

BANCO DE DADOS	3
1. Conceitos de Banco de Dados	3
Banco de Dados	3
SGBD – Sistema Gerenciador de Banco de Dados	3
Dados	3
Bancos de dados disponíveis no mercado	5
Tabela	5
Coluna	5
Domínio	6
Linha	6
Chave primária (primary key)	6
Chave estrangeira (foreign key)	6
Exemplo	7
Views	7
Triggers	7
Procedures	7
Ferramenta de Administração de Banco de Dados	7
2. SQL – Structure query language	8
DDL	8
create table	8
drop table	8
alter table	8
DML	9
Insert	9
Update	10
Delete	10
Select	10
Join	11
Order by	11
Group by	12
Outros comandos usados com select	12
3. Modelagem de Dados	12
MER – Modelo Entidade Relacionamento	12
Entidade	12
Atributo	12
Relacionamento	12
Ferramentas	12
4. JDBC – Conexão com o Banco de Dados	13
JDBC	13
Abrir uma conexão	13
Inserir registros	14
Recuperar registros	14
Atualizar registros	15
Excluir registros	15
5. Referências	15
Firebird	15
MySQL	15
Enterprise Architect	15
Modelagem de Dados	15
6. Apêndice A – Firebird & IB Expert	16

Instalação	16
Criar uma base de dados	16
Janela de execução de scripts múltiplos	16
Janela de execução de scripts unitários	17
Gerenciar tabela	17
7. Apêndice B – MySql & MySql Admin	18
Instalação	18
Tela de acesso	18
Tela principal	18
8. Apêndice C – Enterprise Architect	19
Instalação	19
Criação de um novo projeto	19
Inclusão de entidades	20
Inclusão de relacionamentos	20
Geração de script SQL	21

BANCO DE DADOS

1. CONCEITOS DE BANCO DE DADOS

Banco de Dados

Em suma é uma coleção organizada de dados, inter-relacionados, que representam informações sobre um determinado assunto ou área. Por exemplo, banco de dados de uma universidade, de uma loja, de uma prefeitura, de um site, de uma fábrica, etc.

É também usado de forma genérica, quando queremos falar de todo o conjunto, desde a coleção organizada de dados até os acessos feitos a eles.

SGBD – Sistema Gerenciador de Banco de Dados

Software que disponibiliza os recursos necessários para facilitar a manipulação da estrutura e das informações contidas em um banco de dados.

O SGBD pode gerenciar vários bancos de dados ao mesmo tempo. Em uma mesma máquina poderíamos ter o banco de dados com as finanças da prefeitura, e outro com os dados da secretaria de obras.

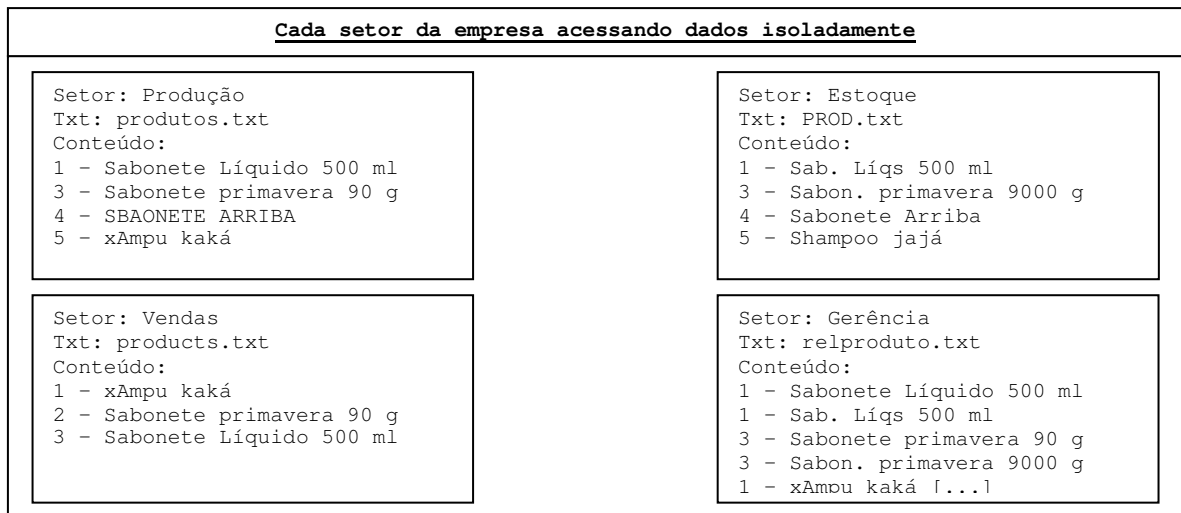
Dados

Dado: aspecto, fato, característica e tudo o mais para o qual há uma definição. É armazenado e recuperado do banco de dados. Um nome, uma notícia, uma cor que poderiam estar em um jornal, caderno, caderneta de telefone, em um arquivo digital.

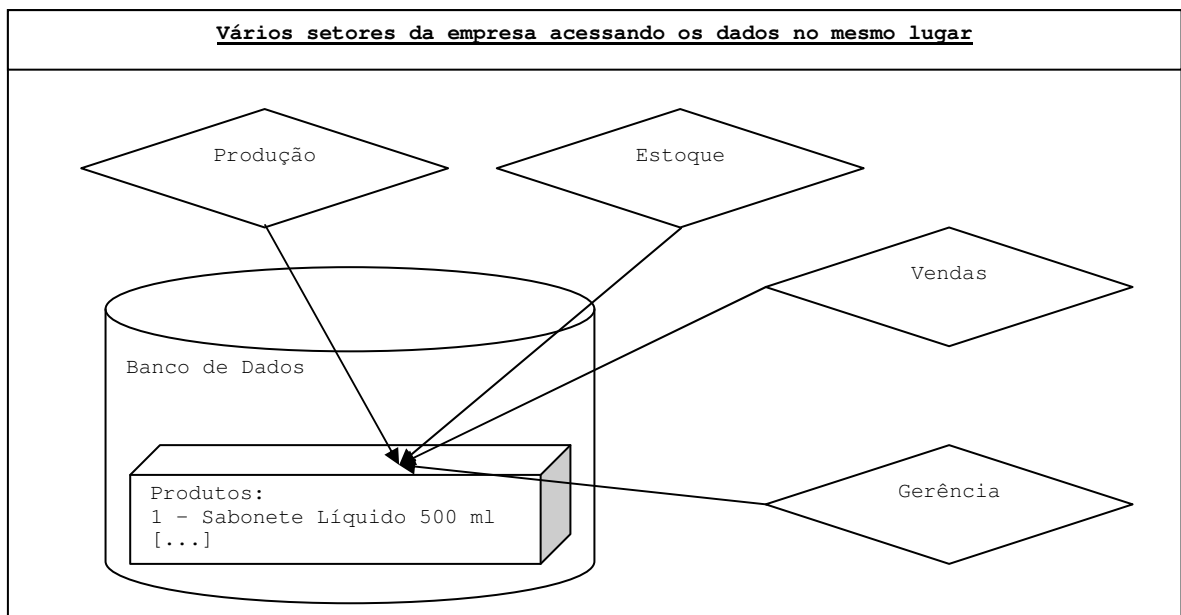
Imagine se ao invés de um banco de dados uma empresa guardasse suas informações em um documento do Word:

<u>Todos os dados ficam misturados no mesmo arquivo</u>
Data: 21/03/2000. Cliente: Pedro da Silva. Comprou: 2 shampoo R\$ 14,50; 5 sabonetes R\$ 4,75; 2 pastas de dente R\$ 3,99; 1 creme de barbear R\$ 6,70; 1 kg de sabão-em-pó R\$ 9,00, 3 detergentes 1 a R\$ 2,30 e 2 a R\$ 1,80. Data: 21/03/2000. Entraram 15 unidades do produto creme de barbear no estoque.

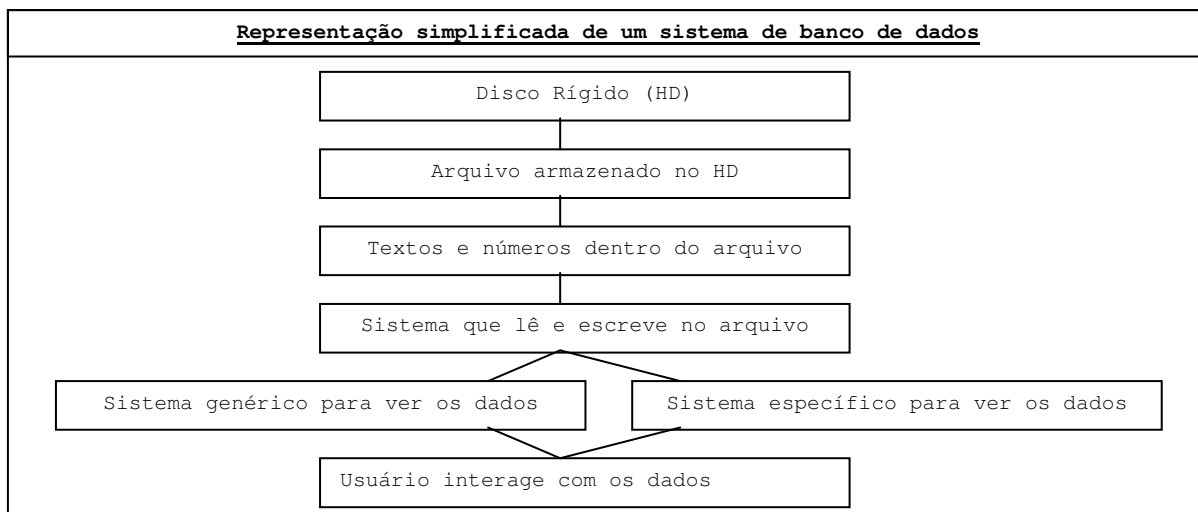
Ou em arquivos textos espalhados pelos setores da empresa:



Vemos que fica difícil organizar todas as informações de forma a se manterem coesas e interligadas. Em um banco de dados, isso se torna trivial:



A figura abaixo nos apresenta de forma simplificada a estrutura que envolve o armazenamento e recuperação de dados em um banco de dados:



Bancos de dados disponíveis no mercado

A lista abaixo trás algumas das opções de bancos de dados do mercado:

Banco de Dados	Empresa	Site	Licença
Oracle	Oracle Corporation	www.oracle.com	Versão free lim.
SqlServer	Microsoft	www.microsoft.com/SQL	Versão free lim.
Sybase	Sybase	www.sybase.com	Pago
Interbase	Borland	www.codegear.com/products/interbase	Pago
Firebird	Firebird Foundation	www.firebirdsql.org	Free
MySql	Sun	www.mysql.com	Free
Outros: PostgreSQL, DB2, Informix, Caché, MSAccess, Paradox, etc.			

Tabela

Nome dado a um conjunto de registros que dizem respeito ao mesmo tipo de informação, ao mesmo assunto, com o mesmo formato.

É como um porta-jóias, um porta-caneta, uma folha de caderno, uma planilha do Excel, uma tabela de um campeonato de futebol.

Por exemplo, para armazenar informações sobre os alunos podemos ter a tabela “Alunos”, sobre professores a tabela “Professores”, além de “Disciplina”, “Turma”, “Notas”, “Frequencia”, etc.

Dentro de um banco de dados uma tabela contém um nome. Pode conter uma ou mais colunas, nenhuma, uma ou mais linhas. Abaixo vemos os conceitos de linha e coluna.

Coluna

A tabela é composta normalmente por um conjunto de dados sobre um determinado tipo de assunto. As colunas permitem organizar esses dados, definindo um nome, um tipo e um tamanho para cada um deles.

Por exemplo, para o aluno podemos ter “nome caractere(50)”, “idade número(3)”, “telefone caractere(20)”, “sexo caractere(1)”.

Não seria nada fácil trabalhar com uma tabela que contivesse apenas uma coluna chamada “infoAluno”, não haveria como garantir uma organização, cada aluno poderia ser cadastrado de forma diferente, começando pelo telefone, outro pela idade, outro pelo. Seria muito difícil, por exemplo, recuperar todos os alunos do sexo masculino.

No banco de dados a configuração de uma coluna ainda vai além, podemos definir se ela é ou não obrigatória, um domínio, e se ela pertence á uma chave primária ou estrangeira como veremos abaixo.

Domínio

Usado para restringir o conteúdo de uma coluna. O tipo e tamanho da coluna é uma forma de restrição, por exemplo, um tipo numérico não permite caracteres alpha, se possui tamanho “(1)”, ele será respeitado.

Contudo, podemos restringir ainda mais. No exemplo do campo “sexo” nós podemos determinar que, só serão aceitos os valores ‘M’ e ‘F’. O campo idade poderia ser restrito ao máximo de 150 anos.

Linha

É cada ocorrência de informação da tabela. Se a tabela não contém informação ela possui 0 (zero) linhas. As colunas definem a estrutura da tabela, as linhas contêm os dados.

