FIESC SENAI

A FORÇA DA INDÚSTRIA CATARINENSE

Banco de Dados

07 – SQL DML

Prof. André Ulisses
andre.ulisses@edu.sc.senai.br

INTRODUÇÃO SQL

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

SQL (Structured Query Language)

- Linguagem de manipulação de dados.
- Linguagem standard para os sistemas de bases de dados relacionais.
- Adotada como padrão para BDs relacionais (média 95% do mercado)

SQL: Structured Query Language (Linguagem de Consulta Estruturada)

Linguagem comercial para BD relacional

- Padrão ANSI e ISO desde a década de 80: SQL-1 (86); SQL-2 (92); SQL-3 (99); SQL-4 (03)
- Embora padronizado pelo ANSI e ISO, o SQL possui muitas variações e extensões produzidos pelos diferentes fabricantes de SGBDs.

Base

- Álgebra relacional e cálculo relacional

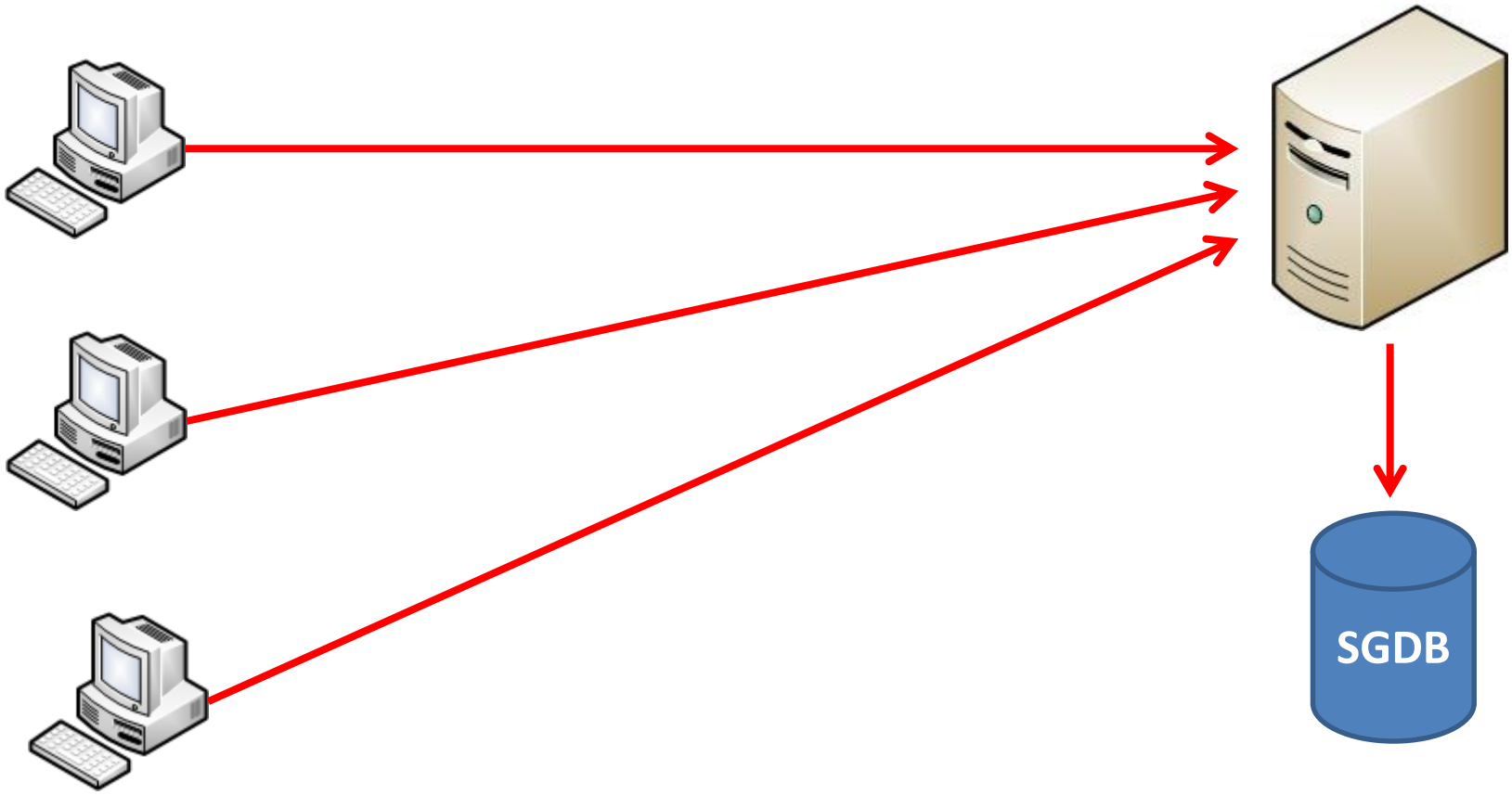
SQL: Structured Query Language (Linguagem de Consulta Estruturada)

Funcionalidades principais

- **DDL**: Linguagem de Definição de Dados
- **DML**: Linguagem de Manipulação de Dados
- **DQL**: Linguagem de Consulta de Dados
- **DCL**: Linguagem de Controle de Dados
- **DTL**: Linguagem de Transação de Dados

APLICAÇÃO CLIENTE-SERVIDOR

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE



INTRODUÇÃO SQL

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

A linguagem SQL engloba numa única linguagem todos os recursos necessários para a manipulação da Base de Dados.

