

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Marcelo Caon de Souza

Miguel Kojiio Nobre

Ferramenta de apoio a Engenharia
Reversa de um Banco de Dados
Relacional

Trabalho de Conclusão de Curso

Prof. Dr. Ronaldo dos Santos Mello

Orientador

Prof. Dr. Jose Leomar Todesco

Prof. Dr. Mario Antonio Ribeiro Dantas

Banca

Florianópolis, Novembro de 2005

**Ferramenta de Apoio a Engenharia Reversa de um Banco de Dados
Relacional.**

Marcelo Caon de Souza

Miguel Kojiio Nobre

SUMÁRIO

SUMÁRIO.....	3
RESUMO.....	5
LISTA DE FIGURAS	6
LISTA DE ABREVIATURAS.....	7
1. INTRODUÇÃO	8
1.1. OBJETIVOS.....	9
1.1.1. GERAL.....	9
1.1.2. ESPECÍFICOS.....	9
1.2. JUSTIFICATIVAS	9
1.3. METODOLOGIA	11
1.4. ORGANIZAÇÃO DO TRABALHO	11
2. FUNDAMENTAÇÃO TEÓRICA.....	12
2.1. PROJETO DE BANCO DE DADOS.....	12
2.1.1. Projeto Conceitual.....	13
2.1.2. Projeto Lógico.....	14
2.1.3. Projeto Físico.....	16
2.2. MODELAGEM CONCEITUAL.....	18
2.2.1. Modelo E-R.....	19
2.2.2. Propriedades do Modelo E-R.....	22
2.3. ENGENHARIA REVERSA.....	24
2.3.1. Porque Engenharia reversa.....	25
2.3.2. Seleção da Metodologia para Engenharia Reversa.....	26
2.3.3. Apresentação da metodologia escolhida.....	29
3. DESCRIÇÃO DA FERRAMENTA.....	37
3.1. Inserção e Leitura do Script SQL	39
3.2. Conversão dos Dados Lidos usando um <i>Parser</i> SQL	42
3.2.1. <i>Parser</i> SQL.....	42

3.3. Geração do Esquema E-R e Interação com o Usuário	43
3.4. Apresentação do esquema E-R gerado	46
3.5. Resultados e Contribuições	48
4. CONSIDERAÇÕES FINAIS	50
4.1. TRABALHOS FUTUROS	50
REFERÊNCIAS	52
ANEXOS	53

RESUMO

A elaboração deste trabalho esta centrada em um problema muito comum existente no projeto de banco de dados, a não documentação. Este fato ocorre devido há muito tempo ser utilizado apenas a experiência do usuário, o não uso de metodologias de projeto de banco de dados e o não pensamento em manutenção.

O projeto de banco de dados ocorre em quatro etapas: Levantamento dos requisitos, Projeto Conceitual etapa de mais alto nível e onde ocorre a descrição do banco de dados, Projeto Lógico onde transforma as abstrações geradas no projeto conceitual, etapa anterior, para um modelo que pode ser usado em qualquer banco de dados relacional, por fim o Projeto Físico que determina a estrutura de armazenamento físico dos dados para o uso do banco de dados.

O processo de engenharia reversa busca refazer toda a parte de desenvolvimento de banco de dados agora usando técnicas e metodologias, não somente a experiência do usuário.

Para isto é necessário então refazer ou retornar ao projeto conceitual a partir do projeto físico.

Para tal desenvolvemos uma ferramenta que auxilie o usuário no processo de recriação do modelo conceitual (E-R), linguagem usada para representação do projeto conceitual, do banco de dados.

A ferramenta gera o esquema E-R de forma semi-automática, interagindo com o usuário quando necessário através de “janelas” que apresentam quando ocorrem situações em que mais de uma resposta é possível. Isto ocorre devido a questões que se diferenciam conforme o domínio do problema.

A ferramenta apresenta então ao usuário o esquema E-R gerado graficamente que posteriormente pode ser persistido.

LISTA DE FIGURAS

Figura 2-1 : Projeto de Banco de Dados	13
Figura 2-2 : Esquema Conceitual.....	15
Figura 2-3 : Esquema Lógico.....	15
Figura 2-4 : Esquema Físico	17
Figura 2-5 : Representação de uma Entidade	20
Figura 2-6 : Representação de um Relacionamento	20
Figura 2-7 : Representação dos Atributos.....	21
Figura 2-8 : Representação de Generalização e Especialização	22
Figura 2-9: Engenharia Reversa.....	24
Figura 2-10 : CP composta por várias CE´s	30
Figura 2-11 : Relacionamento de Especialização	30
Figura 2-12 : Atributo Multivalorado	31
Figura 2-13 : Representação de uma Entidade Fraca.....	31
Figura 2-14: Representação de Hierarquia de Especialização	32
Figura 2-15: Representação de uma Entidade Forte	32
Figura 2-16 : Representação de um Relacionamento Binário	33
Figura 2-17 : Representação de uma Entidade Associativa.....	33
Figura 2-18 : Atributo pertencente a um relacionamento da Entidade	34
Figura 2-19 : Atributo pertencente a uma Entidade Especializada.....	34
Figura 2-20 : Atributo pertencente à Entidade.....	35
Figura 2-21 : Identificador de Entidade.....	35
Figura 2-22 : Identificador de Relacionamento	36
Figura 2-23 : Identificador Externo – Chave Estrangeira.....	36
Figura 3-1 : Etapas da Ferramenta	39
Figura 3-2 : Tela de Inserção e Leitura do Script SQL.....	40
Figura 3-3 : Interação com o Usuário	45
Figura 3-4 : Interação com o Usuário	45
Figura 3-5 : Esquema E-R gerado.....	46
Figura 3-6 : Representação Entidade Especializada	47
Figura 3-7 : Representação da Entidade Fraca	48

LISTA DE ABREVIATURAS

BD	- Banco de Dados
E-R	– Entidade Relacionamento
GALS	– Gerenciador de Analisador Léxico e Sintático
SQL	– Structured Query Language
SGBD	– Sistema Gerenciador de Banco de Dados
CP	– Chave Primária
CE	– Chave Estrangeira
GEF	– Graphical Editing Framework
XML	– Extensible Markup Language
UML	– Unified Modeling Language

1. INTRODUÇÃO

Bancos de dados são “uma coleção de dados persistentes utilizada pelos sistemas de aplicação” (DATE, 2000) e sua aplicação surgiu naturalmente, dado o aumento no volume de dados tratados pelos sistemas computacionais. Este gradativo incremento forçou o desenvolvimento de sistemas que fossem capazes de organizar e administrar dados em grande quantidade.

A popularização dos bancos de dados fez com que não apenas profissionais os projetassem, mas também usuários comuns sem conhecimento de como realizar esta tarefa. Diante desta situação desenvolveram-se técnicas e metodologias afim de que fosse seguido um único caminho no desenvolvimento ou manutenção dos bancos de dados.

O desenvolvimento dos bancos de dados desde então vêm sendo tratado como um ciclo de vida que deve ter início, meio e manutenção. Este ciclo não se encerra com a implementação do banco de dados, Ele deve ser continuado de modo que haja constante refinamento e aperfeiçoamento de acordo com novos requisitos ou tecnologias disponíveis.

Apesar de todas as técnicas existentes para o projeto de banco de dados muitos usuários continuam implementando sem o emprego destas. Este trabalho está inserido neste contexto, motivado pela existência de bancos de dados sem nenhuma documentação ou desenvolvidos de forma empírica pelos usuários, com base apenas nas suas experiências. Neste trabalho é apresentada uma ferramenta de engenharia reversa que é capaz de gerar semi-automaticamente o projeto conceitual do banco de dados. A partir desse projeto conceitual, é possível refazer o projeto de banco de dados com base na aplicação de uma metodologia, e não somente na capacidade de um usuário desenvolvedor.

