

Aula de SQL – Comandos DDL

André Luiz de Oliveira

Eduardo Raul Hruschka

E-mail:

andre_luiz@icmc.usp.br

erh@icmc.usp.br

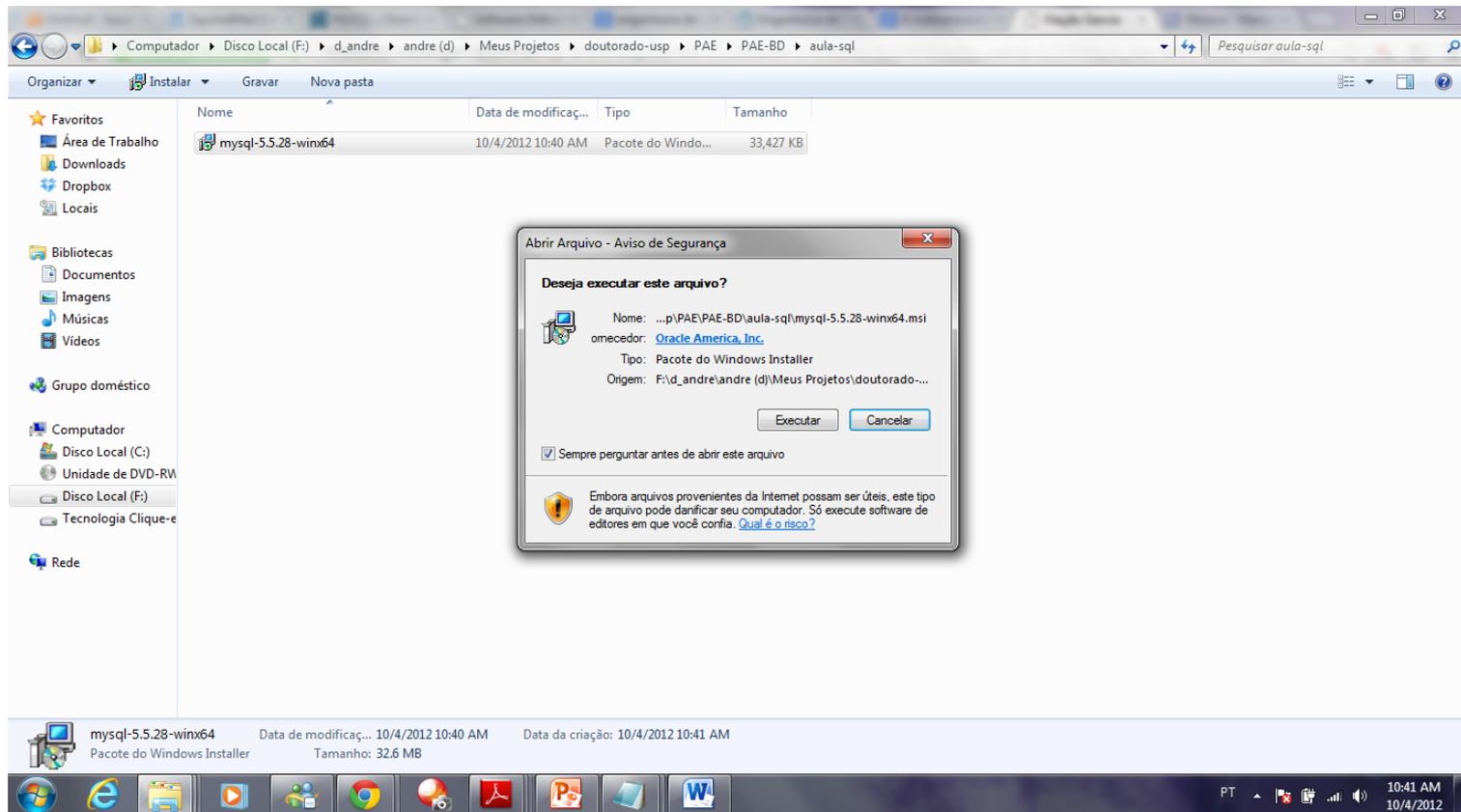
Agenda

- Instalação do MySQL e Workbench
- Comandos DDL – Data Definition Language
 - Criação de banco de dados e tabelas
 - Alteração de campos e tipos de atributos de tabelas
 - Criação de visões
- Exercício

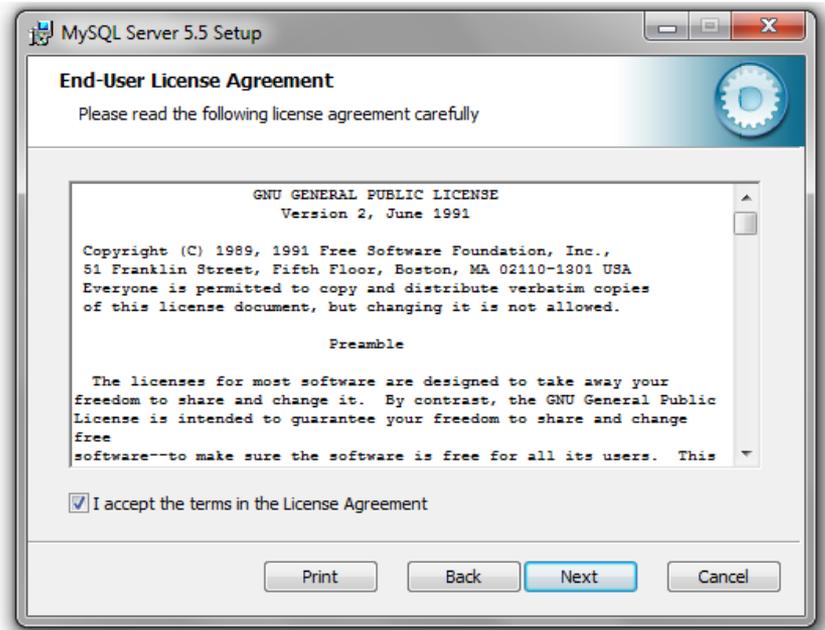
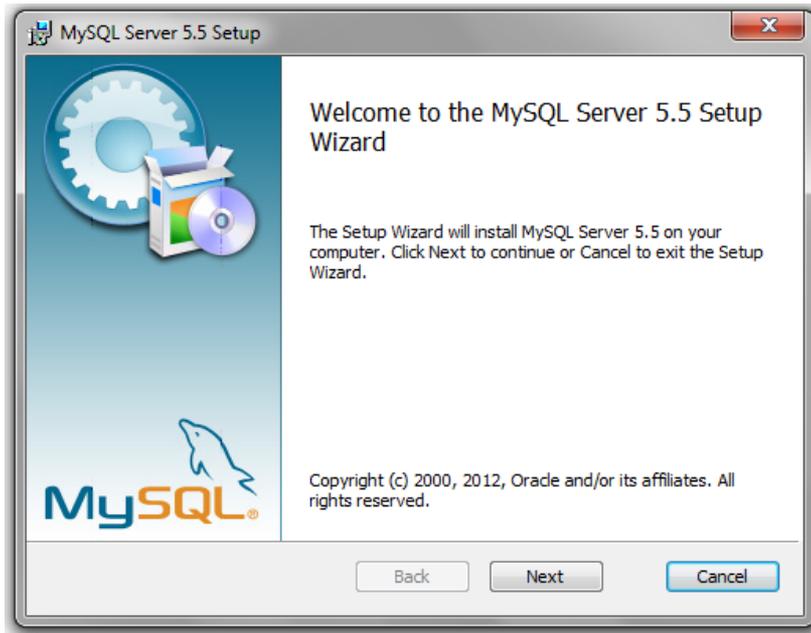
Instalação do MySQL