- Linguagem de Definição de Dados (DDL): inclui comandos para definir, alterar e remover tabelas e índices;

- Linguagem de Manipulação de Dados (DML): inclui comandos para inserir, remover, atualizar e consultar os dados armazenados nas tabelas;

SQL - MANIPULAÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

DML: Linguagem de Manipulação de Dados – Principais comandos.

- INSERT
- DELETE
- UPDATE
- SELECT

Inserção de registros

```
INSERT INTO nome_tabela (nome_campo_a, nome_campo_b,
..., nome_campo_n) VALUES (valor_campo_a,
valor_campo_b, ..., valor_campo_n)
```

Inserção de registros

Exemplo;

```
CREATE TABLE CLIENTE (
      IDCLIENTE INT PRIMARY KEY AUTO_INCREMENT,
      NOME VARCHAR(10),
      SEXO CHAR(1),
      IDADE INT,
      CIDADE VARCHAR(20)
);

INSERT INTO CLIENTE ( NOME, IDADE) VALUES ('JOÃO',12)
```

Inserção de registros

Atenção:

Qualquer valor que possa ir em coluna do tipo VARCHAR, CHAR, DATE, TIME, DATETIME, TIMESTAMP ou BLOB deve ficar entre aspas simples.

Os valores devem ser listados exatamente na mesma ordem que os nomes das colunas.

Inserção de registros

Atenção:

É possível omitir a lista de nome de colunas, mas os valores devem estar todos ai, e na mesma ordem que adicionou as colunas.

É possível deixar algumas colunas de fora do INSERT, desde que não sejam obrigatórias.

Apagando registros

O comando DELETE serve para apagar dados da tabela. Esse comando utiliza a cláusula WHERE para restringir os registros que serão excluídos;

```
DELETE FROM nome_tabela WHERE condição
```

Exemplo;

```
DELETE FROM CLIENTE WHERE IDCLIENTE = 1
```

WHERE

Operadores para a cláusula WHERE:

=	Igual
<>	Diferente
>	Maior que
<	Menor que
>=	Maior e igual à
<=	Menor e igual à
IN	Lista
LIKE	Que contenha
NOT	Negação
IS NULL	Somente Valores nulos
BETWEEN	Entre valores

Alterando dados

O comando UPDATE serve para atualizar uma ou mais colunas, atribuindo valores novos. Esse comando utiliza a cláusula WHERE para restringir os registros que serão alterados;

```
UPDATE nome_tabela SET nome_campo = valor
WHERE condição;
```

Exemplo;

```
UPDATE CLIENTE SET SEXO = 'F' WHERE IDCLIENTE = 7 AND
SEXO IS NULL;
```

Alterando dados

Para alterar mais de uma coluna utilize a vírgula.

```
UPDATE nome_tabela SET
      nome_campo_1 = valor_1,
      nome_campo_2 = valor_2,
WHERE condição;
```

Exemplo;

```
UPDATE CLIENTE SET SEXO = 'F', IDADE = 21
WHERE IDCLIENTE = 7;
```

WHERE

Operadores para a cláusula WHERE:

=	Igual
<>	Diferente
>	Maior que
<	Menor que
>=	Maior e igual à
<=	Menor e igual à
IN	Lista
LIKE	Que contenha
NOT	Negação
IS NULL	Somente Valores nulos
BETWEEN	Entre valores

SQL - CONSULTA DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Consultando dados

```
SELECT * FROM nome_tabela
```

Exemplo;

```
SELECT * FROM CLIENTE
```

codigo	nome	idade	sexo
1	joão	18	M
2	Maria	17	F
3	Carlos	18	M
4	Alberto	21	M
5	Sonia	21	F
6	Renata	22	F
7	Karina	19	f

Consultando dados - WHERE

A cláusula WHERE indica ao SQL que deve procurar algo especifico. Ela limita os resultados, exibe somente as linha que são compatíveis com a condição estabelecida;

```
SELECT * FROM nome_tabela WHERE nome_campo operador valor
```

Exemplo;

```
SELECT * FROM CLIENTE WHERE CODIGO = 1
```

Consultando dados - WHERE

Operadores para a cláusula WHERE:

=	Igual
<>	Diferente
>	Maior que
<	Menor que
>=	Maior e igual à
<=	Menor e igual à
IN	Lista
LIKE	Que contenha
NOT	Negação
IS NULL	Somente Valores nulos
BETWEEN	Entre valores

