

Banco de Dados

Introdução ao SQL

Profa. Fabrícia Damando

fdamando@yahoo.com.br

Introdução

- Projeto de BD
 - Nível conceitual
 - Nível lógico
 - Nível físico



Modelagem Física: demonstra como os dados são fisicamente armazenados.

Modelo Conceitual: O modelo conceitual é **um diagrama em blocos** que demonstra todas as relações entre as entidades, suas especializações, seus atributos e auto-relações.

Modelo Lógico: O modelo lógico mostra as ligações entre as tabelas de banco de dados, as chaves primárias, os componentes de cada uma, etc.

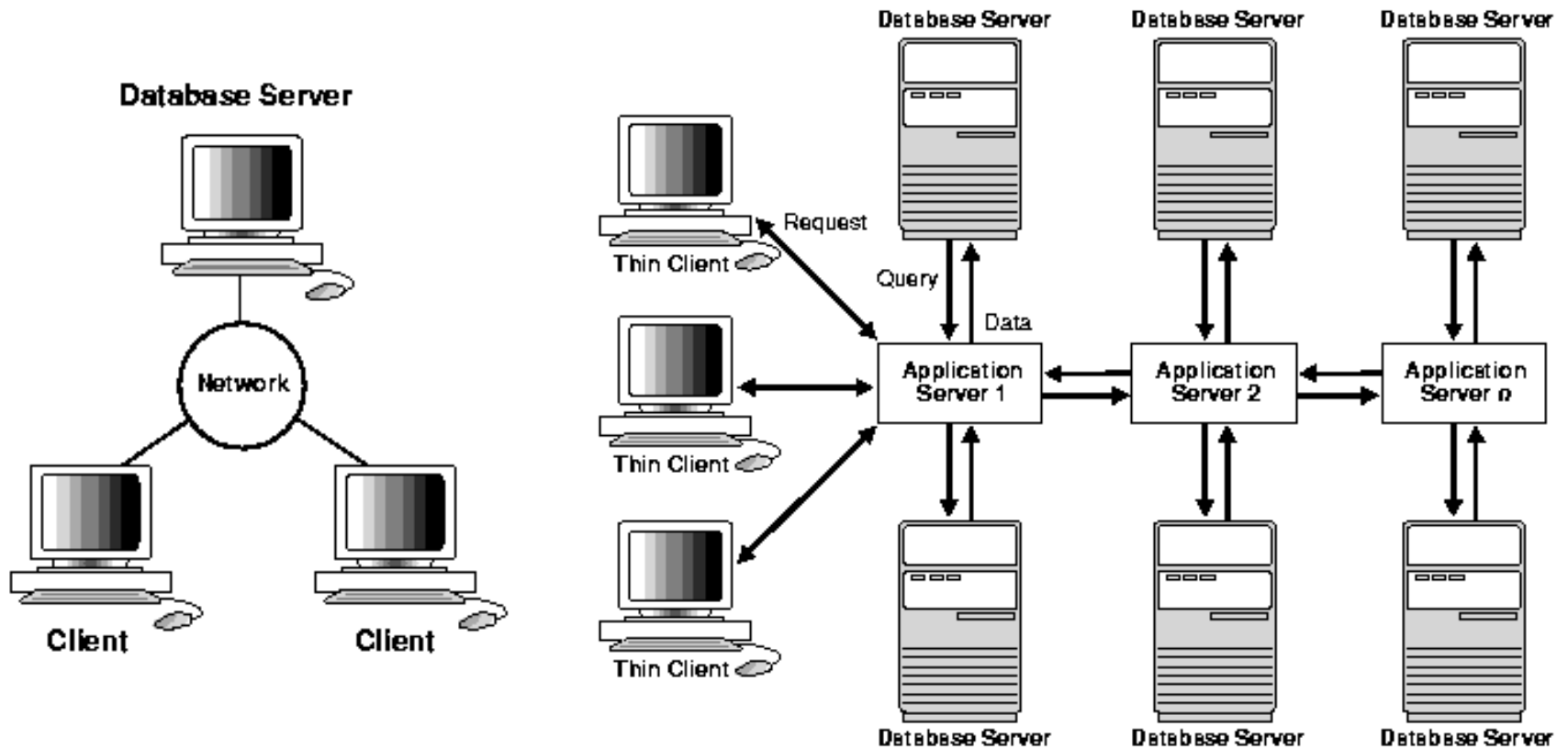
Modelo Físico: Inclui a análise das características e recursos necessários para armazenamento e manipulação das estruturas de dados (estrutura de armazenamento, endereçamento, acesso e alocação física), sendo uma sequência de comandos executados em SQL a fim de criar as tabelas, estruturas e ligações projetadas até então e finalmente criar o banco de dados