- Entre no site <http://www.mysql.com/downloads/mysql/> e selecione a versão compatível com seu computador;
- Execute o instalador e siga os passos a seguir:

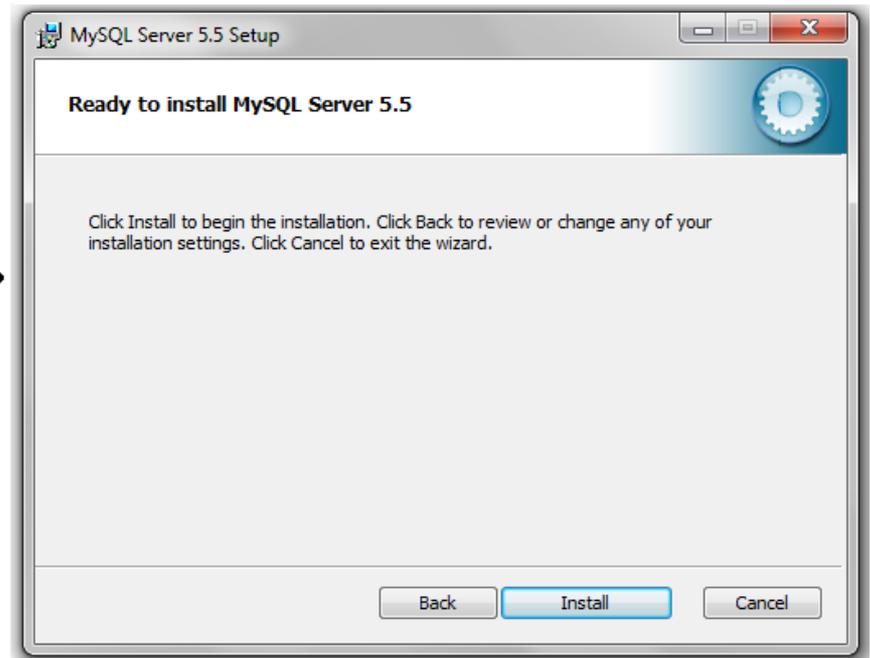
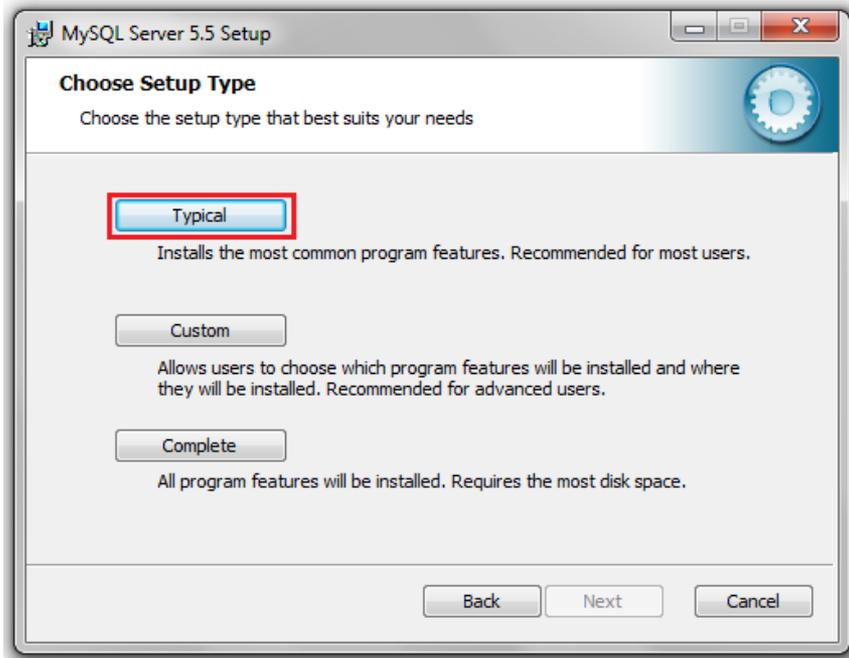
Instalação do MySQL