Consultando dados - WHERE

Operador OR e AND

É possível utilizar os operadores lógicos OR e AND para combinar mais de uma condição na cláusula WHERE;

Consultando dados - WHERE

Operador OR e AND

```
SELECT * FROM CLIENTE
WHERE
        (IDADE >= 18 AND SEXO = 'M') OR
        (IDADE >= 21 AND SEXO = 'F')
```

codigo	nome	idade	sexo
1	joão	18	M
3	Carlos	18	M
4	Alberto	21	M
5	Sonia	21	F
6	Renata	22	F

Consultando dados – Group By

O cláusula GROUP BY serve para agrupar os resultado em uma consulta SQL;

FUNCIONARIO	NOME	DEPARTAMENTO
1	João	administração
2	Maria	marketing
3	Marcia	marketing
4	Ana	financeiro
5	Lucas	financeiro
6	Antonio	logistica

```
SELECT
      FUNCIONARIO
      , NOME
      , DEPARTAMENTO
FROM;
```

COUNT(F.FUNCIONARIO)	DEPARTAMENTO
1	administração
2	financeiro
1	logistica
2	marketing

```
SELECT
            COUNT(FUNCIONARIO)
            , DEPARTAMENTO
FROM
            FUNCIONARIO
GROUP BY
            DEPARTAMENTO;
```

Consultando dados – Group By e suas Funções

```
COUNT        Retorna a quantidade de registros com
             valores não-NULL diferentes
AVG          Retorna o valor médio
MIN          Retorna o menor valor
MAX          Retorna o maior valor
SUM          Retorna a soma dos valores
GROUP_CONCAT Retorna os valores agrupados, concatenados
```

Consultando dados - Ordenação

```
SELECT * FROM CLIENTE ORDER BY NOME
```

codigo	nome	idade	sexo
4	Alberto	21	M
3	Carlos	18	M
1	joão	18	M
7	Karina	19	NULL
2	Maria	17	F
6	Renata	22	F
5	Sonia	21	F

Consultando dados - Ordenação

ASC - Ascendentemente

```
SELECT * FROM CLIENTE ORDER BY idade
SELECT * FROM CLIENTE ORDER BY idade ASC
```

codigo	nome	idade	sexo
2	Maria	17	F
1	joão	18	M
3	Carlos	18	M
7	Karina	19	NULL
4	Alberto	21	M
5	Sonia	21	F
6	Renata	22	F

Consultando dados - Ordenação

DESC - Descendentemente

```
SELECT * FROM CLIENTE ORDER BY idade DESC
```

codigo	nome	idade	sexo
6	Renata	22	F
4	Alberto	21	M
5	Sonia	21	F
7	Karina	19	NULL
1	joão	18	M
3	Carlos	18	M
2	Maria	17	F

Consultando dados - Ordenação

Múltiplas colunas

```
SELECT * FROM CLIENTE ORDER BY IDADE DESC, SEXO, NOME
```

codigo	nome	idade	sexo
6	Renata	22	F
5	Sonia	21	F
4	Alberto	21	M
7	Karina	19	NULL
3	Carlos	18	M
1	joão	18	M
2	Maria	17	F

Consultando dados – Limitando o resultado

LIMIT quatidade

```
SELECT * FROM CLIENTE ORDER BY NOME LIMIT 5
```

codigo	nome	idade	sexo
4	Alberto	21	M
3	Carlos	18	M
1	joão	18	M
7	Karina	19	NULL
2	Maria	17	F

Consultando dados – Limitando o resultado

LIMIT inicio, quatidade

```
SELECT * FROM CLIENTE ORDER BY NOME LIMIT 4,3
```

codigo	nome	idade	sexo
4	Alberto	21	M
3	Carlos	18	M
1	joão	18	M
7	Karina	19	NULL
2	Maria	17	F
6	Renata	22	F
5	Sonia	21	F

codigo	nome	idade	sexo
2	Maria	17	F
6	Renata	22	F
5	Sonia	21	F

A contagem inicia à partir do 0

Pseudônimos para Tabelas e Campos

Podemos criar pseudônimos para os nomes das tabelas assim fica mais fácil fazer referência, o mesmo pode acontecer com o nome das colunas. Para isso devemos utilizar a palavra reservada “AS” logo após o nome da tabela ou o nome da coluna. Pseudônimos para tabelas também são chamados de nomes correlacionais;

Pseudônimos para Tabelas e Campos

Exemplo:

```
SELECT NOME AS NM, IDADE FROM CLIENTE AS CL;
```

NOME	IDADE
joão	18
Maria	17
Carlos	18
Alberto	21
Sonia	21
Renata	22
Karina	19

NM	IDADE
joão	18
Maria	17
Carlos	18
Alberto	21
Sonia	21
Renata	22
Karina	19