Introdução

- Em uma arquitetura cliente/servidor as aplicações e o sistema gerenciador de bancos de dados são separados em duas partes:
 - um *front-end* ou *parte cliente* e
 - um *back-end* ou *parte servidor* — daí a origem do termo.
- O cliente roda a aplicação que acessa informações do banco de dados e interage com os usuários finais através dos seus dispositivos de saída.
- O servidor roda o software do SGBD e trata de capturar e tratar essas requisições quanto a concorrência, compartilhamento, acesso físico e entrega dos dados solicitados ao cliente.

Arquitetura Cliente Servidor

Arquitetura Multicamada



- Arquitetura multicamada
 - um (ou mais) servidor de aplicação provê dados para os clientes e serve como uma interface entre clientes e servidores de bancos de dados
- Possibilita o uso de um servidor de aplicações para:
 - Validar as credenciais de um cliente (como um web browser);
 - Executar rotinas de validação sobre os dados enviados ao servidor de banco de dados;
 - Estabelecer conexão com o servidor de banco de dados;
 - Submeter requisições ao servidor de banco de dados e encaminhar a resposta ao cliente solicitante.

Conectando cliente ao BD

- Ocorre através de um protocolo de comunicação utilizado pelo cliente e reconhecido pelo SGBD.
- No servidor de banco de dados, normalmente, há um serviço cuja função é “escutar” (conhecido por *listener*) uma **porta do servidor** e verificar se chegou alguma requisição que respeite o protocolo de comunicação com o banco de dados nela.
- Uma vez que a **requisição seja reconhecida** esse serviço inclui essa requisição a um processo do servidor de banco de dados que irá acompanhar todo o ciclo de vida dessa requisição através do sistema gerenciador do banco de dados até a devolução dos dados requisitados ao cliente

Padrão ANSI/SPARC - Características de um SGBD

- **Independência física:** o nível físico pode ser alterado independentemente do nível conceitual. Isso significa que o usuário não pode ver todos os componentes de hardware do banco de dados, e que se trata de uma estrutura transparente para representar as informações armazenadas.

Independência lógica: o nível conceitual deve poder ser alterado sem mudar o nível físico, ou seja, o administrador do banco de dados deve poder introduzir melhorias sem afetar a experiência dos usuários.

Facilidade de uso: as pessoas que não estão familiarizadas com o banco de dados devem poder descrever a sua consulta sem fazer referência aos elementos técnicos do banco de dados.

Acesso rápido: o sistema deve poder responder às consultas o mais depressa possível, o que requer algoritmos de busca rápidos.

- **Administração centralizada:** o SGBD deve permitir que o administrador manipule os dados, insira elementos e verifique a sua integridade de maneira centralizada.

Redundância controlada: o SGBD deve poder evitar a redundância de dados sempre que possível, tanto para minimizar os erros quanto para prevenir o desperdício de memória.

Verificação da integridade: os dados devem ser coerentes entre eles e, quando alguns elementos fazem referência a outros, estes últimos devem estar presentes.

Compartilhamento dos dados: o SGBD deve permitir que múltiplos usuários acessem, simultaneamente, o banco de dados.