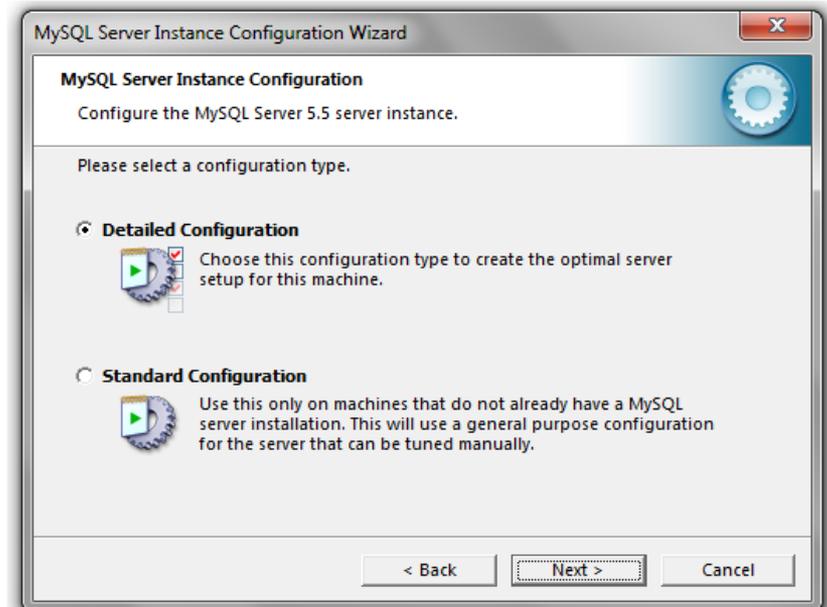
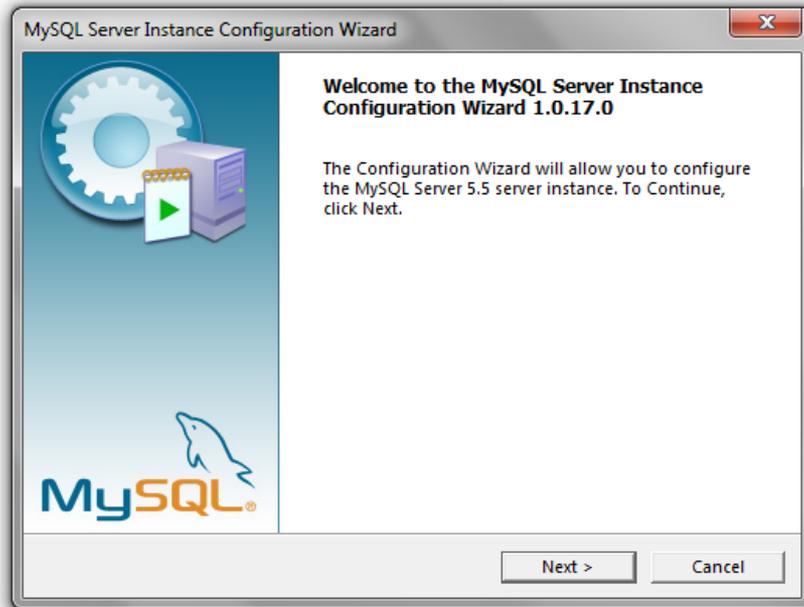
Instalação do MySQL



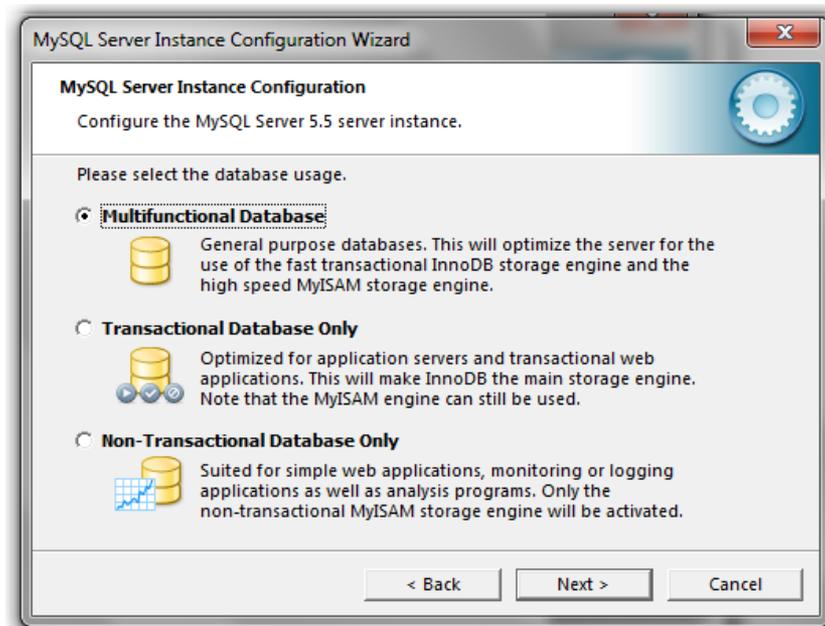
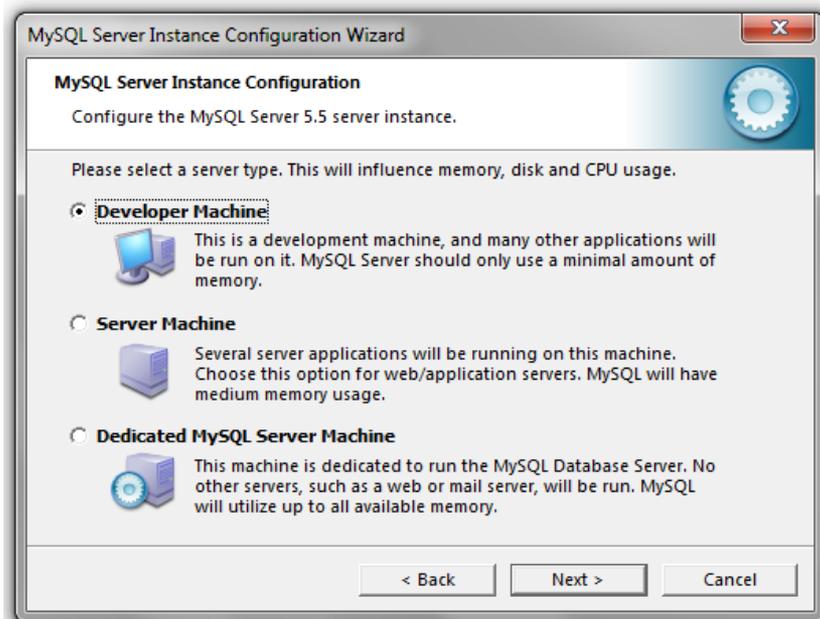
Instalação do MySQL