No exemplo do aluno, se você e o seu colega do lado estiverem cadastrados, teremos ao menos duas linhas na tabela. Se eu estiver cadastrado teremos ao menos uma linha na tabela professor.

A idéia é que cada linha da tabela seja única, assim como nós também o somos. De que valeria possuir você cadastrado duas vezes como aluno, a não ser para trazer problemas de atualização e confusão na utilização. Se mudar o telefone, teria que atualizar os dois registros, por exemplo. E no cadastro das notas precisamos replicar todos os dados do aluno? Não. Veremos isso na sequência.

Chave primária (primary key)

Para cada tabela defini-se uma (chave simples) ou mais (chave composta) colunas que irão individualizar cada linha, garantindo que cada linha seja um registro único, o qual não poderá se repetir.

Para a tabela aluno poderíamos imaginar num primeiro momento que a chave primária seria o nome, porém, talvez na nossa escola não ocorra, mas em um colégio maior é bem provável que duas pessoas tenham o mesmo nome, como “José da Silva”. Uma boa opção seria o CPF, que é único para cada pessoa, mas, nem todo mundo tem CPF, então, podemos optar por um código sequencial, começando de 1,2, Se você for o aluno com código 54.348, pode imaginar que já passaram muitos alunos por essa escola.

No caso da tabela de frequência a chave primária pode ser composta pelo código do aluno mais a data da chamada.

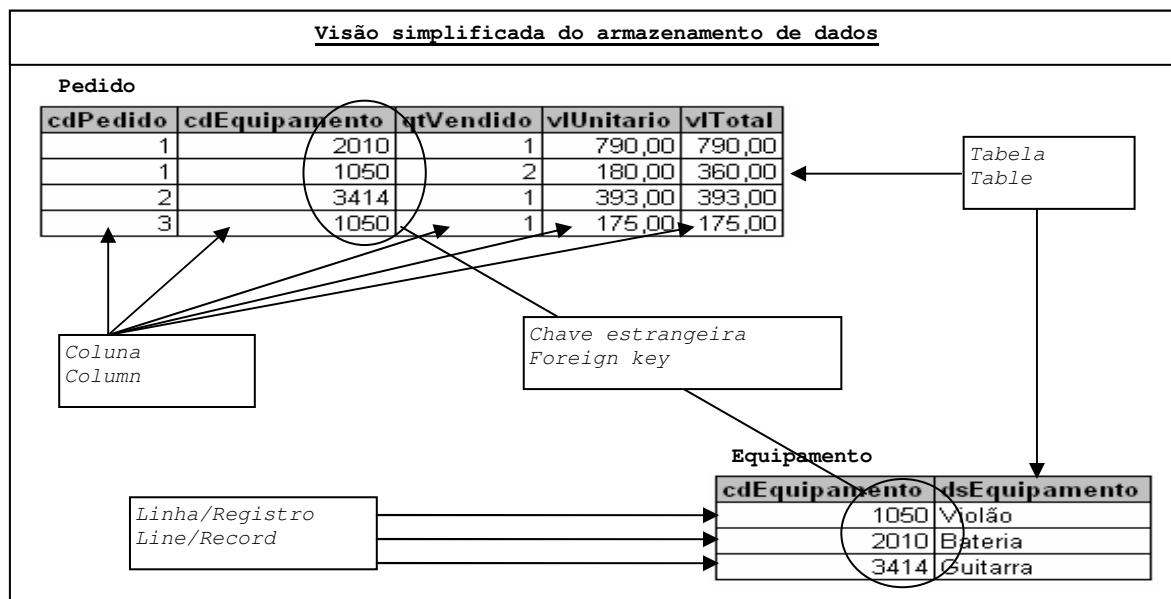
Chave estrangeira (foreign key)

É o relacionamento de um ou mais campos de uma tabela com a chave primária de outra tabela, restringindo que o conteúdo destes se refere a dados de outra tabela e aceita apenas dados cadastrados nesta outra tabela.

É o caso da frequência, para dizer que a presença ou não em uma determinada data é de um aluno do sexo feminino, com telefone X e nome Y, apenas registramos o código na tabela de frequência. Se o maior código de aluno cadastrado for 5.000, não conseguiremos registrar frequência para o aluno 5.001, pois ele não existe.

Exemplo

Gráfico simplificado de como os dados são armazenados no banco de dados.



Views

Visão de parte dos dados de uma ou mais tabelas. Possui as mesmas características estruturais de uma tabela, porém é usada geralmente para visualizar um conjunto selecionado de informações sobre uma ou mais tabelas. Sempre que uma informação de uma tabela for atualizada, essa atualização é refletida na view.

Triggers

Gatilho, disparado quando algum evento ocorre na tabela. Exemplos: antes de uma inserção, após uma alteração, antes de uma exclusão, etc.

Procedures

São rotinas que executam conjuntos, geralmente complexos, de operações sobre o banco de dados.

Ferramenta de Administração de Banco de Dados

Para que possamos interagir com um banco de dados, criar uma base de dados, adicionar tabelas, interagir com os dados precisamos de uma ferramenta de administração de banco de dados.

Para cada banco de dados encontraremos pelo menos uma dessas ferramentas, na maioria das vezes mais de uma. A grande maioria são ferramentas visuais, porém algumas ainda são em modo texto.

No APÊNDICE A apresentamos o Firebird e IB Expert, ferramenta utilizada para administrar o banco de dados Firebird. E no APÊNDICE B apresentamos o MySQL e o MySQL Admin, para administração do banco de dados MySQL.

2. SQL – STRUCTURE QUERY LANGUAGE

DDL

Acrônimo para Data Definition Language (linguagem de definição de dados), é o conjunto de comandos usados para definir a estrutura de um banco de dados. Os comandos vão desde a criação de um banco de dados, montagem de tabelas, colunas, relacionamentos, até definições de comportamento de objetos específicos.