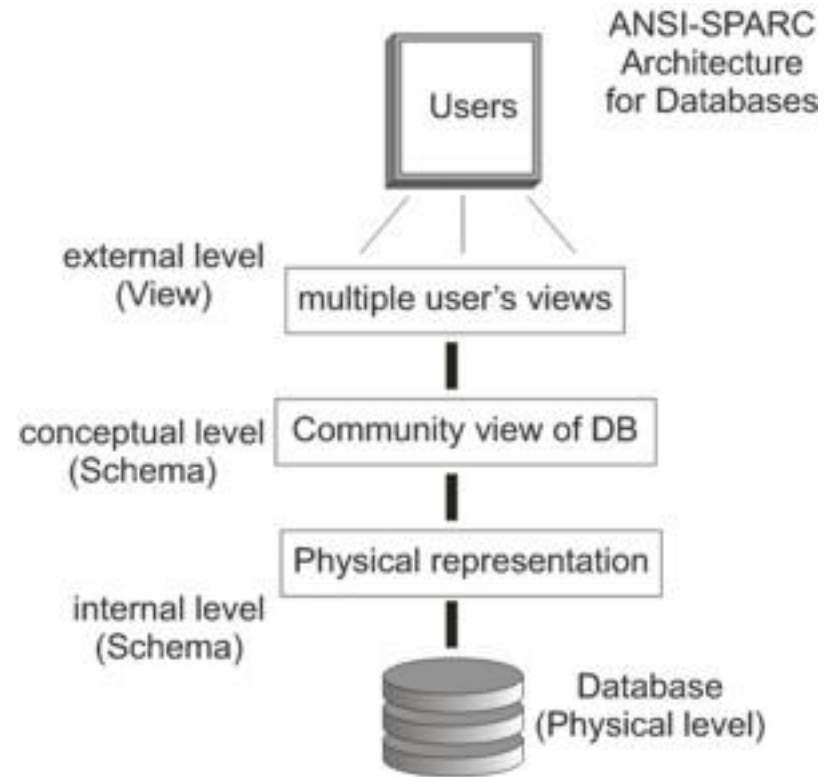
Arquitetura ANSI/SPARC

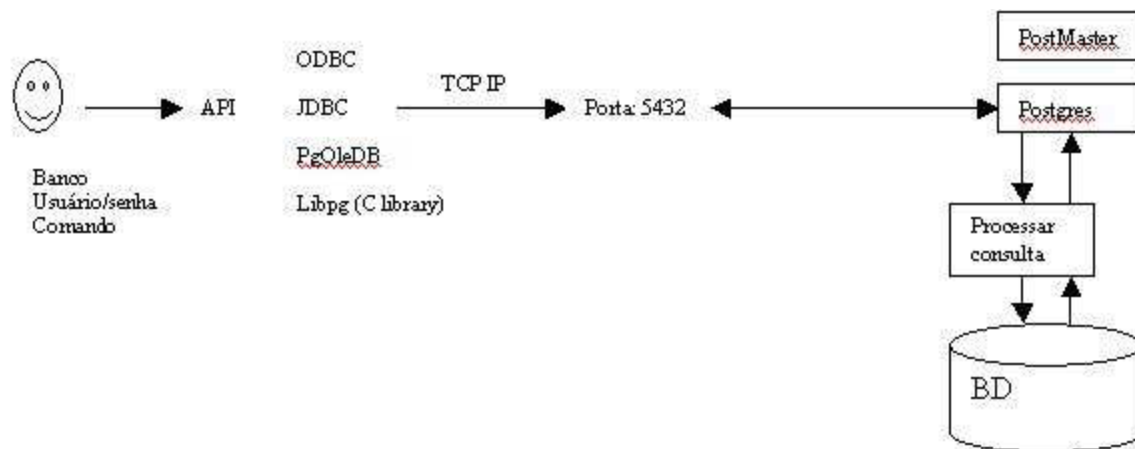
É um padrão de SGBD

Comitê de Padrões e Planejamento de Padrões , é um padrão de design abstrato para um Sistema de Gerenciamento de Banco de Dados

Possui 3 níveis :