O projeto conceitual é o início da documentação de um banco de dados uma vez que ele gera um esquema conceitual do banco de dados que geralmente é um esquema ER. Esta etapa é importante, pois a partir dela todas as etapas seguintes são derivadas. Sua importância é percebida especialmente quando projetistas ou desenvolvedores, que desconhecem a forma com que foi implementado, precisam realizar algum tipo de manutenção sobre o banco de dados. A modelagem conceitual funciona como um *guia* para que o usuário entenda a relação entre as tabelas e seus respectivos dados. A ausência de um modelo conceitual dificulta a compreensão dos dados, seus relacionamentos e a abrangência do banco de dados (domínio do problema).

A última etapa do projeto de banco de dados é a implementação, e para isto é desenvolvido um esquema físico. Este esquema é uma representação do esquema conceitual em uma linguagem de definição de dados de um Sistema Gerenciador de Banco de Dados (SGBD). O modelo físico é o elemento da documentação do banco de dados que possui a linguagem de mais baixo nível.

1.1. OBJETIVOS

1.1.1. GERAL

O objetivo deste trabalho é a implementação de uma ferramenta para auxiliar na criação de um esquema conceitual de um banco de dados, quando não existe nenhuma documentação de projeto, isto é, possuímos apenas o esquema físico.

1.1.2. ESPECÍFICOS

Esta ferramenta realiza o processo de criação do esquema conceitual por engenharia reversa, ou seja, a partir da definição do banco de dados, ela produz um esquema conceitual correspondente. Este processo não é realizado de forma automática, ou seja, o esquema conceitual pode ser enriquecido e complementado pelas experiências adquiridas pelo usuário, já que a fonte da informação (esquema físico) não é suficiente para que a modelagem conceitual seja fiel a realidade que o banco de dados está abstraindo.

1.2. JUSTIFICATIVAS

Apesar de existirem diversas metodologias para o processo de engenharia reversa, é difícil a escolha de uma delas, pois não existe uma que esgote a matéria. Logo foi feita a opção pela metodologia apresentada por (Heuser 2001) por ser uma das mais completas, já que algumas metodologias não tratavam, por exemplo, das tabelas que possuem chave primária composta por chaves estrangeiras e atributos (BATINI 1992).

O modelo conceitual adotado pela ferramenta é o *modelo entidade-relacionamento* dada sua importância e grande aceitação.

1.3. METODOLOGIA

A metodologia de desenvolvimento do nosso trabalho esta desenvolvida em duas partes: a Fundamentação Teórica e a Implementação da Ferramenta.

Na primeira etapa, fundamentação teórica, demonstramos os conceitos de tudo relacionado a banco de dados relacional, esquema E-R e principalmente Engenharia Reversa, justificativa única e principal para a realização da segunda etapa e principalmente deste trabalho.

Após então a base teórica passamos a etapa de implementação da ferramenta que servira ao processo de engenharia reversa de banco de dados relacionais. Para esta etapa a metodologia de desenvolvimento usada foi a Orientada a Objetos, usando o conceito de “herança”. Tentamos seguir o padrão MVC (model, view, controller).

Após concluir estas etapas finalizamos então o processo de implementação da Ferramenta de auxilio a Engenharia Reversa de Banco de Dados Relacionais.

1.4. ORGANIZAÇÃO DO TRABALHO

Este trabalho contém outros três capítulos: Introdução, Fundamentação Teórica, Descrição da Ferramenta e Considerações Finais.

O capítulo 2 apresenta as áreas relacionadas ao trabalho: banco de dados relacional, projeto de banco de dados relacional e engenharia reversa, dando ênfase na fundamentação teórica para execução do trabalho, os principais conceitos envolvidos e a metodologia utilizada para o processo de engenharia reversa.

O capítulo 3 é dedicado à descrição detalhada da implementação e do funcionamento da ferramenta desenvolvida através da aplicação dos conhecimentos descritos no capítulo 2.

O capítulo 4 apresenta as conclusões e trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. PROJETO DE BANCO DE DADOS

Bancos de dados podem ser considerados a parte mais importante de qualquer sistema de informação, pois neles estão contidos os dados que irão dar suporte às atividades de uma ou mais aplicações. É necessário ressaltar que um banco de dados é uma abstração da realidade para um determinado domínio, ou seja, neles estão armazenadas todas as informações para que possamos *reconstruir* a realidade de acordo a percepção feita do mundo real. O banco de dados implementado e seus dados é o nível mais baixo de abstração enquanto o nível mais alto é o mundo real que este representa.

O desenvolvimento de um banco de dados demanda conhecimento da realidade que está sendo modelada de modo que sejam construídas diversas abstrações antes da implementação. Dada sua importância, existem diversas etapas e requisitos que devem ser cumpridos para o seu desenvolvimento. A conexão de todas estas etapas é o que chamamos de PROJETO DE BANCO DE DADOS.

“Projeto de Banco de Dados é o processo de organização do Banco de Dados, incluindo sua estrutura, conteúdo e aplicações que irão utilizá-lo” (BATINI 1992).

Como em todo *Projeto*, as etapas vão sendo documentadas formalmente durante sua execução, criando um *mapa* que posteriormente servirá de referência para novas implementações sobre o banco de dados existente.

Uma vez levantados os requisitos o Projeto de Banco de Dados é normalmente feito em três etapas, conforme mostra a figura 2.1. A primeira etapa, chamada *projeto conceitual*, produz uma abstração de alto nível da realidade. A segunda etapa, chamada *projeto lógico*, transforma esta abstração em uma representação de dados que possa ser processada por um SGBD. A terceira etapa, chamada de *projeto físico*, determina a estrutura de armazenamento física e os métodos de acesso necessários para um uso eficiente do Banco de Dados (BATINI, 1992).

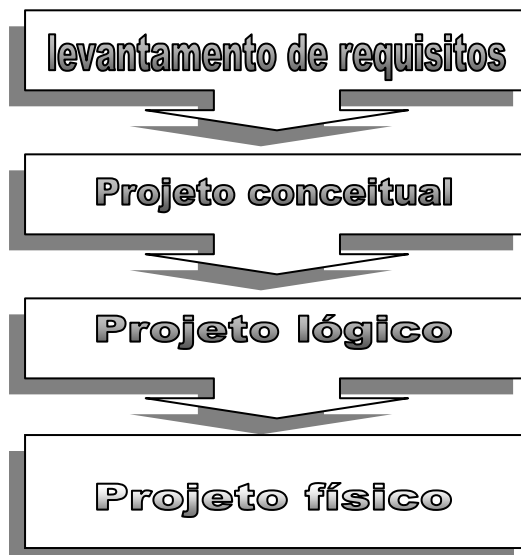


Figura 2-1 : Projeto de Banco de Dados

A descrição de cada etapa do projeto de banco de dados é apresentada a seguir:

2.1.1. Projeto Conceitual

O projeto conceitual é a etapa onde o banco de dados mais se aproxima da visão do mundo real dos dados, ou seja, possui o maior nível de abstração. O objetivo desta etapa é elaborar uma representação de dados com os usuários leigos, uma vez que um modelo conceitual é mais facilmente compreendido por eles.

O modelo conceitual “é uma descrição do banco de dados de forma independente de implementação em um SGBD. O modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de um SGBD.” (HEUSER, 2001). O mesmo “representa e/ou descreve a realidade do ambiente do problema, constituindo uma visão global dos principais dados e relacionamentos (estruturas de informação), independente das restrições de implementação” (MACHADO, 1999).

O projeto conceitual conforme (BATINI, 1992) lida com dois conceitos:

- **Esquema conceitual:** Descrição de alto nível da estrutura do banco de dados, independente do SGBD que é utilizado na implementação. O esquema é o resultado da aplicação desta etapa;
- **Modelo conceitual:** Conjunto de conceitos utilizado para descrever esquemas conceituais. Um exemplo de modelo conceitual é o modelo entidade-

relacionamento (ver seção 2.2), que possui conceitos como entidade, atributo e relacionamento.