A seguir veremos alguns, os mais usados, pois a lista completa é grande e daria um curso completo se juntássemos os comandos dos principais bancos.

create table

Usado na criação de tabelas, a sintaxe é:

```
create table <nome da tabela> (  
    <nome da coluna> <tipo da coluna> <modificadores>,  
    <... repete para cada coluna ...>  
);
```

O script abaixo cria as tabelas PESSOA e FUNCIONARIO.

```
//Pessoa  
create table PESSOA (  
    cdPessoa decimal(9,0) not null,  
    nmPessoa varchar(50) not null  
);  
  
//Funcionário  
create table FUNCIONARIO (  
    cdFuncionario decimal(9,0) not null,  
    cdPessoa decimal(9,0),  
    dtAdmissao date,  
    vlSalario decimal(12,3)  
);
```

drop table

Para excluir a tabela, conforme sintaxe:

```
drop table <nome da tabela>;
```

O script abaixo exclui as tabelas PESSOA E FUNCIONARIO.

```
drop table PESSOA;  
drop table FUNCIONARIO;
```

alter table

As alterações de tabela mais comuns são para:

Adicionar chave primária:

```
alter table <nome da tabela> add constraint <nome da chave primária> primary key ( <lista de colunas que formam a chave primária separados por vírgula>);
```

Adicionar chave estrangeira:

```
alter table <nome da tabela> add constraint <nome da chave estrangeira> foreign key (<lista de colunas da tabela>) references (<lista de colunas da tabela relacionada>);
```

Adicionar novas colunas:

```
alter table <nome da tabela> add <nome da coluna> <tipo da coluna> <modificadores>;
```

Usando drop, podemos excluir uma constraint ou coluna pelo nome:


```
alter table <nome da tabela> drop constraint/foreign key <nome da constraint>;
```

Abaixo, alguns exemplos de alteração de tabela:

```
alter table PESSOA
add constraint PK_PESSOA primary key (cdPessoa);

alter table FUNCIONARIO
add constraint PK_FUNCIONARIO PRIMARY KEY (cdFuncionario);

alter table FUNCIONARIO
add constraint FK_FUNCIONARIO_PESSOA foreign key (cdPessoa)
references PESSOA (cdPessoa);

alter table FUNCIONARIO
add dsObservacao varchar (200);

alter table FUNCIONARIO drop constraint/foreign key FK_FUNCIONARIO_PESSOA;

alter table FUNCIONARIO
drop dsObservacao;
```

DML

Acrônimo para Data Manipulation Language (linguagem de manipulação de dados), é o conjunto de comandos usado para trabalhar com os dados que estão devidamente armazenados na estrutura definida com base nos comandos DDL utilizados.

Essa manipulação se resume em inserir, alterar, excluir e consultar. Em muitas literaturas vamos encontrar a sigla CRUD do inglês Create-Read-Update-Delete (Criar-Ler-Atualizar-Excluir).

Insert

Comando usado para inserir registros na tabela de banco de dados. Diferentemente do comando “select” que pode recuperar dados de uma ou mais tabelas, este comando, assim como o “update” e o “delete” agem sobre uma única tabela por vez.

A sintaxe do comando “insert” é:

```
insert into <nome da tabela> (<lista de colunas separados por vírgula>)
values (<lista de dados, correspondentes a cada campo, separados por vírgula>);
```

A lista de valores pode ser fixa como acima ilustrado, ou receber o resultado de um “select”:

```
insert into <nome da tabela> (<lista de colunas separados por vírgula>)
<comando SQL formado por um select>
```

Abaixo alguns exemplos de comandos de inserção. Atente para detalhes como o das aspas simples para delimitar um conjunto de caracteres (String/Varchar), o formato da data, e ponto como separador decimal.

```
insert into PESSOA (cdPessoa, nmPessoa)
values (1, 'João Cavalcanti Ferreira');

insert into PESSOA
values (2, 'Maria Madalena Botafogo');

insert into PESSOA (nmPessoa, cdPessoa)
values ('José João Lima', 3);

insert into FUNCIONARIO (cdFuncionario, cdPessoa, dtAdmissao, vlSalario)
values (2, 2, '2005-05-30', 578.99);

insert into FUNCIONARIO (cdFuncionario, cdPessoa, dtAdmissao, vlSalario)
select p.cdPessoa, p.cdPessoa, '2008-01-01', 1000.5 from PESSOA p;
```

Update

Utilizado para atualizar informações já contidas no banco de dados. A sintaxe é um pouco mais elaborada que a de inserção. Além de definir a tabela e as colunas que serão alteradas, podemos (e na maioria devemos) informar condições na cláusula where. Segue sintaxe:

```
update <nome da tabela>
set <nome da coluna> = <novo valor para a coluna>
    <lista de colunas a serem atualizadas precedidas de vírgula>
where (opcional) <lista de condições>
```

Abaixo alguns exemplos de utilização de comandos de atualização:

```
update pessoa
set nmPessoa = 'A';

update PESSOA
set nmPessoa = 'José João Limeira'
where cdPessoa = 3;

update funcionario
set dtAdmissao = '2008-03-03',
    vlSalario = 1680.00
where cdFuncionario = 1;
```

Delete

Utilizado para excluir registros de uma tabela do banco de dados. A sintaxe é simples, conforme podemos observar abaixo:

```
delete from <nome da tabela>
where <lista de condições para a exclusão>
```

Seguem exemplos de exclusão de registros:

```
delete from funcionario
where dtAdmissao < '2008-01-01' and
    vlSalario > 1000.0;

delete from pessoa
where cdPessoa = 2;
```

Select

Comando usando para consultar as informações gravadas em uma ou mais tabelas do banco de dados.

É um comando simples, mas que em conjunto com outros comandos pode se tornar mais complexo, porém, em contrapartida, muito poderoso.

A sintaxe mais simples para o comando select é:

```
select <* para todos, ou lista de colunas separadas por vírgula> from <nome da tabela>
<apelido opcional> where <condições para a consulta>
```

Na cláusula Select definimos os dados que nos interessam conhecer. Na cláusula From definimos a origem, de onde os dados serão coletados. E, por fim na cláusula Where, as condições que devem ser respeitadas na consulta aos dados. A cláusula Where é opcional, se não for informada serão retornados todos os dados.

Exemplos de comandos SQL para consulta:

```

select * from PESSOA;

select cdFuncionario, dtAdmissao, vlSalario from FUNCIONARIO;

select p.cdPessoa, p.nmPessoa from PESSOA p;

select p.nmPessoa from PESSOA p
where p.cdPessoa = 1;

select f.cdPessoa from FUNCIONARIO f
where f.vlSalario > 1000.0;

select f.cdPessoa from FUNCIONARIO f
where f.dtAdmissao >= '2008.01.01' and
      f.vlSalario < 500.0;

```

Os itens “p” e “f” utilizados nos comandos acima chamamos de apelido “alias”. Usado para não precisar repetir o nome completo da tabela nas referências aos nomes de colunas, normalmente opcional, mas necessário quando o mesmo nome de coluna aparece em mais de uma tabela.

