

Trabalho de Banco De Dados

Participantes: Luan Freire, Camilly Chagas, Marcio Filho, Nicolly Gabrielle
Data: 13/12/2021

Fazer um relatório sobre os comandos da linguagem SQL seguindo como base os comandos do site w3schools.

Comandos Exigidos no relatório:

1. create database
 2. create table
 3. drop database
 4. SQL Constraints
 5. SQL PRIMARY KEY
 6. SQL SELECT
 7. SQL WHERE
 8. SQL AND, OR and NOT Operators
 9. SQL ORDER BY Keyword
 10. SQL INSERT INTO
 11. SQL UPDATE Statement
 12. SQL DELETE
 13. SQL COUNT(), AVG() and SUM()
-

1. CREATE DATABASE:

- O Comando 'CREATE DATABASE' tem o encargo de criar um novo banco de dados com um nome dado.

- Exemplo > {

```
CREATE DATABASE EscolaExemplo;
```

```
}
```

- Assim será Criado um Banco de Dados com o nome 'EscolaExemplo'

2. CREATE TABLE:

- O Comando 'CREATE TABLE' tem a função de criar uma tabela dentro de um banco de dados

- Exemplo > {

```
CREATE TABLE Funcionarios(  
    ID int PRIMARY KEY AUTO_INCREMENT,  
    NOME varchar(35) not null,  
    CARGO varchar(20),  
    SALARIO smallint DEFAULT not null  
);  
  
}
```

- Assim será criado uma tabela 'Funcionários' dentro do banco de dados EscolaExemplo com informações de pré-requisitos como o Nome, o Salário e o Cargo.

3. DROP DATABASE:

- Esse Comando tem a função de Excluir um Banco de Dados
- Para demonstrar, iremos supor que eu queira excluir o banco de dados 'EscolaExemplo' que criamos anteriormente.

- Exemplo > {

```
DROP DATABASE EscolaExemplo;
```

```
}
```

- Assim, tanto os bancos de dados quanto às tabelas criadas foram excluídas.

4. SQL Constraints:

- Constraints são pequenas restrições setadas pelo desenvolvedor na hora da criação do banco de dados, isso impede que dados inválidos sejam incluídos em tabelas
- Exemplo - {

```
USE EscolaExemplo;
```

```
CREATE TABLE Funcionarios(  
    ID int PRIMARY KEY AUTO_INCREMENT,  
    NOME varchar(35) not null,  
    CARGO varchar(20),  
    SALARIO smallint DEFAULT not null  
);
```

```
ADD CONSTRAINT CHK_FUNCIONARIO_SALARIO;  
CHECK( SALARIO > 1000 );  
  
}
```

5. SQL PRIMARY KEY:

- Primary Key é, no SQL um Parâmetro a ser passado dentro de uma variável na tabela que indica que aquele valor a ser inserido é único e exclusivo e, dentro do banco de dados não poderá ter mais valor como ele.
- Exemplo > {

```
CREATE TABLE Alunos (  
  
    cpf int PRIMARY KEY NOT NULL,  
    matricula int not null auto_increment,  
    NOME varchar(50) NOT NULL,  
  
    IDADE int NOT NULL
```

);}

6. SQL SELECT:

- O Comando SELECT é usado para extrair valores de uma tabela, sendo útil para consultar uma única variável setada em específico.

- Exemplo - {

```
USE EscolaExemlo;
```

```
SELECT cpf FROM Alunos;
```

```
}
```

7. SQL Where:

- O Comando serve para filtrar valores em específico e fazê-los serem selecionados mais facilmente.

- Exemplo > {

```
SELECT * FROM Funcionarios
```

```
Where NOME = "Guilherme";
```

```
}
```

8. SQL AND, OR and NOT Operators:

- AND:

- O Comando AND permite o uso de mais uma especificação quando se usa a condição Where.

- Exemplo {

SELECT * FROM Funcionarios

Where NOME = "Wagner" AND CARGO = "Professor";

}

- OR:
- O Comando OR faz o Contrario do comando AND, Invez de selecionar variáveis que atendem as duas especificações ele seleciona uma variável que possua uma, ou outra
- Exemplo > {

SELECT * FROM Funcionarios

Where NOME = "Wagner" OR NOME = "Guilherme";

}

- NOT:
- O Comando NOT impede que certas variáveis sejam selecionadas caso possua um valor atribuído no WHERE.
- Exemplo > {

SELECT * FROM Funcionarios

Where NOT CARGO = "Professor";

}

9. SQL ORDER BY:

- O Comando ORDER BY Funciona como um método para organizar os valores selecionados
- Exemplo > {

```
SELECT NOME, IDADE FROM Alunos
```

```
ORDER BY IDADE, NOME;
```

```
}
```

10. SQL INSERT INTO:

- O Comando INSERT INTO tem a função de inserir valores em colunas de tabelas, com base em parâmetros passados.
- Exemplo > {

```
USE EscolaExemplo;
```

```
INSERT INTO ALUNOS (CPF, NOME, IDADE)
```

```
VALUES (01234567801, 'Luan Freire', 17);
```

```
INSERT INTO ALUNOS (CPF, NOME, IDADE)
```

```
VALUES (01234567802, 'Camilly Chagas', 17);
```

```
INSERT INTO ALUNOS (CPF, NOME, IDADE)
```

```
VALUES (01234567803, 'Nicolly Gabrielle', 17);
```

```
INSERT INTO ALUNOS (CPF, NOME, IDADE)
```

```
VALUES (01234567804, 'Marcio Filho', 17);
```

}

11. SQL UPDATE Statement:

- O Comando UPDATE Statement tem a função de alterar dados de uma TABLE
- Exemplo > {

UPDATE Alunos

SET IDADE = NASCIMENTO, matricula = idAluno;

}

12. SQL DELETE:

- O Comando SQL DELETE Serve para deletar uma ou mais linhas de uma tabela, sem alterar sua estrutura no Banco de Dados
- Exemplo > {

USE EscolaExemplo

DELETE IDADE FROM Alunos;

}

13. SQL COUNT(), AVG() and SUM():

- COUNT:
- A Função COUNT regressa o numero de linhas de um valor inserido em uma coluna
- Exemplo > {

SELECT COUNT(NOME)

FROM Alunos

WHERE idAluno = 0;

}

- AVG:
- Essa função apresenta a média aritmética dos valores selecionados

- SUM:
- Essa função apresenta a soma dos valores selecionados.