- **externo** (visualizações de dados),
- **conceitual** (descreve itens de dados e seus relacionamentos, restrições de integridade),
- **interno** (definições de registros, apresentação de campos)





DDL e DML

- **DDL- Linguagem de Definição de Dados:**

- fornece comandos para definições de esquemas de relação, criação/remoção de tabelas, criação de índices e modificação de esquemas
 - CREATE, ALTER, DROP

- **DML: Linguagem de Manipulação de Dados.**

- Usada para se ter acesso aos dados armazenados no BD. Inclui uma linguagem de consulta baseada na álgebra relacional e cálculo relacional de tupla. Compreende comandos para inserir, consultar, remover e modificar tuplas num BD.
 - INSERT, UPDATE, DELETE

SQL - (Structured Query Language)

A interface SQL é, portanto, implementada em todos os sistemas de bancos de dados relacionais existentes.

- O modelo relacional encontra-se padronizado pela indústria de informática. Ele é chamado de padrão
- O padrão SQL define precisamente uma interface SQL para a definição de tabelas, para as operações sobre as mesmas (seleção, projeção, junção, e outras) e para a definição de regras de integridade de bancos de dados.

- SQL (Structured Query Language): desenvolvida pela IBM (70) como parte do sistema System R.
- A SQL foi inicialmente chamada de SEQUEL
- É a linguagem de consulta padrão para os SGBDR's
- Já existem padrões propostos: ANSI-SQL(SQL-89), SQL-92 e padrões mais recentes: SQL:1999 e SQL:2003.

Tipos de dados

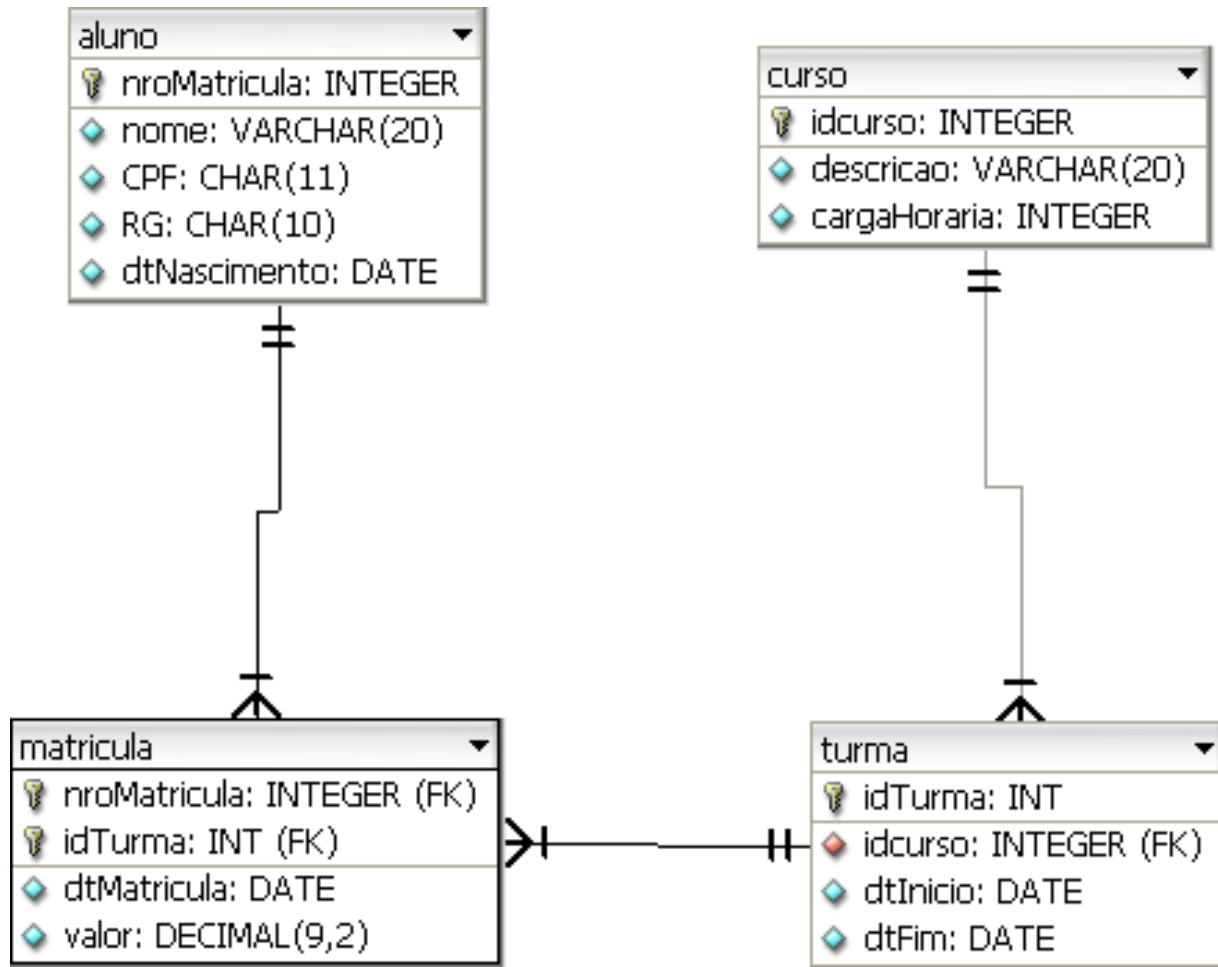
<i>Tipo</i>	<i>Descrição</i>
<i>Char(n) ou character</i>	String de tamanho fixo n
<i>Varchar(n) ou</i> <i>Character varying</i>	String de tamanho variável com tamanho máximo n
<i>int ou integer</i>	Número inteiro
<i>Smallint</i>	Inteiro pequeno
<i>Numeric(p,d)</i>	Número ponto fixo com precisão definida pelo usuário. Com p indicando o número de dígitos e d indicando dos p dígitos os que estão à direita do ponto decimal
<i>real,</i> <i>double precision</i>	Ponto flutuante e ponto flutuante de precisão dupla
<i>Float(n)</i>	Ponto flutuante com precisão definida pelo usuário
<i>Date</i>	Data: aaaa/mm/dd
<i>Time</i>	Hora: hh:mm:ss