Join

Essa extensão do comando Select é utilizada para relacionar duas tabelas. Possui os modificadores “left outer” (pega da primeira tabela mesmo que na segunda não haja correspondência – muito usado), “right outer” (pega da segunda mesmo que não tenha na primeira – pouco usado).

```

//Só trás os dados da pessoa se ela for funcionário
select p.cdPessoa, p.nmpessoa, f.vlSalario
from PESSOA p
join FUNCIONARIO f on f.cdPessoa = p.cdPessoa;

//Trás os dados da pessoa mesmo que ela não seja funcionário
select p.cdPessoa, p.nmpessoa, f.vlSalario
from PESSOA p
left outer join FUNCIONARIO f on f.cdPessoa = p.cdPessoa;

```

Order by

Define a ordem pela qual os resultados devem ser retornados. Pode haver uma ou mais colunas, separadas por vírgula. A ordem preferencial é conforme a ordem de colocação na cláusula “order by”. Podemos usar os modificadores “asc” e “desc” para ordenação crescente e decrescente respectivamente. Detalhe, a ordenação crescente é a padrão e não precisa ser especificada.

```

select f.vlSalario, count(*)
from FUNCIONARIO f
group by f.vlSalario
order by f.vlSalario;

select p.nmPessoa, f.vlSalario
from FUNCIONARIO f
join PESSOA p on p.cdPessoa = f.cdPessoa
order by 2 desc, p.nmPessoa asc;

```

Group by

Usado para agrupar registros, em conjunto com funções especiais como o count, que veremos abaixo. Todos os campos que não são funções especiais devem constar na cláusula “group by” para formar os grupos.

O SQL abaixo retorna a quantidade de pessoas para cada valor de salário. O segundo o somatório dos salários por data de admissão:

```
select f.vlSalario, count(*)  
from FUNCIONARIO f  
group by f.vlSalario;  
  
select f.dtAdmissao, sum(f.vlSalario)  
from FUNCIONARIO f  
group by f.dtAdmissao;
```

Outros comandos usados com select

Nos exemplos acima vimos como utilizar os comandos “count” e “sum”, os demais possuem a mesma sintaxe de utilização. Segue breve descrição de cada comando: count = quantidade, max = maior valor encontrado para a coluna, min = menor valor, sum = somatório, avg = média dos valores, distinct = retorna apenas uma ocorrência mesmo que hajam várias com o mesmo valor.

3. MODELAGEM DE DADOS

MER – Modelo Entidade Relacionamento

Forma utilizada para representar as tabelas, colunas e relacionamento de um banco de dados de forma amigável e de fácil compreensão. Muito utilizada no pré-projeto pelos analistas para validação, durante para servir de orientação, e no pós com documentação de fácil acesso.

Entidade

Representa uma tabela do banco de dados.

Atributo

Representante uma coluna do banco de dados com suas características: tipo, obrigatoriedade, chave primária, etc.

Relacionamento

Representa as chaves estrangeiras que ligam as tabelas, apresentando as características que as definem.

Ferramentas

Existem várias ferramentas no mercado que nos permitem gerenciar visualmente o modelo ER. Muitas delas permitem gerar o script de criação para vários bancos de dados.

Outras permitem inclusive geração de script de alteração, reengenharia (montar o modelo com base no banco).

Segue pequena amostra de opções existentes para modelagem de dados:

Ferramenta	Empresa	Site	Licença
EA	Sparx Systems	http://www.sparxsystems.com.au	Pago/Trial
Power Designer	Sybase	http://www.sybase.com	Pago/Trial
JUDE	Change Vision	http://jude.change-vision.com	Free
Erwin	CA	http://ca.com/br/	Pago/Trial
DBDesigner	Fabulous Force Database Tools	http://www.fabforce.net/dbdesigner4/	Free
Data Architect	theKompany	http://www.thekompany.com	Pago/Trial
Rational	IBM	http://www-306.ibm.com/software/br/rational/	Pago

No APÊNDICE C mostramos como utilizar a ferramenta Enterprise Architect (EA), para modelar e gerar script SQL.

4. JDBC – CONEXÃO COM O BANCO DE DADOS

JDBC

Sigla para Java Database Connectivity, um driver JDBC é uma ponte entre o aplicativo Java e o banco de dados. Composto por um conjunto de classes escritos em Java que implementam as regras necessárias para que os dados possam ser gravados e recuperados de um banco de dados.

Geralmente os próprios fabricantes de banco de dados disponibilizam um driver JDBC. Mas também podemos encontrar driver JDBC desenvolvido por terceiros, algumas vezes mais rápido.

Quem desenvolve um driver JDBC segue um padrão definido pelas JSR do JCP, e para o aplicativo que utilizar conexão JDBC independe qual driver que está usando pois todos possuem um mesmo conjunto de funcionalidades básicas iguais.

Abrir uma conexão

Iniciamos definindo a classe do driver, url, usuário e senha. A chamada “Class.forName” é necessária para garantir que o jar do driver está à disposição. A url define o caminho, o local onde o banco está. Para completar user e password necessários para ter acesso ao banco.

Seguem abaixo exemplos de conexão com os bancos Firebird e MySQL.

```
//Exemplo de conexão com o Firebird
String driver = "org.firebirdsql.jdbc.FBDriver";
String url = "jdbc:firebirdsql:localhost/3050:C:/Testes/Banco/TESTE.GDB";
String user = "sysdba";
String password = "masterkey";
Connection conn;
try {
    Class.forName(driver);
    conn = DriverManager.getConnection(url, user, password);
    System.out.println("Conexão aberta com sucesso");
    //--
    conn.close();
} catch (ClassNotFoundException cnfEx) {
    System.out.println(cnfEx.getMessage());
} catch (SQLException sqlEx) {
    System.out.println(sqlEx.getMessage());
}
```

```
//Exemplo de conexão com o MySQL
String driver = "com.mysql.jdbc.Driver";
String url = "jdbc:mysql://localhost:3306/meubanco";
String user = "root";
String password = "mysql";
Connection conn;
try {
    Class.forName(driver);
    conn = DriverManager.getConnection(url, user, password);
    System.out.println("Conexão aberta com sucesso");
    //--
    conn.close();
} catch (ClassNotFoundException cnfEx) {
    System.out.println(cnfEx.getMessage());
} catch (SQLException sqlEx) {
    System.out.println(sqlEx.getMessage());
}
```