2.1.2. Projeto Lógico

A partir do esquema conceitual gera-se o esquema lógico que é “uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD” (HEUSER, 2001). Essa representação, por sua vez, é independente dos dispositivos ou meios de armazenamento físico e das estruturas de dados por elas definidas.

Nesta etapa do projeto de banco de dados, diferentemente da anterior, existe a preocupação em fazer a modelagem de acordo com o SGBD que vai gerenciar o banco de dados. Enquanto na modelagem conceitual o objetivo é fazer o esquema mais completo e expressivo possível, o objetivo na modelagem lógica é obter uma representação eficiente que obedeça a um modelo de dados de um SGBD, como o modelo relacional.

A seguir temos um exemplo de esquema conceitual para um domínio hospitalar (figura 2.2) e um exemplo de modelagem lógica baseado nela (figura 2.3).

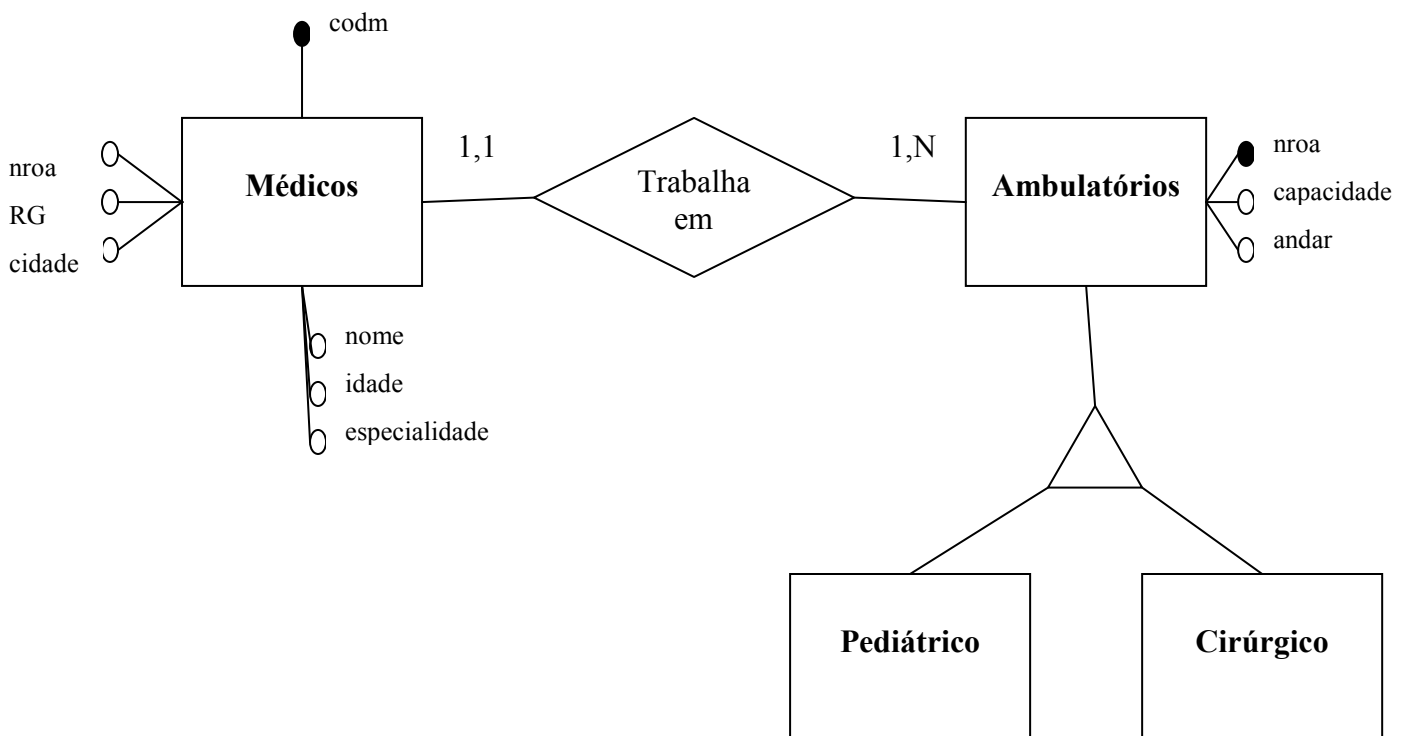


Figura 2-2 : Esquema Conceitual

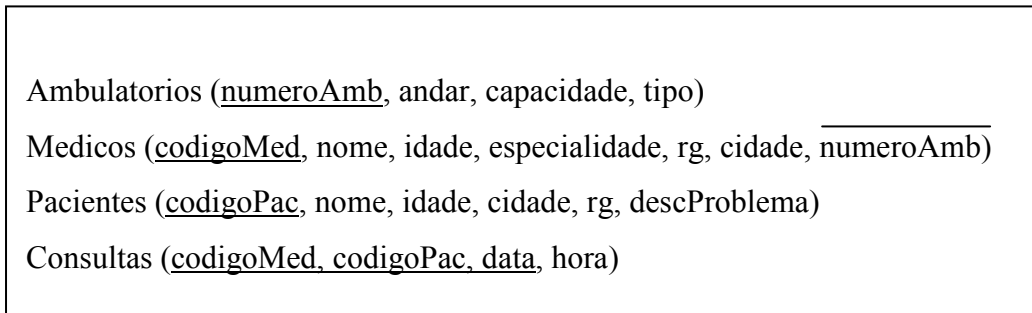


Figura 2-3 : Esquema Lógico

2.1.3. Projeto Físico

O objetivo do projeto físico é desenvolver um esquema físico que “descreve as estruturas físicas de armazenamento de dados, tais como: tamanho de campos, índices, tipo de preenchimento destes campos, etc, projetadas de acordo com os requisitos de processamento e uso mais econômico dos recursos computacionais” (MACHADO, 1999). Além disso, todos os requisitos de dados que não puderam ser expressos nos modelos anteriores, como restrições de integridade, devem neste momento ser detalhados e implementados.

É necessário ressaltar que esta é a etapa onde a linguagem é a de mais baixo nível entre todas as etapas. Logo, um leigo ou mesmo um especialista tem muita dificuldade para compreender a dimensão do banco de dados no que se refere a suas entidades e seus relacionamentos.

O modelo físico é a descrição da implementação do banco de dados na memória secundária. Ele descreve as estruturas de armazenamento e os métodos de acesso usados para acessar efetivamente a informação. Portanto, o modelo físico serve a um SGBD específico sendo utilizada a linguagem de definição de dados do SGBD (DDL) para realizar esta implementação.

Na figura 2.4 a seguir é apresentado um esquema físico para o esquema lógico de banco de dados apresentado anteriormente na linguagem SQL.


```

create database hospital

create table ambulatorios
(
  numeroAmb      int,
  andar          numeric(3) NOT NULL,
  capacidade     smallint,
  tipo           varchar(40),
  primary key(nroa)
)

create table medicos
(
  codigoMed      int,
  nome           varchar(40) not null,
  idade         tinyint not null,
  especialidade  char(20),
  rg            numeric(10) unique,
  cidade        varchar(30),
  numeroAmb     int,
  primary key (codm),
  foreign key(nroa) references ambulatorios
)

create table pacientes
(
  codigoPac      int,
  nome           varchar(40) not null,
  idade         tinyint not null,
  cidade        char(30),
  rg            numeric(10) unique,
  descProblema  varchar(40) not null,
  primary key (codp)
)

create table consultas
(
  codMedico      int,
  codPaciente    int,
  data          datetime,
  hora          datetime,
  foreign key (codm) references medicos,
  foreign key (codp) references pacientes,
  primary key (codm, codp, data)
)

```

Figura 2-4 : Esquema Físico

2.2. MODELAGEM CONCEITUAL

O objetivo da modelagem conceitual é produzir um esquema de uma dada realidade, com maior ou menor grau de fidelidade. Este esquema faz parte da primeira etapa do projeto de banco de dados e neste momento não existe preocupação com a implementação no banco de dados. Apesar de o esquema conceitual ser uma abstração de alto nível, ele deve possuir um formalismo para que seja possível sua implementação em computadores.