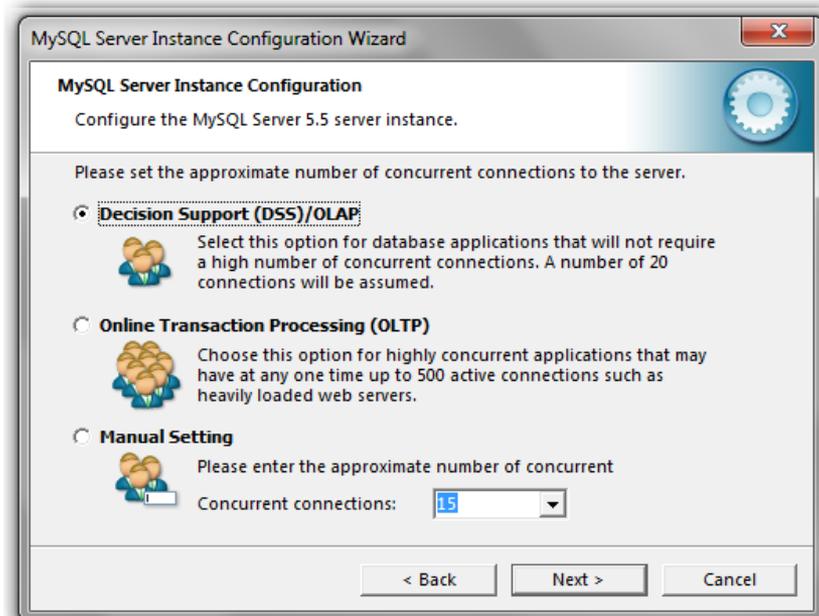
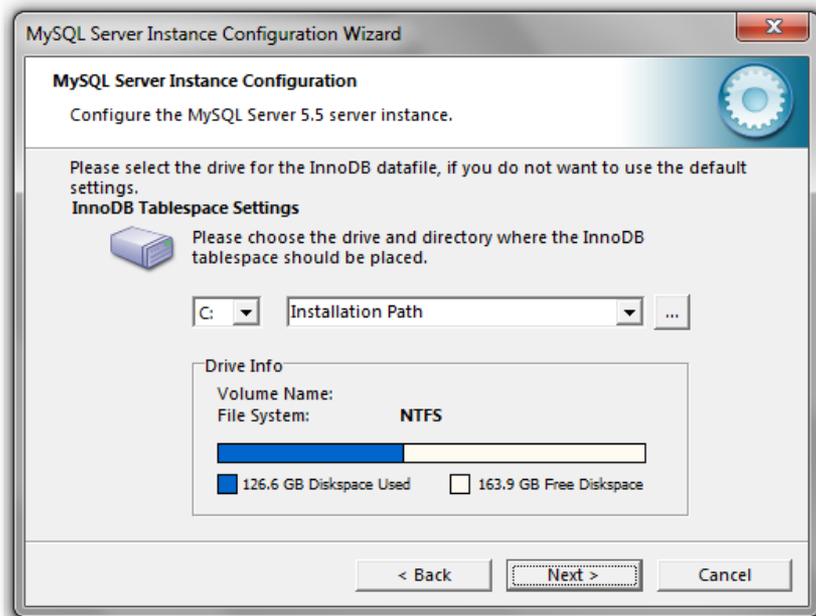
Instalação do MySQL



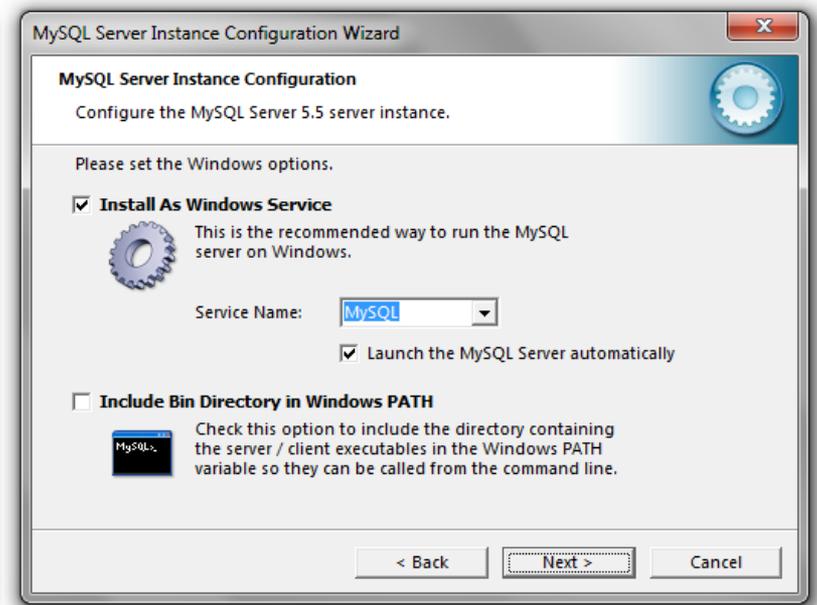
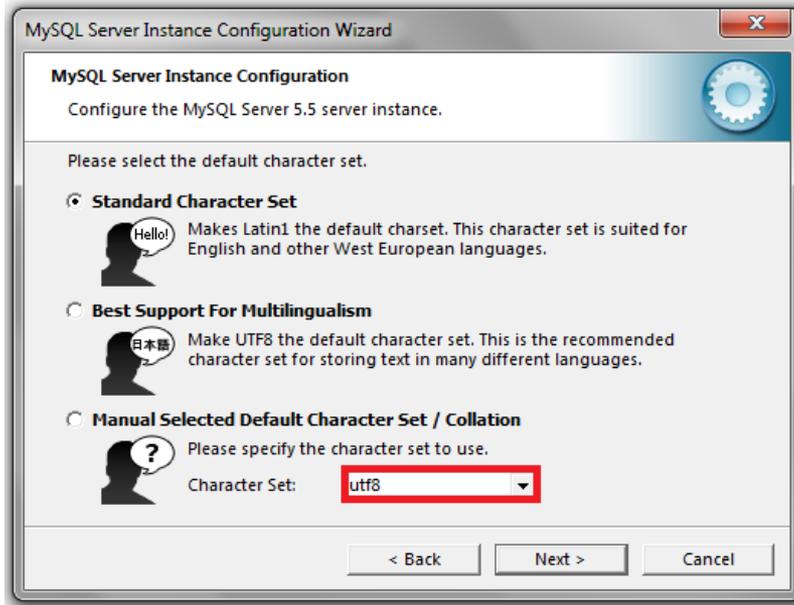
Instalação do MySQL



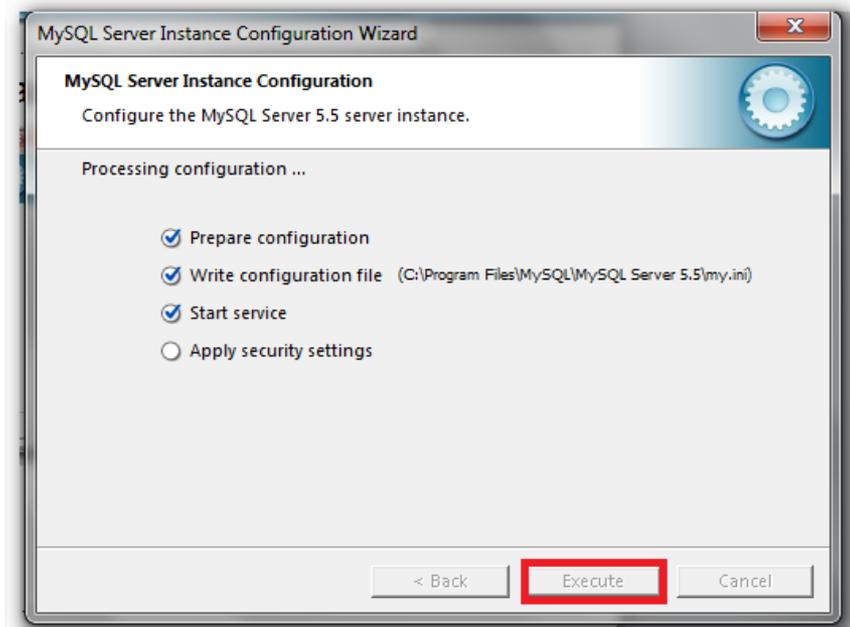
Instalação do MySQL



Instalação do MySQL



Instalação do MySQL



Instalação do MySQL Workbench

- MySQL Workbench é uma ferramenta gráfica para trabalhar com o SGBD MySQL;
- Instalação: entre no site:
<http://www.mysql.com/downloads/workbench/>
- Selecione a versão compatível com o seu sistema operacional e baixe arquivo;
- Execute o instalador.