Inserir registros

Para inserir o registro usamos um comando SQL que possui o caractere “?” para definir os pontos em que espera receber parâmetros. Para isso utilizamos a classe `PreparedStatement`. Depois os parâmetros são preenchidos, e o comando SQL executado. Se ocorrer a inserção uma mensagem de sucesso é exibida.

```
PreparedStatement pstmt = conn.prepareStatement("insert into pessoa (cdPessoa, nmPessoa) values(?, ?)");
pstmt.setInt(1, 1);
pstmt.setString(2, "João Paulo da Silva");
int count = pstmt.executeUpdate();
if (count > 0) {
    System.out.println("Insert executado com sucesso");
}
```

Recuperar registros

Para comandos mais simples que não possuem parâmetros usamos a classe `Statement`. O retorno de uma consulta do banco é armazenado na classe `ResultSet`, que entre outros disponibiliza os métodos `next` para definir se ainda há registros, e os métodos que pegam o valor de um campo conforme o seu tipo.

```
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select cdPessoa, nmPessoa from pessoa");
while (rs.next()) {
    System.out.print("Código: "+rs.getInt("cdPessoa"));
    System.out.println(" - Nome: "+rs.getString("nmPessoa"));
}
```

Atualizar registros

Utilizamos também um PreparedStatement, o retorno da execução nos informa a quantidade de registros atualizados.

```
pstmt = conn.prepareStatement("update pessoa set nmPessoa = ? where cdPessoa = ?");
pstmt.setString(1, "Ana Paula Linhares");
pstmt.setInt(2, 2);
count = pstmt.executeUpdate();
System.out.println(count+" registros atualizados com sucesso");
```

Excluir registros

Nenhuma novidade para criar um comando de exclusão.

```
pstmt = conn.prepareStatement("delete from pessoa where cdPessoa = ?");
pstmt.setInt(1, 1);
count = pstmt.executeUpdate();
System.out.println(count+" registros excluídos com sucesso");
```

5. REFERÊNCIAS

Firebird

<http://www.firebirdsql.org/>

<http://www.firebase.com.br/fb/>

<http://ibexpert.net/ibe/>

<http://www.ibexpert.info/documentation/homeindex.html>

MySQL

<http://www.mysql.com/>

Enterprise Architect

<http://www.sparxsystems.com.au/>

<http://www.sparxsystems.com.au/products/ea/trial.html>

Modelagem de Dados

http://pt.wikipedia.org/wiki/Modelagem_de_dados

6. APÊNDICE A – FIREBIRD & IB EXPERT

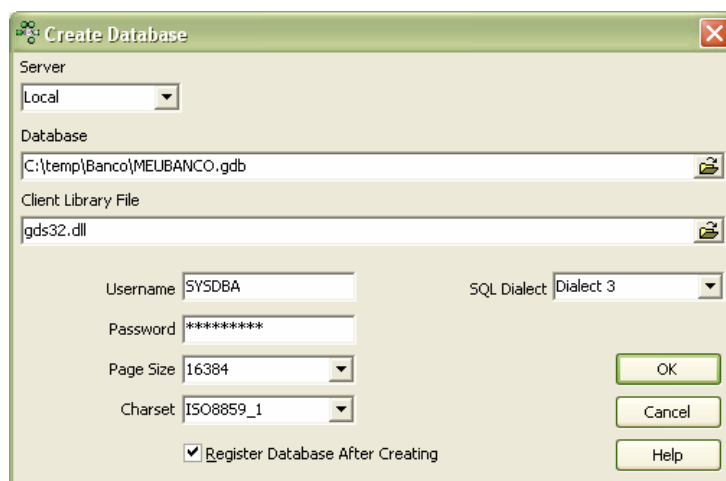
Instalação

Firebird: instalação padrão. Basta seguir os passos da instalação sem precisar alterar nenhuma das configurações. Ao final o serviço do firebird, composto por fbguardian e fbserver estará em operação (SGBD). Para gerenciar o estado e execução acesse com o botão direito sobre “Meu Computador”, a opção Gerenciar, navegue até a aba “Serviços”. Lá encontraremos o “Firebird Guardian” e “Firebird Server”. Todas as configurações e ações devem ser realizadas sobre o “Firebird Guardian”.

IBExpert: instalação padrão. Basta seguir os passos da instalação sem precisar alterar nenhuma das configurações. Segue abaixo detalhamento de como utilizar o IBExpert para interagir com o SGDB do Firebird.

Criar uma base de dados

Acesse o menu Database→Create Database. Defina o servidor (Server: local/remoto), o caminho completo com o nome do banco de dados (Database), usuário de acesso (Username: padrão = SYSDBA) , senha de acesso (Password: padrão = masterkey), Charset e SQL Dialect.



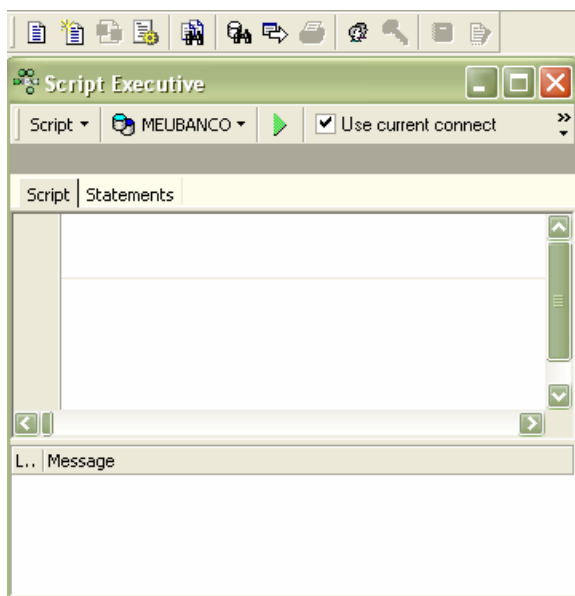
Depois de criado abre-se a tela de registro, informe a versão do Firebird (Server Version), e um apelido para o banco (Database Alias), e confirme.

Caso o banco já exista, acesse diretamente o menu Database→Register Database.

O banco irá aparecer na lista do Database Explorer, para abrir basta um duplo clique sobre o apelido do banco.

