3) REAL NOT NULL,
        Horário TIME NOT NULL
    );
    CREATE TABLE curso_disciplina (
        Id_curso INT NOT NULL,
        Id_disciplina INT NOT NULL,
    );

```

12. Especifique as instruções DDL necessárias para criar um banco de dados para o seguinte esquema relacional:

```

cidade(id_cidade integer, nome_cidade varchar(80), uf char(2),
      numero_habitantes integer)

ponto_turistico(id_ponto_turistico integer, *id_cidade integer,
               nome_varchar(80), tipo varchar(20),
               endereco varchar(80))
    *ponto_turistico.id_cidade: cidade.id_cidade;

companhia_aerea(id_companhia integer, nome_varchar(80),
               tipo_voo varchar(40))

voo(id_voo integer, *id_origem integer, *id_destino integer,
    horario time)
    *voo.id_origem: cidade.id_cidade
    *voo.id_destino: cidade.id_cidade

```

```

CREATE DATABASE exercicio12;
CREATE TABLE cidade (
    Id_cidade INT NOT NULL,
    Nome_cidade VARCHAR(80) NOT NULL,
    Uf CHAR(2)
    Numero_habitantes INT NOT NULL,
);
CREATE TABLE ponto_turistico (
    Id_ponto_turistico INT NOT NULL,
    Id_cidade INT NOT NULL,
    Nome VARCHAR(60) NOT NULL,
    tipo VARCHAR(20) NOT NULL,

```

```

        endereco VARCHAR(80) NOT NULL,
    );
    CREATE TABLE companhia_aerea (
        Id_companhia INT NOT NULL,
        Nome VARCHAR(80) NOT NULL,
        tipo_voo VARCHAR(40) NOT NULL,
    );
    CREATE TABLE voo (
        Id_voo INT NOT NULL,
        Id_origem INT NOT NULL,
        Id_destino INT NOT NULL,
        Horário TIME NOT NULL,
    );

```

13. Especifique as instruções DDL necessárias para criar um banco de dados para o seguinte esquema relacional:

```

equipe(id_equipe integer, nome varchar(60), estado char(2),
       patrocinador varchar(50), tecnico varchar(60))

juiz(id_juiz integer, nome varchar(60), idade integer)

jogador(id_jogador integer, nome varchar(60), posicao varchar(30),
        salario real, numero_camisa integer, *id_equipe integer)
    *jogador.id_equipe: equipe.id_equipe

partida(id_partida integer, *id_equipe_local integer,
        *id_equipe_visitante integer, data date, cidade varchar(60),
        *id_juiz integer)
    *partida.id_equipe_local: equipe.id_equipe
    *partida.id_equipe_visitante: equipe.id_equipe
    *partida.id_juiz: juiz.id_juiz

gol(id_gol integer, *id_partida integer, *id_equipe integer,
    *id_jogador integer, minutos integer)
    *gol.id_partida partida.id_partida
    *gol.id_equipe equipe.id_equipe
    *gol.id_jogador jogador.id_jogador

```

```

CREATE DATABASE exercicio13;
CREATE TABLE equipe (
    Id_equipe INT NOT NULL,
    Nome VARCHAR(60) NOT NULL,
    Estado CHAR(2) NOT NULL,
    patrocinador VARCHAR(50) NOT NULL,
    tecnico VARCHAR(60) NOT NULL,
);
CREATE TABLE juiz (
    Id_juiz INT NOT NULL,

```

```

        Nome VARCHAR(60) NOT NULL,
        Idade INT NOT NULL,
    );
CREATE TABLE jogador (
    Id_jogador INT NOT NULL,
    Nome VARCHAR(60) NOT NULL,
    posicao VARCHAR(30) NOT NULL,
    salario REAL NOT NULL,
    numero_camisa INT NOT NULL,
    Id_equipe INT NOT NULL,
);
CREATE TABLE partida (
    Id_partida INT NOT NULL,
    Id_equipe_local INT NOT NULL,
    Id_equipe_visitante INT NOT NULL,
    Data DATE NOT NULL,
    Cidade VARCHAR(60) NOT NULL,
    Id_juiz INT NOT NULL,
);
CREATE TABLE gol (
    Id_gol INT NOT NULL,
    Id_partida INT NOT NULL,
    Id_equipe INT NOT NULL,
    Id_jogador INT NOT NULL,
    Minutos INTEGER NOT NULL
);

```

14. Especifique as instruções DDL necessárias para criar um banco de dados para o seguinte esquema relacional:

```

fazenda(codigo_fazenda integer, nome_fazenda varchar(60),
        municipio varchar(60))

animal(codigo_animal integer, raca varchar(60), nome varchar(60),
        data_nascimento date)

vacina(codigo_vacina integer, nome_vacina varchar(60))

vacinacao(*codigo_animal integer, *codigo_vacina integer, data date,
        lote integer)
        *vacinacao.codigo_animal: animal.codigo_animal
        *vacinacao.codigo_vacina: vacina.codigo_vacina

```

```

CREATE DATABASE exercicio14;
CREATE TABLE fazenda (
    Código_fazenda INT NOT NULL,
    Nome_fazenda VARCHAR(60) NOT NULL,
    Municipio VARCHAR(60) NOT NULL,

```



```

);
CREATE TABLE animal (
    Código_animal INT NOT NULL,
    raca VARCHAR(60) NOT NULL,
    nome VARCHAR(60) NOT NULL,
    data_nascimento DATE NOT NULL,
);
CREATE TABLE vacina (
    Código_vacina INT NOT NULL,
    Nome_vacina VARCHAR(60) NOT NULL,
);
CREATE TABLE vacinacao (
    Código_animal INT NOT NULL,
    Código_vacina INT NOT NULL,
    Data DATE NOT NULL,
    lote INT NOT NULL
);

```

15. Faça as seguintes alterações no banco de dados do exercício 10:

(a) Inclua o campo CPF na tabela funcionario;

```
ALTER TABLE funcionario ADD cpf CHAR(11);
```

(b) Inclua uma restrição para que não existam funcionários com CPF duplicados;

```
ALTER TABLE funcionario ADD CONSTRAINT cpf UNIQUE (cpf);
```

(c) Inclua uma restrição para que um funcionário gerencie apenas um departamento.

```
ALTER TABLE funcionario ALTER media SET DEFAULT 0.0;
```

16. Faça as seguintes alterações no banco de dados do exercício 11:

(a) Inclua o campo CPF na tabela aluno;

```
ALTER TABLE aluno ADD cpf CHAR(11);
```

(b) Inclua uma restrição para que não existam alunos com CPF duplicados;

```
ALTER TABLE aluno ADD CONSTRAINT cpf UNIQUE (cpf);
```

(c) Altere o campo media da tabela aluno para que não aceite valores nulos e o seu valor padrão seja 0.0.

```
ALTER TABLE aluno ALTER media SET DEFAULT 0.0;
```

17. Faça as seguintes alterações no banco de dados do exercício 12:

(a) Inclua uma tabela para armazenar os tipos de pontos turísticos identificados por um código;

```
ALTER TABLE tipos ADD codigo CHAR(11);
```

- (b) Modifique a tabela ponto_turistico para que referencie os tipos de pontos turísticos da tabela anterior através de uma chave estrangeira (elimine os campos que não serão mais utilizados).

ALTER TABLE codigo ALTER media SET DEFAULT 0.0;

18. Faça as seguintes alterações no banco de dados do exercício 13:

- (a) Inclua uma tabela com para armazenar os técnicos das equipes;
ALTER TABLE tipos ADD tecnicos CHAR(60);

- (b) Modifique a tabela equipe para que referencie os técnicos da tabela anterior através de uma chave estrangeira (elimine os campos que não serão mais utilizados).

ALTER TABLE tecnicos ALTER media SET DEFAULT 0.0;

19. Faça as seguintes alterações no banco de dados do exercício 14:

- (a) Inclua uma tabela com para armazenar os municípios das fazendas;
ALTER TABLE tipos ADD municipios CHAR(60);

- (b) Modifique a tabela fazenda para que referencie os municípios da tabela anterior através de uma chave estrangeira (elimine os campos que não serão mais utilizados).

ALTER TABLE tipos ADD tabela_fazenda CHAR(60);

20. Faça as seguintes alterações em todos os bancos de dados anteriores:

- (a) Substitua o campo idade das tabelas de todos os exercícios anteriores pelo campo data_nascimento;
ALTER TABLE tipos ADD data_nascimento CHAR(60);

- (b) Atribua uma sequência aos campos que podem ser incrementados automaticamente.

...