Um modelo conceitual é geralmente baseado em símbolos para os quais deve haver uma conceituação rigorosa e possui algumas propriedades como “poder modelar qualquer realidade e ter características gráficas que sejam bastante simples de construir e entender” (MACHADO, 1999).

Segundo (BATINI, 1992), um modelo conceitual deve possuir os seguintes requisitos:

- **Expressividade:** Muitos modelos possuem uma grande variedade de conceitos o que permite uma representação abrangente da realidade, ou seja, procuram representar qualquer abstração do mundo real no que tange a dados. Portanto modelos ricos em conceitos também são muito expressivos. Em alguns casos o alto nível de expressividade prejudica a simplicidade, pois o modelo se tornaria muito complexo. Neste caso o modelo pode ser traduzido como “baixa simplicidade”;
- **Simplicidade:** Um modelo conceitual deve ser simples, para que o esquema construído utilizando-se aquele modelo seja de fácil compreensão para os projetistas e usuários das aplicações do Banco de dados;
- **Minimalidade** (Minimality): Esta propriedade é garantida se todo conceito presente no modelo tem um significado distinto com respeito a cada um dos outros conceitos;
- **Formalidade** (Formality): Esquemas produzidos utilizando modelos de dados conceituais representam uma especificação formal do dado. Formalidade requer que todos os conceitos do modelo tenham uma única, precisa e bem definida interpretação.

Antes do início do processo de modelagem, o projetista de banco de dados deve observar algumas características da realidade que ele deseja modelar. Para que haja sucesso, devem ser definidos a abrangência e o nível de detalhamento. O conjunto destas definições é chamado de *domínio do problema*.

A qualidade de um modelo conceitual não está apenas ligada ao domínio das técnicas que o projetista de banco de dados possui. É necessário ressaltar que o modelo escolhido também é muito importante na abstração obtida.

Diversos modelos conceituais são utilizados para projeto de banco de dados, como o Modelo Entidade-Relacionamento (modelo E-R) e o modelo ORM. O modelo E-R, por ser o mais popular destes, é adotado neste trabalho e detalhado a seguir.

2.2.1. Modelo E-R

O modelo E-R é abrangente, relativamente simples e tem sido amplamente utilizado. Ele foi apresentado em 1976 por Peter Chen sob um trabalho intitulado “The Entity-Relationship Model: Toward the unified view of data” (ACM Press New York, NY, USA) e desde então se tornou cada vez mais popular.

Segundo Chen, a visão de uma dada realidade “Baseia-se no relacionamento entre as entidades, os quais retratam os fatos que governam esta mesma realidade, e que cada um (entidade ou relacionamento) pode possuir atributos (qualificadores desta realidade)” (MACHADO, 1999).

O objetivo principal deste modelo é a formalização das entidades e seus relacionamentos e fazer a representação destes conceitos em um diagrama. Segundo COUGO este objetivo principal é “classificado como formalização do óbvio”.

Originalmente o modelo E-R incluía apenas os conceitos de entidade, relacionamento e atributos. No entanto, outros conceitos foram sendo adicionados (atributo composto, generalização/especialização, etc.) o que tornou este modelo mais expressivo.

Os conceitos básicos presentes no modelo E-R são os seguintes:

Entidades: Abstraem um conjunto de objetos do mundo real que possuem dados relevantes de serem mantidos no banco de dados.

No diagrama *entidade-relacionamento* este objeto é representado através de um retângulo.

Entidade fraca: É um caso particular de entidade que se caracteriza “através da análise de existência de duas condições básicas, dependência de existência ou dependência de identificador. Uma entidade é fraca se um desses dois tipos de dependência se verifica entre uma entidade A e uma entidade B” (Cougo, 1997).

A figura 2.5 a seguir representa graficamente uma entidade no modelo E-R.



Figura 2-5 : Representação de uma Entidade

Relacionamentos: Representa a associação entre duas ou mais entidades. Um relacionamento pode ocorrer entre entidades do mesmo tipo ou de tipos diferentes. O relacionamento entre entidades de tipos diferentes é o mais comum. No entanto, a associação entre entidades do mesmo tipo recebe o nome de *auto-relacionamento* ou *relacionamento recursivo*. É importante salientar que o nome dado ao relacionamento é de suma importância, pois uma nomenclatura inadequada pode passar a imagem de um conceito diferente daquele observado. No modelo E-R um relacionamento é representado por um losango com ligação às respectivas entidades (ver figura 2.6).

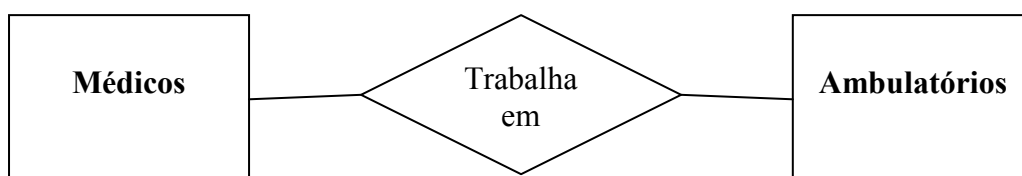


Figura 2-6 : Representação de um Relacionamento

Atributos: Especificam as propriedades elementares de uma entidade ou relacionamento e individualizam cada objeto ou associação em meio ao conjunto deles. Os atributos são sempre descritivos (descrevem características intrínsecas dos objetos). Relacionamentos podem não ter atributos, mas no caso de entidades esta é uma condição obrigatória. Atributos são representados graficamente através de apêndices conforme mostra a figura 2.7

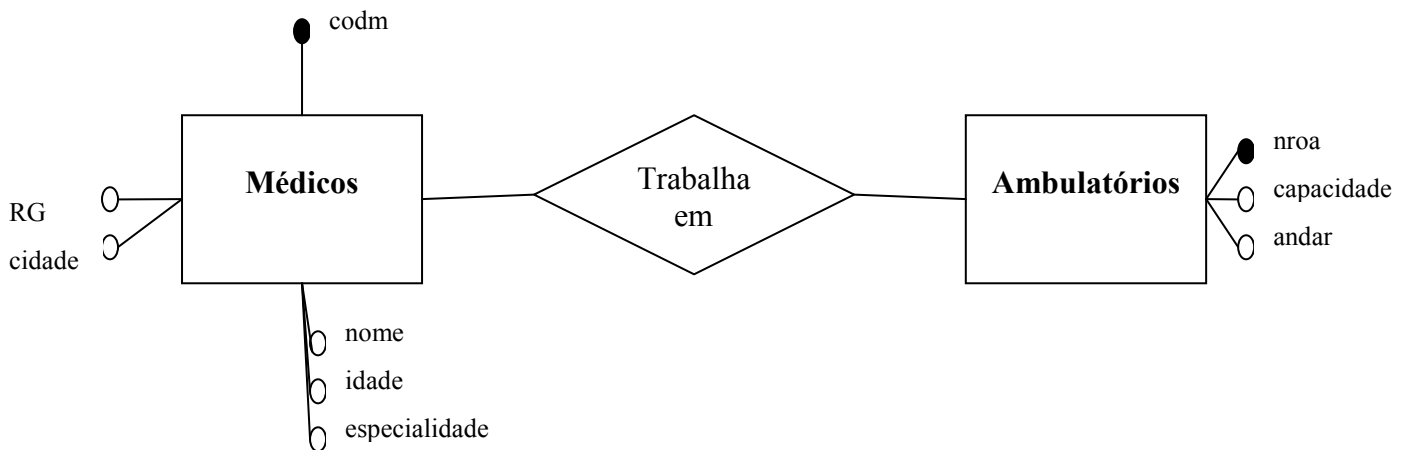


Figura 2-7 : Representação dos Atributos

Generalização/Especialização: Representa uma propriedade especial, que distingue um grupo de objetos de seus pares. A generalização/especificação permite que sejam derivados de entidades genéricas subconjunto(s) de entidade(s) distinta(s), mantendo a percepção de que a entidade faz parte tanto do subconjunto distinto quanto do conjunto completo. Este conceito é representado no diagrama através de um triângulo isósceles ligando as entidades *especializadas* a sua genérica (ver figura 2.8).