BD – MySQL

SQL - Structured Query Language

1. vamos utilizar nas primeiras aulas o MySQL
2. cada aluno tem uma base própria
3. Utilizar o front-end: mysql query browser
4. Criar o banco do próximo slide

Banco para estudos



Comando básicos SQL – Sintaxe

Comando CREATE

- Criar tabelas:

```
CREATE TABLE <nome_da_tabela> (  
    <nome_do_campo> tipo_de_dado [NULL / NOT NULL]  
    [DEFAULT valor_padrao], ...  
)
```

- Nome da tabela – nome da tabela a ser criada
- Nome do campo – nome pelo qual o campo será referenciado
- Tipo de dado – tipo de dado para o campo
- NULL / NOT NULL – define se o campo pode aceitar valores nulos ou não
- DEFAULT – define valor padrão para inserção na tabela. Esse valor será utilizado se nenhum valor para este campo for informado.

Criação de Tabelas

```
CREATE TABLE `uergs`.`Curso` (  
  `idCurso` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  `descricao` VARCHAR(45) NOT NULL,  
  `carga_horaria` INTEGER UNSIGNED NOT NULL,  
  PRIMARY KEY (`idCurso`)  
)  
ENGINE = InnoDB;
```

```
CREATE TABLE `uergs`.`Aluno` (  
  `nroMatricula` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(45) NOT NULL,  
  `CPF` CHAR(11) NOT NULL,  
  `RG` CHAR(10) NOT NULL,  
  `dtNascimento` DATE NOT NULL,  
  PRIMARY KEY (`nroMatricula`)  
)  
ENGINE = InnoDB;
```

```
CREATE TABLE `uergs`.`Turma` (  
  `idTurma` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  `dtInicio` DATE NOT NULL,  
  `dtFim` DATE NOT NULL,  
  `idCurso` INTEGER UNSIGNED NOT NULL,  
  PRIMARY KEY (`idTurma`),  
  CONSTRAINT `idCurso` FOREIGN KEY `idCurso` (`idCurso`)  
    REFERENCES `curso` (`idCurso`)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT  
)  
ENGINE = InnoDB;
```

```
CREATE TABLE `uergs`.`Matricula` (  
  `nroMatricula` INTEGER UNSIGNED NOT NULL DEFAULT NULL  
    AUTO_INCREMENT,  
  `idTurma` INTEGER UNSIGNED NOT NULL,  
  `dtMatricula` DATE NOT NULL,  
  `valor` DECIMAL(9,2) NOT NULL,  
  CONSTRAINT `nroMatricula` FOREIGN KEY `nroMatricula` (`nroMatricula`)  
    REFERENCES `aluno` (`nroMatricula`)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT,  
  CONSTRAINT `idTurma` FOREIGN KEY `idTurma` (`idTurma`)  
    REFERENCES `turma` (`idTurma`)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT
```

- NOT NULL
 - Os campos criados com essa cláusula devem ser obrigatoriamente preenchidos
 - Caso não for, ocorrerá um erro na execução do insert

- O que o UNSIGNED?
 - todos os tipos inteiros no MySQL podem ter o atributo opcional UNSIGNED.
 - Esse atributo é usado para permitir somente valores não negativos em uma coluna, gerando uma faixa numérica maior de numeros positivos nessa coluna.
 - Por exemplo, uma coluna do tipo INT:
Faixa: de -2147483648 a 2147483647
Faixa UNSIGNED: de 0 a 294967295

Fonte: <http://www.rogeralmeida.com.br/sql/mysql-para-que-servem-unsigned-e-zerofill>

- O que é o ZEROFILL?
 - O atributo opcional ZEROFILL, preenche espaços em uma coluna, substituindo por zeros.
 - Por exemplo, uma coluna declarada como INT(4) ZEROFILL, o valor 5 é recuperado como 0005.
 - Se você especificar para uma coluna numérica ZEROFILL, automaticamente o MySQL adiciona o atributo UNSIGNED a coluna.

Comando INSERT

- Responsável por incluir dados em uma tabela:
 - INSERT INTO <nome_tabela> VALUES (valor1, valor2, ..., valorx);
- ou
- INSERT INTO <nome_tabela> (nome_campo1, nome_campo2, ..., nome_campox) VALUES (valor1, valor2, ..., valorx);
- Obs:
 - Valores numéricos não devem ser delimitados por aspas
 - Dados do tipo char, varchar, date devem ser delimitados por aspas

Os valores digitados devem estar na mesma ordem dos campos da tabela

Exemplo

- insert into curso VALUES (1, 'ads',72)

Comando DELETE

- Serve para excluir um ou mais registros de determinada tabela
- DELETE FROM <nome_tabela> WHERE <condicoes>
- OBS:
 - Se a cláusula where não for utilizada, todos os registros da tabela serão excluídos

Exemplo

- delete from curso where idCurso=4
 - Outro exemplo
- delete from curso where carga_horaria=72

Comando UPDATE

- Realiza alterações nos valores dos registros de determinada tabela
- Pode alterar um ou mais registros simultaneamente
- Pode-se alterar uma tabela inteira
- `UPDATE <nome_tabela> SET campo1=valor1, campo2=valor2 WHERE <condicoes>`

- update curso set descricao='teste_atualizacao' where carga_horaria=72
- update curso set descricao='teste2' where idCurso=2

DROP TABLE

- Exclui uma tabela inteira do BD
- Ao excluir uma tabela exclui todo o seu conteúdo
- `DROP TABLE curso`

DROP DATABASE

- Exclui todo um banco de dados
- É preciso ter a permissão do administrador para isso
- Para o PostgreSQL é possível excluir um BD por meio do comando do Linux – `destroydb <nome_banco>`