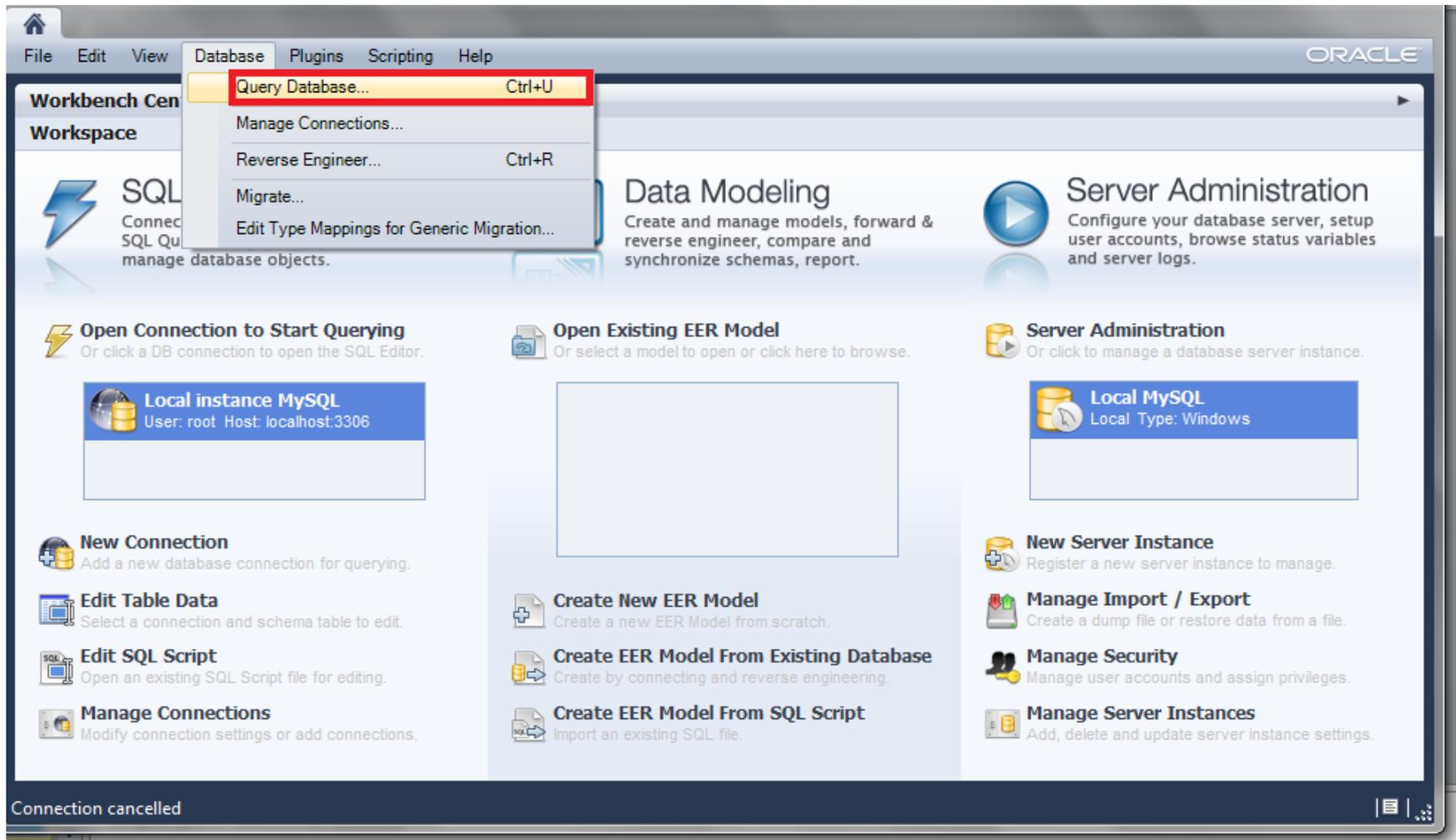
Configurando o MySQL Workbench

The screenshot displays the MySQL Workbench 'Workbench Central' workspace. The 'Database' menu item in the top navigation bar is highlighted with a red box. The interface is organized into three main columns of tool categories:

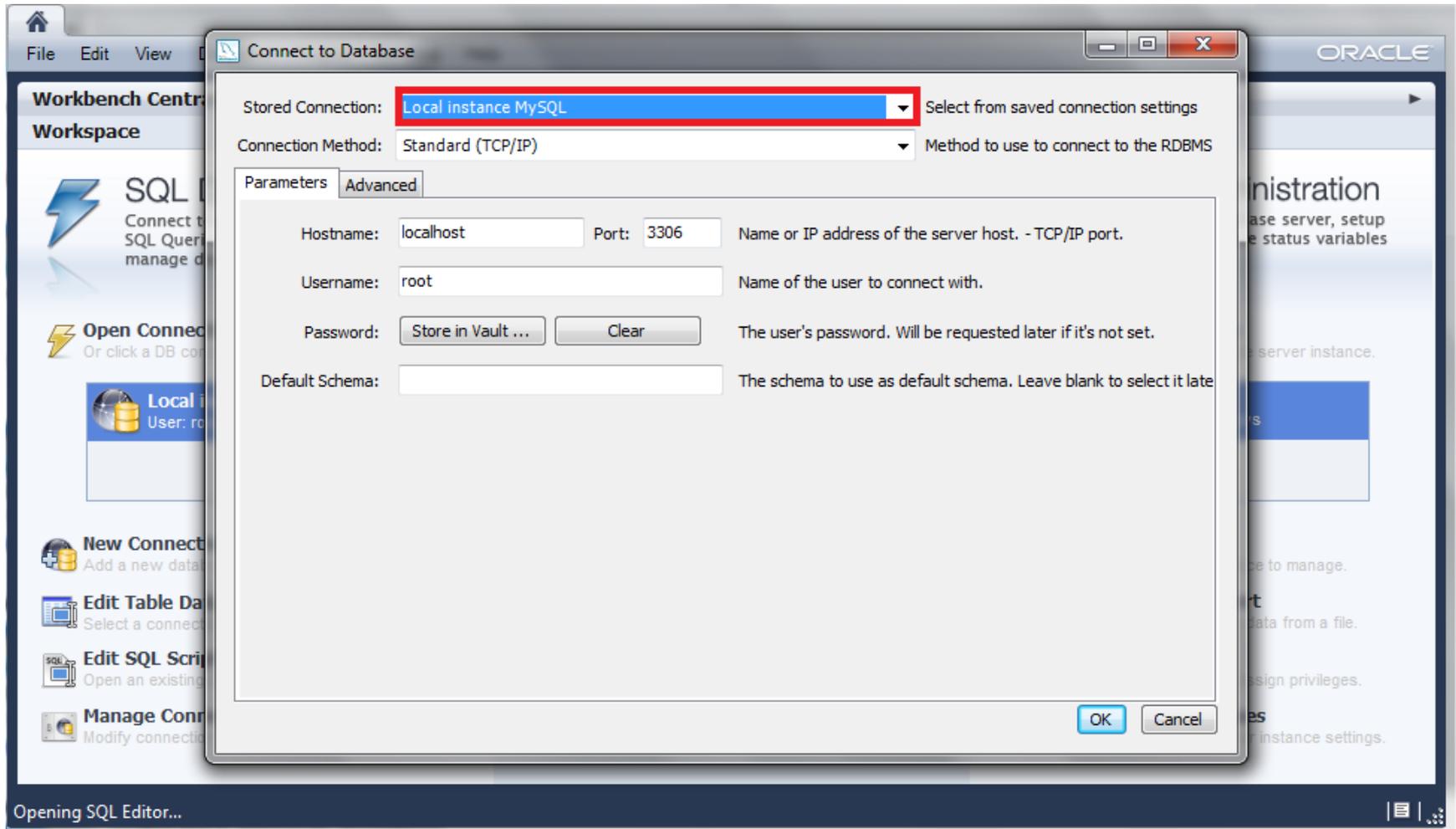
- SQL Development:** Includes 'Open Connection to Start Querying' (with a 'Local instance MySQL' connection listed: User: root, Host: localhost:3306), 'New Connection', 'Edit Table Data', 'Edit SQL Script', and 'Manage Connections'.
- Data Modeling:** Includes 'Open Existing EER Model' (with a large empty box below it), 'Create New EER Model', 'Create EER Model From Existing Database', and 'Create EER Model From SQL Script'.
- Server Administration:** Includes 'Server Administration' (with a 'Local MySQL' instance listed: Local Type: Windows), 'New Server Instance', 'Manage Import / Export', 'Manage Security', and 'Manage Server Instances'.

The status bar at the bottom left shows 'Ready.' and the Oracle logo is in the top right corner.