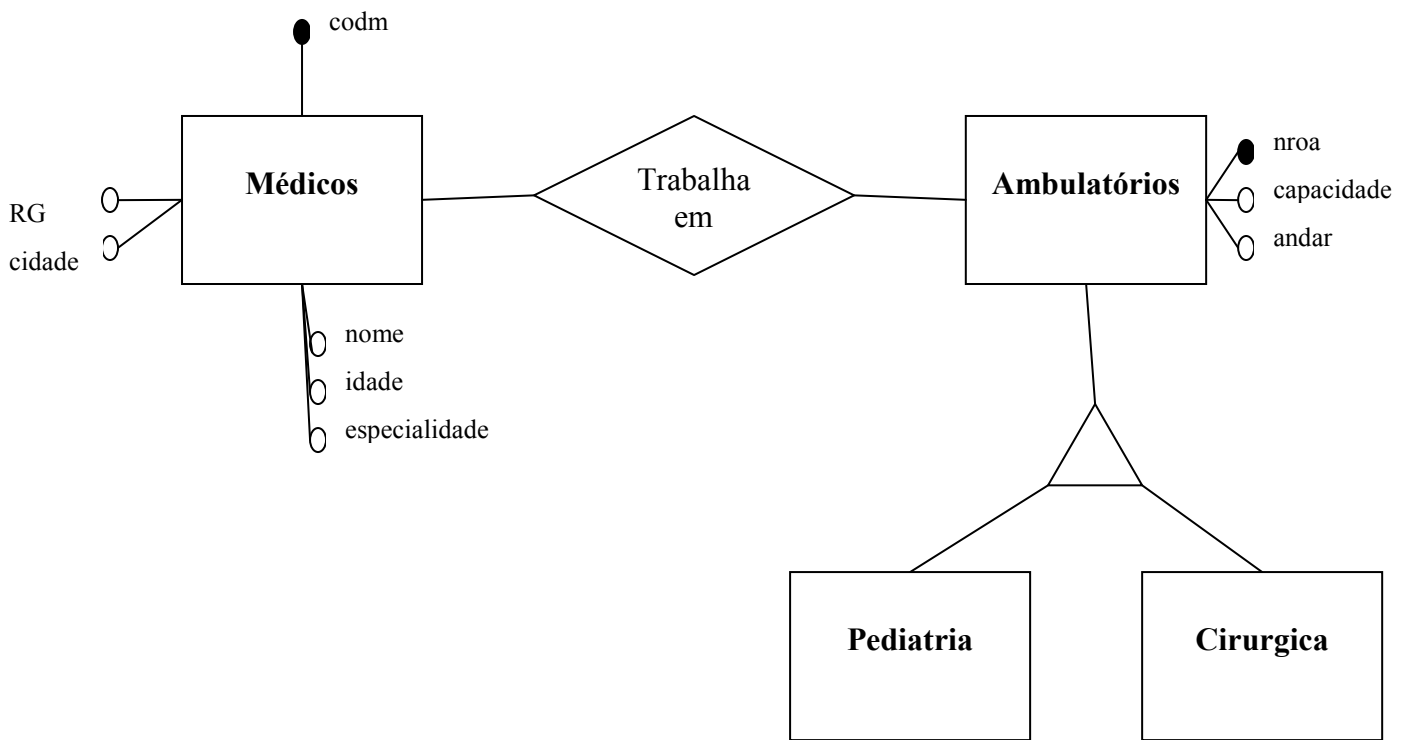


Figura 2-8 : Representação de Generalização e Especialização

2.2.2. Propriedades do Modelo E-R

Nesta seção, o modelo E-R é avaliado em relação aos principais requisitos que foram apresentados anteriormente.

- **Expressividade:** É garantida através dos conceitos apresentados (entidades, relacionamentos e atributos entre outros), pois as entidades são objetos do mundo real com suas propriedades. Relacionamentos são fatos elementares que relacionam duas ou mais entidades e atributos são as propriedades das entidades e dos relacionamentos representadas por valores;

- **Simplicidade:** Quando observamos um modelo E-R com todos os seus conceitos, este requisito fica prejudicado, mas se observarmos apenas os conceitos principais (entidades, relacionamentos e atributos) o modelo se torna de fácil interpretação;
- **Minimalidade:** Não é garantida, apesar do modelo E-R apresentar significação distinta, diminuindo a ambigüidade de representação, para cada um de seus conceitos. Por exemplo, existem casos onde a combinação de entidades e relacionamentos podem substituir um atributo e vice-versa.
- **Formalidade:** Este requisito está presente na modelagem E-R, pois todos os conceitos apresentados possuem um significado único e preciso. A garantia deste requisito permite que a ambigüidade na interpretação do modelo conceitual seja bastante reduzida.

Apesar do modelo E-R ser um dos mais completos, em geral, ele não é capaz de expressar todas as propriedades de uma determinada realidade. Algumas destas propriedades devem ser expressas através de declarações que completam a modelagem. No entanto, o número de declarações adicionais pode ser reduzido pela incorporação de conceitos mais expressivos ao modelo. A escolha do nível apropriado de complexidade do modelo é difícil, pois é interessante utilizar a maior quantidade possível de conceitos sem comprometer a simplicidade. Algumas vezes é necessário abrir mão da minimalidade de um modelo em detrimento à sobreposição de conceitos, já que uma grande quantidade rica de conceitos pode ajudar o projetista a ter uma melhor percepção da realidade. (BATINI, 1992).

2.3. ENGENHARIA REVERSA

Dentro do contexto de Projeto de Banco de Dados, a *engenharia reversa* é utilizada para “extrair o abstrato do concreto para obtenção de um entendimento conceitual do banco de dados existente” (BATINI, 1992).

De forma geral, um processo de engenharia reversa pode ser definido como um processo de abstração, que parte de um modelo de implementação e resulta em um modelo conceitual que descreve abstratamente a implementação em questão. O termo engenharia reversa vem do fato de usar como ponto de partida do processo um produto implementado (o modelo físico ou de implementação) para obter sua especificação (o modelo conceitual como mostra a figura 2.9) (HEUSER, 2001).