Consultas - SELECT

- Permite selecionar todas as linhas de uma ou mais tabelas, ou apenas parte delas
- `SELECT <lista de campos> FROM <tabela> WHERE <condicoes>`
- OBS:
 - * lista todos os campos
 - Se o where não for usado, serão mostrados todos os registros da tabela determinada

Exemplo

- `SELECT * FROM curso`
- `Select descricao from curso`
- `Select descricao from curso where carga_horaria=72`
- `SELECT * FROM curso where carga_horaria > 80`

SELECT

- O comando Where deve ser seguido por uma ou mais condições
- As condições podem conter os operadores de comparação (>, <, = , etc) e operadores lógicos AND e OR
- `select noem_produto, preco from produtos where cod_categoria=4 AND preco>50 AND preco <= 500`

SELECT LIKE

- Cláusula utilizada para selecionar registros com início com alguma letra, ou fim com determinada letra, ou até mesmo que possuam determinado caracter, por exemplo.
- EX:
 - Select * from veiculos where adicionais LIKE '%ar condicionado%'

- Exemplos

- Selecionar todos os cursos que na descrição terminem com a palavra sistemas

- Select * from curso where descricao like '%sistemas'
- Indepente do conteúdo a esquerda

- Selecionar todos os curso que possuem na descrição engenharia, independente do conteúdo a direita, ou que possuam em sua descrição a palavra engenharia

- Select * from curso where descricao like 'engenharia%'

-

COUNT

ORDER BY

- COUNT

- Para contar quantos registros existem

- `SELECT count(*) FROM curso where ch < 100`

- ORDER BY

- Para ordenar uma seleção

- `SELECT * FROM curso order by descricao`

- Para ordenar de forma decrescente ou crescente

- `Select * from curso order by descricao DESC`

- `Select * from curso order by descricao ASC`

DISTINCT

- Apresenta valores distintos
- Se tivermos vários registros, com os mesmos dados em um determinado campo, essa cláusula retorna somente os valores distintos
- `Select distinct (cidade) from usuarios`
- `SELECT distinct ch FROM curso`
 - Retornará somente os valores existentes de carga horária, sem repetição

Exercício

- Primeiro inserir em cada tabela pelo menos 20 registros
 - Inserir os cursos: ENG. COMPUTAÇÃO, SISTEMAS EMBARCADOS, AUTOMAÇÃO, PEDAGOGIA
 - Cadastrar data de matrículas do ano de 2017 e 2018, carga horária de 500, 800, 3200 e 4000 horas.
- Fazer as seguintes operações
 - Listar todos os alunos que são do curso ENG. COMPUTAÇÃO
 - Listar todos os alunos que são do curso SISTEMAS EMBARCADOS
 - Listar todos os alunos que são do curso PEDAGOGIA
 - Listar todos os alunos que são do curso AUTOMAÇÃO
 - Listar todos os curso que iniciaram em MARÇO e terminaram em JULHO
 - Listar todos as matriculas realizadas no ano de 20 e que tenham sido superiores a R\$1000,00
 - Listar todas as turmas do curso de SISTEMAS EMBARCADOS
 - Listar quantas turmas tem no curso de SISTEMAS EMBARCADOS
 - Alterar a carga horária dos cursos com menos de 800 horas para 880 horas
 - Listar quais as cargas horárias existentes no cursos
 - Quantos alunos estão frequentando cursos acima de 1000 horas

Referências

- Dica para leitura:
 - <http://www.devmedia.com.br/sqlmagazine/>
- **Tutoriais do MySql**
 - <http://www.mysql.com.br/>
 - <http://www.analysisandsolutions.com/code/mybasic.htm>
 - http://painel.hospedagemweb.com.br/suporte/linux/mysql/manual/manual_to_c.html
 - <http://www.castelobranco.br/projetos/linux/documentacao/mysql.txt>
- **Tutoriais do Postgre**
 - [PostgreSQL Programmer's Guide](#)
 - [PostgreSQL Tutorial](#)
 - [Guia do usuário do PostgreSQL 7.3.4](#)
-

Gerar dados automaticamente

- www.generatedata.com/jogodavelhageneration