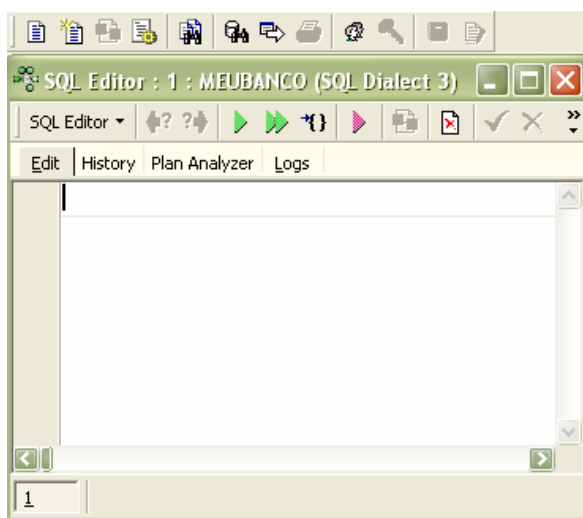
Janela de execução de scripts múltiplos

Clicando no quarto ícone da esquerda para a direita da barra de ferramenta visível na figura abaixo, abrimos a janela Script Executive, usada para executar vários scripts de uma só vez. O botão verde é usado para iniciar a execução dos scripts que devem estar na caixa intermediária. Na última caixa “Message”, podemos acompanhar o resultado da execução.



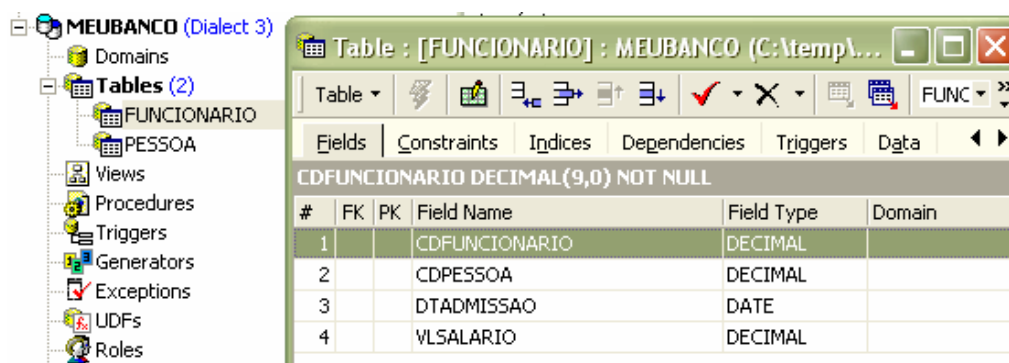
Janela de execução de scripts unitários

Para scripts únicos (primeiro ícone), geralmente de consulta utilizamos o SQL Editor. O comando SQL é escrito e executado a partir da aba “Edit”, e será visualizado na aba “Result”. E caso deseja confirmar a operação, precisa clicar no botão vermelho “Commit”. Podemos ter vários editores abertos ao mesmo tempo, nesse exemplo temos apenas um, como pode ser visualizado no rodapé.



Gerenciar tabela

Para acessar uma tabela basta duplo clique sobre a tabela desejada na árvore da esquerda. As abas “Fields”/Colunas, “Constraints”/Chaves, contém a definição estrutural da tabela, e a aba “Data”/Dados, contém os dados da tabela.



7. APÊNDICE B – MYSQL & MYSQL ADMIN

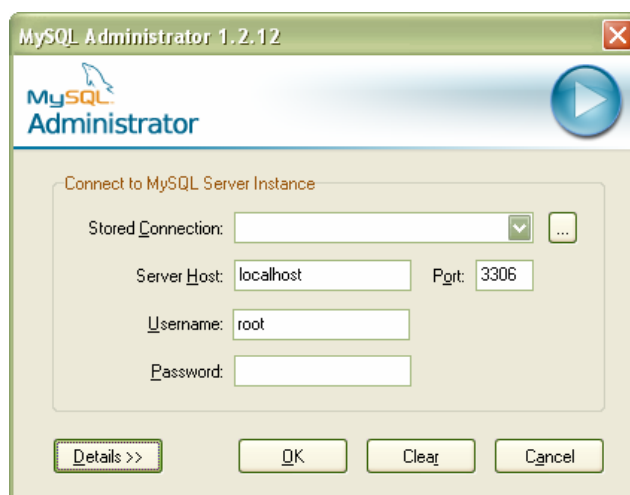
Instalação

MySQL: instalação padrão. Segue os passos sem modificações até a janela “MySQL Server Instance Configuration” onde será necessário informar uma senha. Pode ser qualquer senha, quando mais segura melhor. Coloque “mysql” para uma senha padrão. Para gerenciar o estado e execução acesse com o botão direito sobre “Meu Computador”, a opção Gerenciar, navegue até a aba “Serviços”. Lá encontraremos o MySQL.

MySQL Admin: instalação padrão. Basta seguir os passos da instalação sem precisar alterar nenhuma configuração. Segue abaixo detalhamento de como interagir com o MySQL Admin para gerenciar o SGDB do MySQL.

Tela de acesso

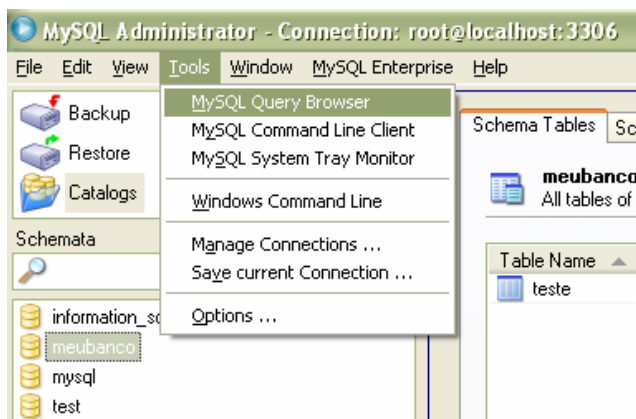
Informe o servidor, “localhost” para a máquina local, ou o nome ou IP da máquina a qual se deseja conectar. Nome de usuário e senha, conforme configurado na instalação, “root/mysql” para o nosso caso.



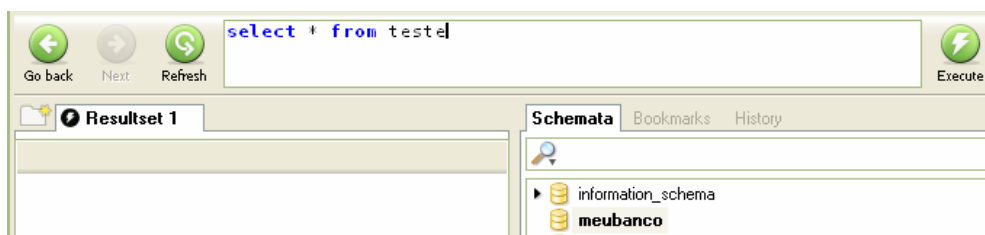
Tela principal

A relação de bancos de dados é acessada clicando no item “Catalogs”, veja na figura abaixo selecionado o item “meubanco”, criado clicando-se com o botão direito na área branca, no item “Creta new Schema”.