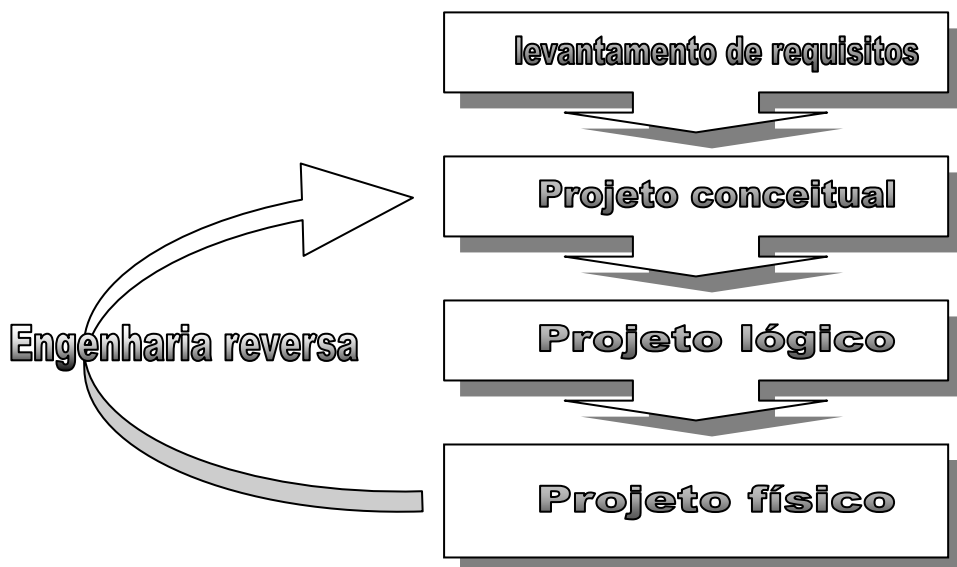


Figura 2-9: Engenharia Reversa

Este é o processo inverso ao que normalmente é executado em um Projeto de Banco de dados, onde primeiro é desenvolvido o modelo conceitual para posteriormente ser desenvolvido o modelo físico.

Cabe ressaltar que assim como em todo processo de modelagem de um determinado ambiente (realidade) utilizando o modelo E-R, surgirão diferentes modelagens para o mesmo

ambiente, uma vez que cada pessoa percebe a realidade de maneira distinta. Logo é natural que o processo de engenharia reversa produza diferentes modelos E-R de acordo com a percepção de cada usuário.

2.3.1. Porque Engenharia reversa

Surge um questionamento com relação à realização de um processo de engenharia reversa. Por que é preciso definir uma modelagem conceitual, visto que o banco de dados já existe? Esta modelagem não serve apenas para se projetar um novo banco de dados?

A resposta é não por vários motivos. Primeiro, a importância de uma documentação a nível conceitual é a necessidade de migração para um SGBD com um modelo de dados diferente do modelo do SGBD existente. A modelagem conceitual servirá de base para um novo projeto lógico para este novo SGBD a ser adotado.

Segundo, o projeto conceitual não é apenas uma etapa que deva ser executada apenas se houver necessidade. Ela é parte importante do processo de documentação do banco de dados. É imprescindível lembrar que os bancos de dados são projetados para persistirem durante anos, possivelmente décadas, e, muito provavelmente, as pessoas que trabalharam no desenvolvimento do mesmo não irão permanecer durante toda a vida útil do banco de dados.

Justamente por esta prolongada existência e funcionalidade, faz-se necessário entender que outras pessoas (desenvolvedores, analistas, programadores e usuários) necessitarão manipular a informação contida neste banco de dados, e que cada uma delas enxerga a realidade de maneira diferente. Portanto, possuem diferentes modelos mentais do fluxo da informação. Deste modo, o projeto conceitual tem grande importância dentro do ciclo de vida de um banco de dados, pois ele é parte da documentação necessária a futuras alterações efetuadas por pessoas que desconheçam a realidade que o banco de dados está abstraindo.

Por fim, esta modelagem é tão abrangente em relação à realidade que se está abstraindo que ela não serve apenas como passo no projeto de banco de dados. Ela pode também ser empregada como base para uma compreensão mais formal, e, portanto mais objetiva e menos ambígua da empresa. Uma das consequências da obtenção de um modelo conceitual é precisamente a conceituação da empresa” (SETZER, 1989).

Enfim, durante o ciclo de vida de um sistema, eventualmente vão sendo agregadas novas funcionalidades e naturalmente com a expansão das atribuições do mesmo, o seu banco

de dados também poderá precisar sofrer adaptações, alterações ou ainda migração de plataforma de implementação para ter capacidade de dar suporte às novas necessidades que passam a existir.

Para todas estas tarefas, “a disponibilidade de uma documentação abstrata, na forma de um esquema conceitual dos dados do sistema existente, pode acelerar em muito o processo de manutenção, pois permite encurtar a etapa de modelagem dos dados do novo banco de dados” (HEUSER, 2001).

Da mesma forma que a execução do Projeto de Banco de dados exige uma seqüência (ciclo), com tarefas pré-definidas durante cada etapa, o processo de engenharia reversa também possui regras que devem ser seguidas para que seja obtido um melhor resultado. Metodologia para engenharia reversa é apresentada a seguir.

2.3.2. Seleção da Metodologia para Engenharia Reversa

Dentro do projeto de banco de dados, o processo de engenharia reversa de banco de dados relacional não foi totalmente esgotado. Na verdade, muitos autores nem abordam a engenharia reversa como parte do projeto de banco de dados, em seus livros. Isto implica que a quantidade e diversidade de metodologias para este processo ficam muito aquém daquela encontrada para o projeto de banco de dados, onde o material disponível é mais amplo.

Para este trabalho duas metodologias foram estudadas, uma apresentada por (BATINI 1992) e a outra por (HEUSER 2001).

Para que seja possível a utilização da metodologia de BATINI algumas condições devem ser satisfeitas:

As relações devem estar normalizadas (Terceira forma normal);

Chaves primárias e estrangeiras em diferentes relacionamentos que possuam nomes idênticos devem ser renomeadas, de forma que não produzam ambigüidade;

Atributos com o mesmo nome devem possuir o mesmo domínio, caso possuam mesmo nome mas com domínios diferentes devem ser apropriadamente renomeados;

Os relacionamentos devem estar classificados conforme segue abaixo, observe que esta classificação não é completa mas abrange os casos mais comuns:

O processo de engenharia reversa de BATINI inicia-se pela classificação das relações, conforme mostrado a seguir:

1. **Relações primárias:** Esta é uma relação onde a chave primária não contém nenhuma chave estrangeira. Ela é convertida em uma entidade;
2. **Relação primária fraca:** Se uma chave primária de uma relação contém a chave primária de uma outra relação ela é chamada de relação primária fraca;
3. **Relação secundária:** É a relação que a chave primária é formada por uma concatenação de chaves primárias de outras relações.

Depois de realizada esta classificação é necessário então seguir uma seqüência de passos conforme mostrado a seguir:

Passo 1:

Pré-processamento e classificação das relações: Os relacionamentos devem ser analisados para ver se cumprem todas as pré-condições;

Passo 2:

Mudança de chave primária: O objetivo deste passo é preparar os relacionamentos para os passos seguintes, que dependem da detecção de relacionamentos entre as chaves primárias. Se a chave primária de uma entidade é igual a uma possível chave primária de outras entidades, então as outras chaves primárias devem ser substituídas por esta nova;

Passo 3:

Atribua nomes apropriados para as partes da chave primária que ocorram em relacionamentos secundários de forma que removam ambigüidades: Esse passo envolve relacionamentos secundários, aqueles nos qual a chave primária de uma entidade é composta por mais de uma chave estrangeira, e o objetivo é remover qualquer tipo de ambigüidade que possa acontecer;

Passo 4:

Mapeamento de relacionamentos primários em entidades: Nesta etapa todos os relacionamentos que estiverem classificados como Relacionamento Primário serão mapeados em entidades;

Passo 5:

Mapeamento de relacionamentos primários fracos: Todos os relacionamentos que foram classificados como Primários Fracos, devem ser mapeados para entidades;

Passo 6:

Detectar conjuntos de relacionamentos entre entidades: Uma análise mais profunda do esquema relacional deve ser feita neste momento a fim de que especializações/generalizações sejam detectadas e implementadas.

Passo 7:

Mapeamento de relacionamentos secundários em relacionamentos entre entidades: A presença de um relacionamento secundário sugere um relacionamento entre as entidades correspondentes no nível conceitual. Cada relacionamento secundário é mapeado em um relacionamento entre as entidades nas quais as chaves estão concatenadas para formar a chave primária.

Passo 8:

Mapear as restrições de integridade de atributos não chaves em relacionamentos entre entidades: A presença de uma chave estrangeira indica a existência de um relacionamento entre as duas entidades correspondentes.