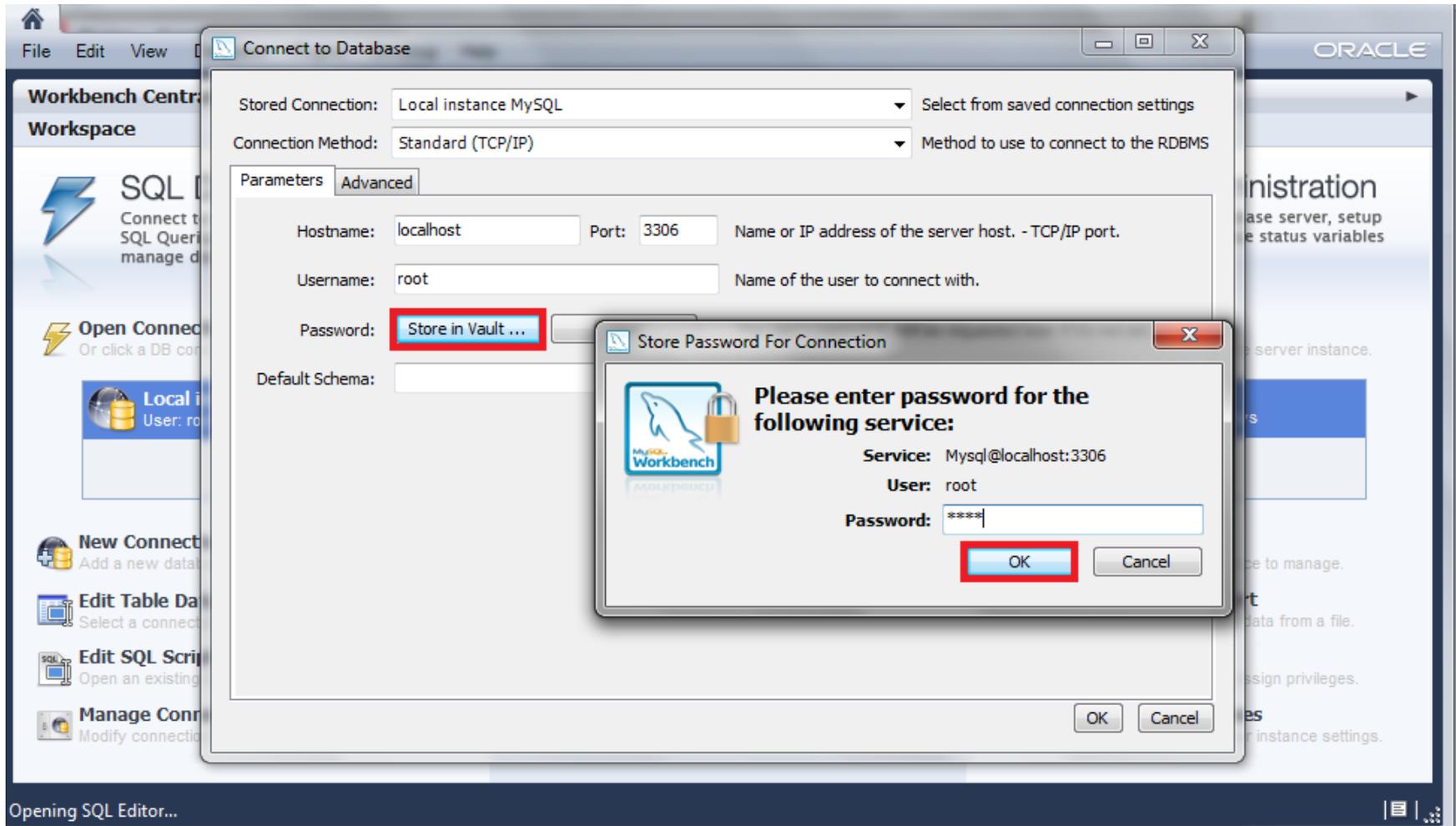
Configurando o MySQL Workbench



Configurando o MySQL Workbench



Configurando o MySQL Workbench



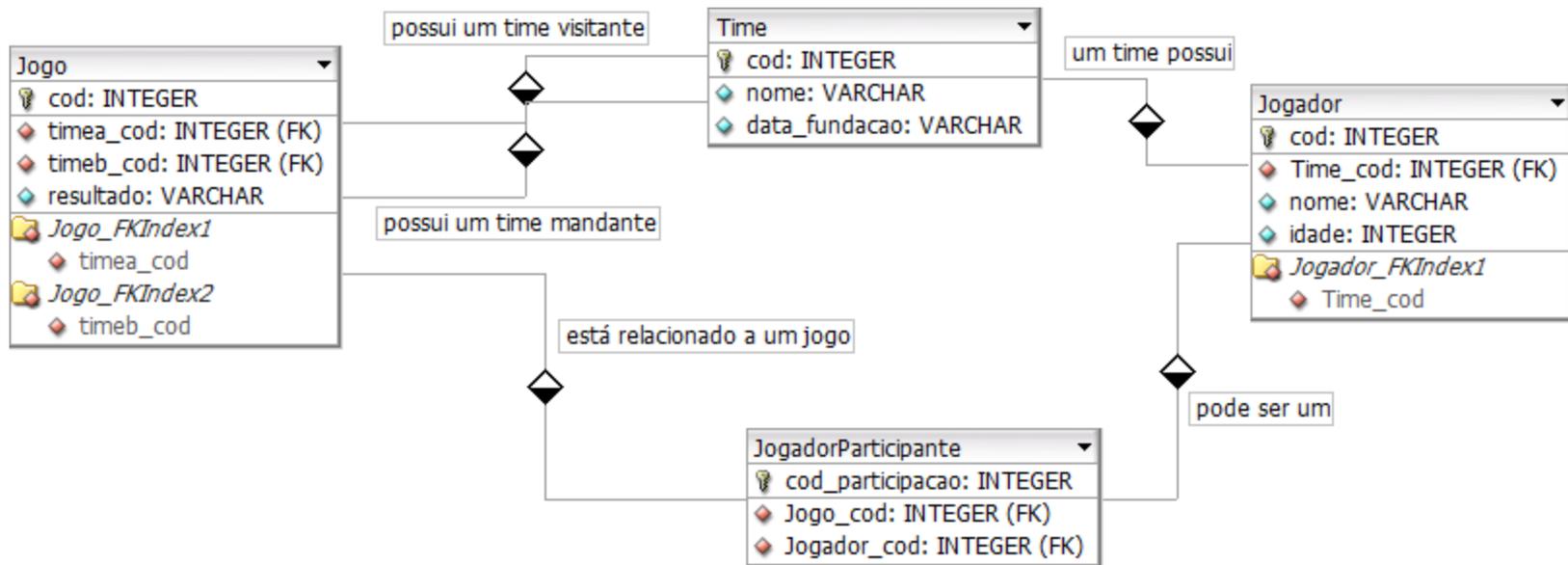
Front-end do MySQL Workbench

The screenshot shows the MySQL Workbench SQL Editor interface. The title bar reads "SQL Editor (Local instance M...". The menu bar includes "File", "Edit", "View", "Query", "Database", "Plugins", "Scripting", and "Help". The Oracle logo is in the top right corner.

Annotations in the image include:

- A red box highlights the toolbar icons for creating a database, table, and view. A text box next to it says: "Ícones de acesso rápido para criação banco de dados, tabelas e visões".
- The "Object Browser" on the left shows a tree view under "SCHEMAS" with a "test" database. A text box below it says: "Área de listagem dos banco de dados existentes no SGBD".
- The central "Query 1" editor has a toolbar and a large text area. A text box in the center says: "Área de execução de comandos SQL".
- The "SQL Additions" panel on the right shows "My Snippets".
- The "Output" panel at the bottom shows "Action Output" with a table header: "Time", "Action", "Message", "Duration / Fetch".
- The status bar at the bottom left says "SQL Editor Opened."

Criação de Banco de dados



SQL – Criação/Deleção de Banco de dados

- **Create database** Campeonato;
- **Drop database** Campeonato;

SQL – Criação de Tabelas

```
create table Time( // nome da tabela
    codigo integer not null,
    nome varchar(50) not null, // atributos
    data_fundacao date not null,
    primary key (codigo) // chave primária
);
```

SQL – Criação de Tabelas com Restrições de Integridade

```
create table jogo ( // nome da tabela
    codigo integer not null,
    cod_time_mandante integer not null,
    Check(cod_time_mandante <> cod_time_visitante),
    cod_time_visitante integer not null, // atributos
    data_realizacao date not null,
    resultado varchar(20) not null,
    primary key (codigo), // chave primária
    foreign key (cod_time_mandante) // chave estrangeira
        references _time(codigo),
    foreign key (cod_time_visitante) // chave estrangeira
        references _time(codigo)
);
```

SQL – Criação de Tabelas com Restrições de Integridade

```
create table jogo ( // nome da tabela
    codigo integer not null,
    cod_time_mandante integer not null,
    Check(cod_time_mandante <> cod_time_visitante),
    cod_time_visitante integer not null, // atributos
    data_realizacao date not null,
    resultado varchar(20) not null,
    primary key (codigo), // chave primária
    foreign key (cod_time_mandante) // chave estrangeira
        references _time(codigo) on update cascade on delete cascade,
    foreign key (cod_time_visitante) // chave estrangeira
        references _time(codigo) on update cascade on delete cascade
);
```

SQL – Remoção de Tabelas

- Comando:
 - `Drop table NomeDaTabela;`
- Remove uma tabela do BD, incluindo seus dados e índices.