A direita aparece a lista de tabelas contidas no banco de dados selecionado. No caso do “meubanco” consta apenas a tabela “teste”. Pode-se interagir com as tabelas existentes e criar novas tabelas.



Para submeter script de metadata e comandos SQL, clique no menu “Tools→MySQL Query Browser”, como pode ser observado na figura anterior. Irá abrir a janela de comandos SQL, conforme figura abaixo. A Janela é composta por uma área para entrada do comando SQL, o botão de execução, o caixa de resultados e na direita definimos sobre qual banco de dados a consulta será submetida.



8. APÊNDICE C – ENTERPRISE ARCHITECT

Instalação

Instalação padrão. A versão é trial, ou seja, após um curto período de uso ela expira e se faz necessária a reinstalação. Efetue a desinstalação, reinicie o computador, exclua a pasta “Sparx Systems” que se encontra nos arquivos de programas e proceda com a reinstalação.

Criação de um novo projeto

Menu File→New Project ...→Selecione o local e o nome do projeto→Selecione a opção Database.

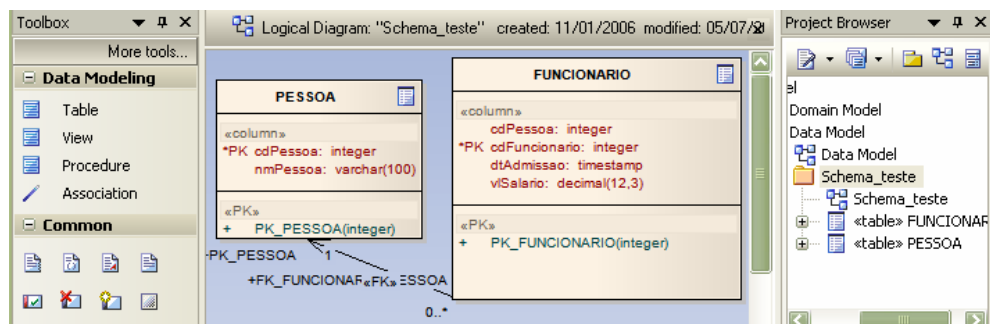
Do lado direito aparecerá a estrutura do projeto no “Project Manager”.

Do lado esquerdo as barras de ferramentas no “Toolbox”.

No centro a área de trabalho.

Para abrir a área de trabalho de criação do Modelo ER, duplo clique sobre o esquema “Schema1” do modelo de dados “Data Model”.

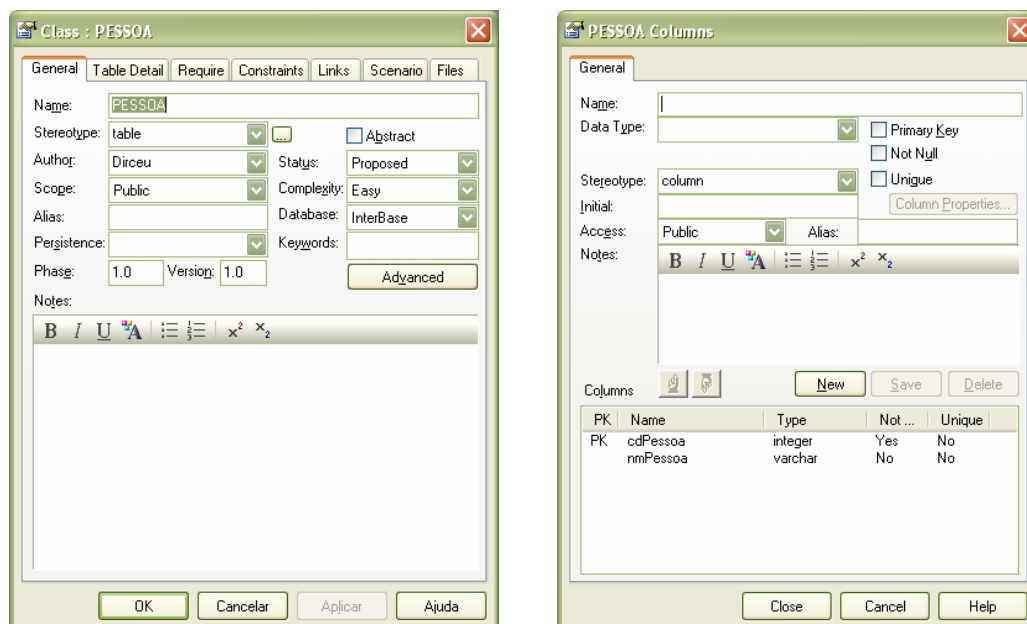
Para visualizar a barra de ferramentas de edição de modelos ER, clique em “More tools...” na barra de ferramentas e escolha “Extended→Data modeling”.



Inclusão de entidades

Clique sobre o componente “Table” e depois clique na área de trabalho no ponde onde deseja inserir. Automaticamente será aberta a tela de propriedades, onde informamos o nome “Name”, e para qual banco de dados será gerado o script “Database”. Isso tudo na aba geral “General”.

Já na aba de detalhes da tabela “Table Detail”, clicamos sobre o botão “Columns Attributes” para gerenciar as colunas da tabela, informando nome “Name”, tipo de dado “Datatype”, e características como chave primária “Primary Key”, obrigatório “Not Null”.



Inclusão de relacionamentos

Para adicionar uma chave estrangeira, primeiro crie uma associação usando o componente “Association”, clicando no “filho” e arrastando até o “pai”. Duplo clique na linha e podemos definir a direção da seta “Direction” em “Source -> Destination”.

Para definir a linha de conexão como uma chave estrangeira “Foreign Key”, clique com o botão direito sobre a linha em “Foreign Keys...”, defina um nome “Name” e selecione as colunas necessárias em cada tabela, depois “Save”.

Foreign Key Constraint

Name:

Source: FUNCIONARIO Target: PESSOA

Key	Column	Type
PK	cdPessoa	integer
	cdFuncionario	integer
	dtAdmissao	timestamp
	vlSalario	decimal

Key	Column	Type
PK	cdPessoa	integer
	nmPessoa	varchar

Source Cardinality: 0..* Target Cardinality: 1

Referential Integrity

☐ Delete Cascade ☐ Update Cascade

Column	Type
cdPessoa	integer

Geração de script SQL

Podemos gerar o script SQL por entidade, clicando com o botão direito em “Generate DDL...” ou no menu Projectc→Database Engineering→Generate Package DDL...