Passo 9:

Mapeamento de relações não classificadas: Grande parte das situações mais comuns foram tratadas por esta metodologia, os casos remanescentes devem ser analisados e processados individualmente.

BATINI não faz a classificação de chaves primárias que possuam uma concatenação de chaves primárias de outras relações e também de atributos, o que acontece com alguma regularidade em bancos de dados. Apesar de propor no passo 9 (nove) de sua metodologia o mapeamento das relações não classificadas, ele não define formas de como executar esta operação ficando a cargo do projetista identificar e classificar caso a caso, o que torna o trabalho bastante desgastante e não permitindo sua automatização.

Em seu passo 7 (sete), BATINI comenta que “não há forma automática para determinar a cardinalidade mínima e máxima das entidades participantes em um recém-criado

relacionamento”. No entanto, o mesmo não apresenta sugestões de como atribuir valores para as cardinalidades.

Por estes motivos, optou-se pela metodologia apresentada por (HEUSER 2001), que é mais adequada ao trabalho proposto e é detalhada a seguir.

2.3.3. Apresentação da metodologia escolhida

A metodologia utilizada para o processo de engenharia reversa de banco de dados relacionais apresentada pela ferramenta segue as especificações apresentadas por (HEUSER, 2001), que divide todo o processo em quatro etapas:

- 1. Identificação da construção E-R correspondente a cada tabela;**
- 2. Definição de relacionamentos 1:n e 1:1;**
- 3. Definição de atributos;**
- 4. Definição de identificadores de entidades e relacionamentos.**

Estas etapas são detalhadas nas seções que seguem.

1. Identificação da construção E-R correspondente a cada tabela

Nesta primeira etapa define-se, para cada tabela do modelo relacional, qual a construção correspondente a nível de modelo E-R.

Uma tabela pode corresponder a:

- Uma entidade;
- Um relacionamento n:n;
- Uma entidade especializada.

O fator determinante da construção E-R que corresponde a uma tabela é a composição de sua chave primária. Tabelas podem ser classificadas de acordo com sua chave primária:

- **Regra 1: Chave primária composta por mais de uma chave estrangeira**

A tabela que possui uma chave primária composta de múltiplas chaves estrangeiras implementa um *relacionamento n:n* entre as entidades

correspondentes às tabelas referenciadas pelas chaves estrangeiras. Ver exemplo na figura 2.10;

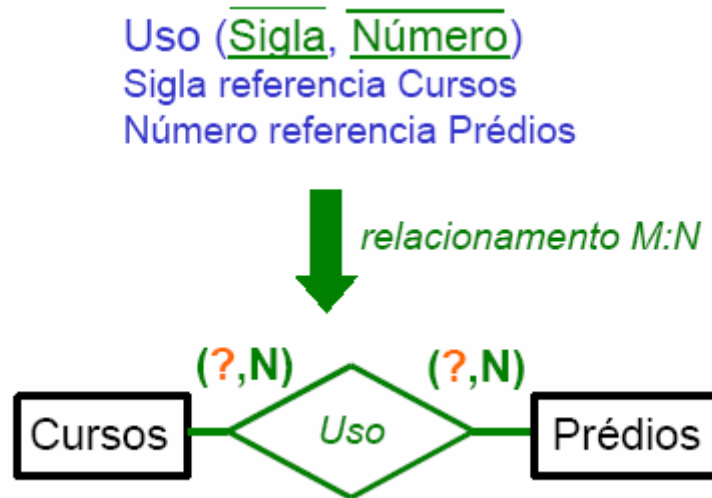


Figura 2-10 : CP composta por várias CE's

- **Regra 2: Toda chave primária é uma chave estrangeira**

A tabela cuja chave primária é toda ela uma chave estrangeira representa uma entidade que forma uma *especialização* da entidade correspondente à tabela referenciada pela chave estrangeira. Ver exemplo na figura 2.11;

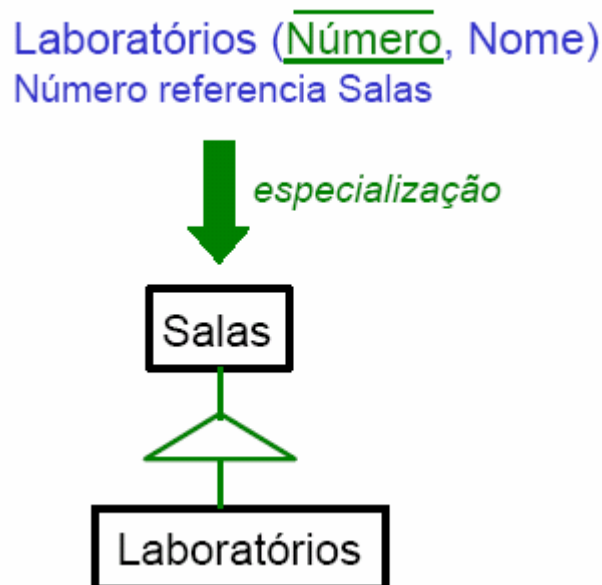


Figura 2-11 : Relacionamento de Especialização

- **Regra 3: Demais casos**

Quando a chave primária da tabela não for composta de múltiplas chaves estrangeiras (regra 1), nem for toda ela uma chave estrangeira (regra 2), a tabela representa uma entidade que pode conter:

Um atributo multivalorado, isto ocorre devido à existência de apenas uma CP e uma CE (ver figura 2.12).



Figura 2-12 : Atributo Multivalorado

Torna-se uma entidade fraca quando existe uma CP e pelo menos uma CE na tabela (ver figura 2.13).

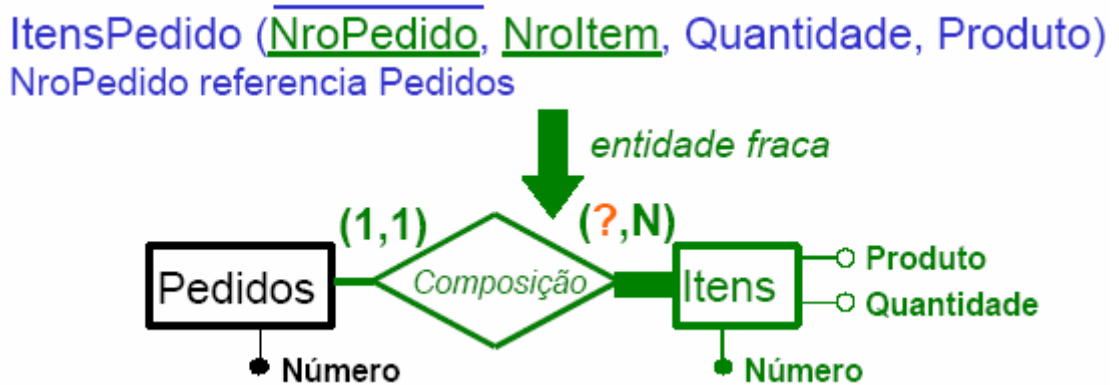


Figura 2-13 : Representação de uma Entidade Fraca

Torna-se uma hierarquia de Especialização, a existência de um atributo de qualificação confirma este tipo de mapeamento (ver figura 2.14).