SQL – Alteração de Tabelas

- Adição de campos:

```
alter table _time  
    add column cidade varchar(20),  
    add column estado varchar(2);
```

- Remoção de campos:

```
alter table _time drop cidade;
```

SQL – Alteração de Tabelas

- Alteração de nome coluna:

```
alter table _time  
    change cidade logradouro varchar(20);
```

- Alteração de tipo de dado da coluna:

```
alter table _time  
    modify estado varchar(2);
```

Criação de Visões

- Comando:

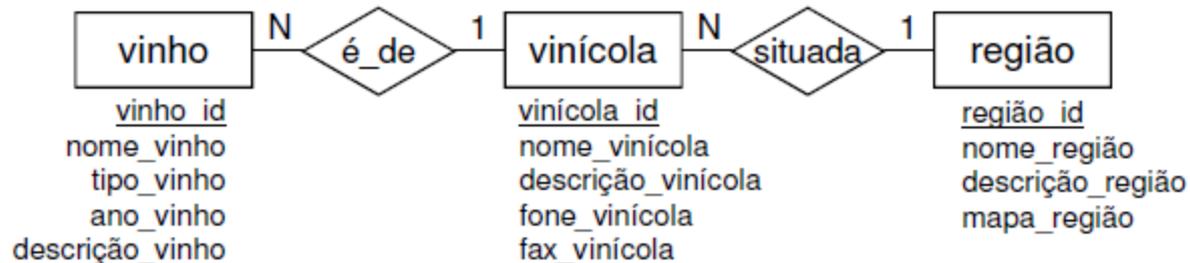
```
create view TimeJogos as
```

```
select _time.codigo as timecodigo, _time.nome as equipe,  
_jogo.codigo as cod_jogo, _jogo.resultado as  
jogoresult from _time, _jogo where
```

```
_jogo.cod_time_mandante = _time.codigo or  
_jogo.cod_time_visitante = _time.codigo; //Comando  
DML
```

Exercício

- Criar um banco de dados com base na seguinte descrição:



- **região** (região_id, nome_região, mapa_região, descrição_região)
- **vinícola** (vinícola_id, nome_vinicola, descrição_vinicola, fone_vinicola, fax_vinicola, **região_id**)
- **vinho** (vinho_id, nome_vinho, tipo_vinho, ano_vinho, descrição_vinho, **vinícola_id**)

Aula de SQL – Comandos DML

André Luiz de Oliveira

Eduardo Raul Hruschka

E-mail:

andre_luiz@icmc.usp.br

erh@icmc.usp.br

Agenda

- Comandos DML – Data Manipulation Language
 - Sintaxe SQL para inserção, remoção e alteração de dados em tabelas;
 - Sintaxe SQL para consultas em tabelas;
- Exercício

SQL – Insert

- Insert into `_time` (`codigo`, `nome`, `data_fundacao`)
values (`001`, `'Fluminense'`, `'1912-04-26'`); ou
- Insert into `_time` values (`001`, `'Fluminense'`,
`'1912-04-26'`);

SQL – Update e Delete

- **Update** _time **set** nome = 'Barcelona' **where**
codigo = 001;
- **Delete from** _time **where** codigo = 001;

SQL - Consultas

- Simples: `select * from _time;`

- Com cláusula where:

```
select * from _jogo
```

```
where cod_time_mandante = 001;
```

SQL – Consultas

- Cláusula Like:

```
select * from _time  
      where nome like 'fl%'
```

- Cláusula UNION:

```
SELECT data_fundacao FROM _time  
      WHERE estado = 'SP'  
  
UNION ALL  
  
SELECT data_fundacao FROM _time  
      WHERE estado = 'RJ';
```

SQL – Junção de Tabelas

SELECT

```
_time.codigo As timecodigo, _time.nome As equipe,  
_jogo.codigo As cod_jogo, _jogo.resultado As jogoresult  
{definição de campos da consulta}  
from _time, _jogo {tabelas envolvidas} where  
    _jogo.cod_time_mandante = _time.codigo or  
    _jogo.cod_time_visitante = _time.codigo;  
{restrição de junção}
```

SQL – Order by

SELECT

```
_time.codigo As timecodigo, _time.nome As equipe,  
_jogo.codigo As cod_jogo, _jogo.resultado As jogoresult  
{definição de campos da consulta}  
from _time, _jogo {tabelas envolvidas} where  
    _jogo.cod_time_mandante = _time.codigo or  
    _jogo.cod_time_visitante = _time.codigo;  
{restrição de junção}  
ORDER BY _time.nome asc;
```

SQL – Group by

SELECT

```
_time.codigo As timecodigo, _time.nome As equipe,  
_jogo.codigo As cod_jogo, _jogo.resultado As jogoresult  
{definição de campos da consulta}  
from _time, _jogo {tabelas envolvidas} where  
    _jogo.cod_time_mandante = _time.codigo or  
    _jogo.cod_time_visitante = _time.codigo;  
{restrição de junção}  
GROUP BY _time.estado;
```

SQL – Cláusula HAVING

SELECT

```
_time.codigo As timecodigo, _time.nome As equipe,  
_jogo.codigo As cod_jogo, _jogo.resultado As jogoresult  
{definição de campos da consulta}  
from _time, _jogo {tabelas envolvidas} where  
    _jogo.cod_time_mandante = _time.codigo or  
    _jogo.cod_time_visitante = _time.codigo;  
{restrição de junção}  
GROUP BY _time.estado  
HAVING _time.data_fundacao > '1900-01-01';
```

SQL – Funções de Agregação de dados

- Count: Quantos times estão cadastrados?

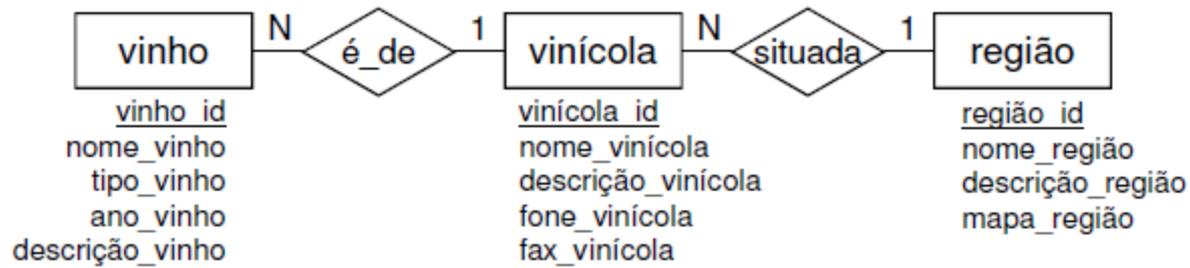
```
SELECT count(codigo) from _time;
```

- Distinct: Quantos estados diferentes possuem times cadastrados?

```
SELECT count(distinct estado) from _time;
```

Exercício

- Executar cláusulas SQL de inserção, remoção e alteração de dados no seguinte BD:



- **região** (região_id, nome_região, mapa_região, descrição_região)
- **vinícola** (vinícola_id, nome_vinícola, descrição_vinícola, fone_vinícola, fax_vinícola, **região_id**)
- **vinho** (vinho_id, nome_vinho, tipo_vinho, ano_vinho, descrição_vinho, **vinícola_id**)

Exercício

- Executar consultas:
 - Simples;
 - Where;
 - Like;
 - UNION;
 - Realizar junções de tabelas;
 - Utilizar cláusulas GROUP BY, ORDER BY e HAVING;
 - Criar consultas com funções de agregação de dados.