Salas (Número, Andar, Vagas, Tipo, Prédio)
Prédio referencia Prédios

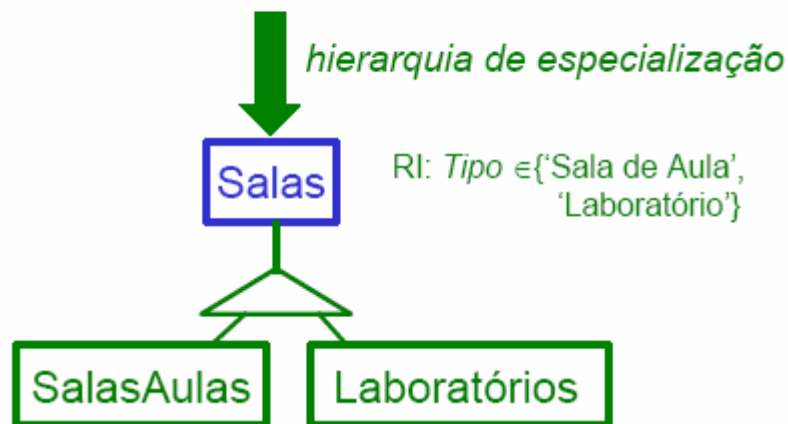


Figura 2-14: Representação de Hierarquia de Especialização

Torna-se uma entidade forte quando nenhuma das situações anteriores é atendida.

Prédios (Número, Andares, Centro)
Centro referencia Centros



Figura 2-15: Representação de uma Entidade Forte

2. Identificação de relacionamentos 1:n ou 1:1

Toda chave estrangeira que não se enquadra nas regras 1 e 2 apresentadas na seção anterior, ou seja, toda chave estrangeira que não faz parte de uma chave primária composta por múltiplas chaves estrangeiras, nem é toda ela uma chave primária, representa um relacionamento 1:n ou 1:1. Em outros termos, toda chave estrangeira que não corresponde a um relacionamento n:n, nem a uma entidade especializada representa um relacionamento 1:n ou 1:1. A regra não permite definir se a cardinalidade do relacionamento é 1:n ou 1:1. Para definir qual dos dois tipos de relacionamentos está sendo representado pela chave estrangeira, é necessário verificar os conteúdos do banco de dados.

Neste caso são analisadas as CE's que não se enquadram no tópico anterior. Para a definição das cardinalidades dos relacionamentos é necessária a investigação dos dados do banco de dados. São duas as alternativas neste caso:

- **Regra 1:** Relacionamento binário, ver figura 2.16.

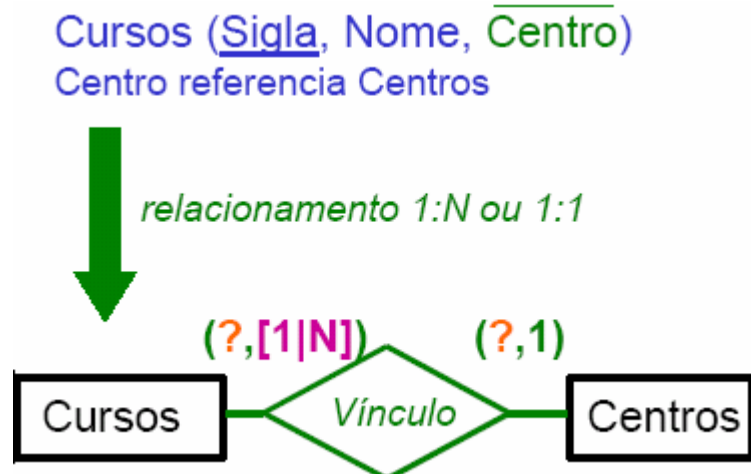


Figura 2-16 : Representação de um Relacionamento Binário

- **Regra 2 :** Entidade Associativa (tabela com CP composta), conforme figura 2.17.

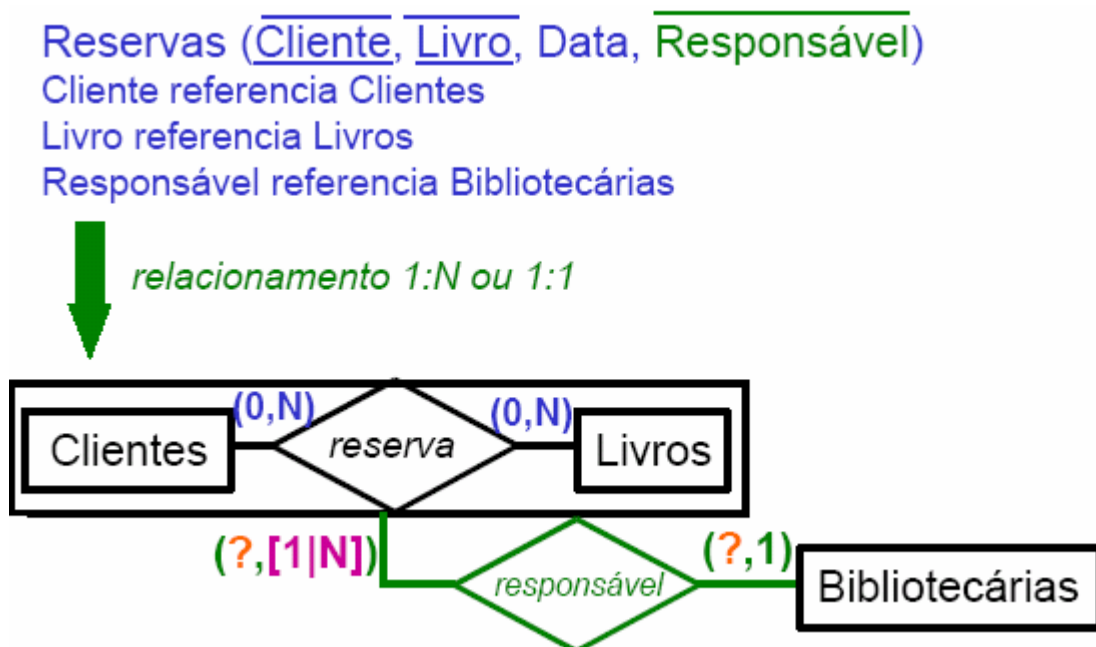


Figura 2-17 : Representação de uma Entidade Associativa

3. Definição de atributos

Nesta etapa, para cada coluna de uma tabela, que não seja chave estrangeira, é definido um atributo na entidade ou no relacionamento correspondente à tabela. Observe-se que colunas chave estrangeira não correspondem a atributos no diagrama E-R, mas sim a relacionamentos, e por isso já foram tratadas nas etapas anteriores.

- **Regra 1** : O atributo pertence a um relacionamento da Entidade, ocorre com atributos de tabelas com CP composta por várias CE's ver figura 2.18.

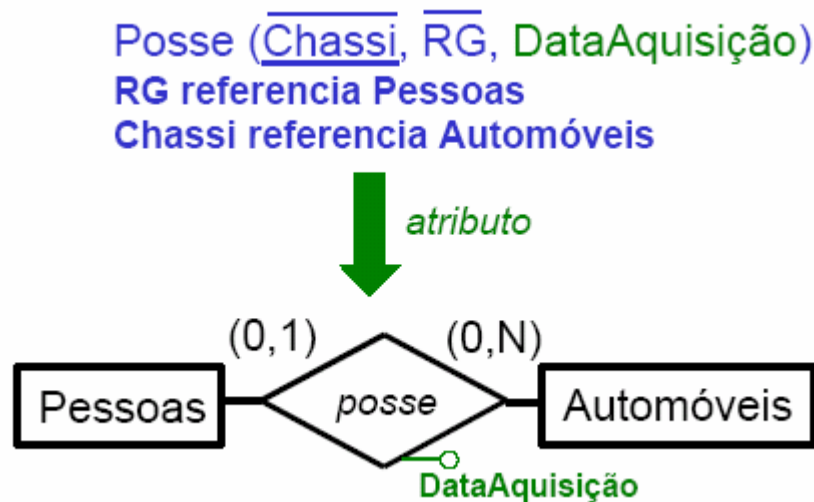


Figura 2-18 : Atributo pertencente a um relacionamento da Entidade

- **Regra 2** : O atributo pertence a uma entidade especializada, ocorre quando a tabela encapsula uma hierarquia de especialização, existe a necessidade de analisar os dados na tabela para concluir a qual entidade pertencem os atributos (ver figura 2.19).

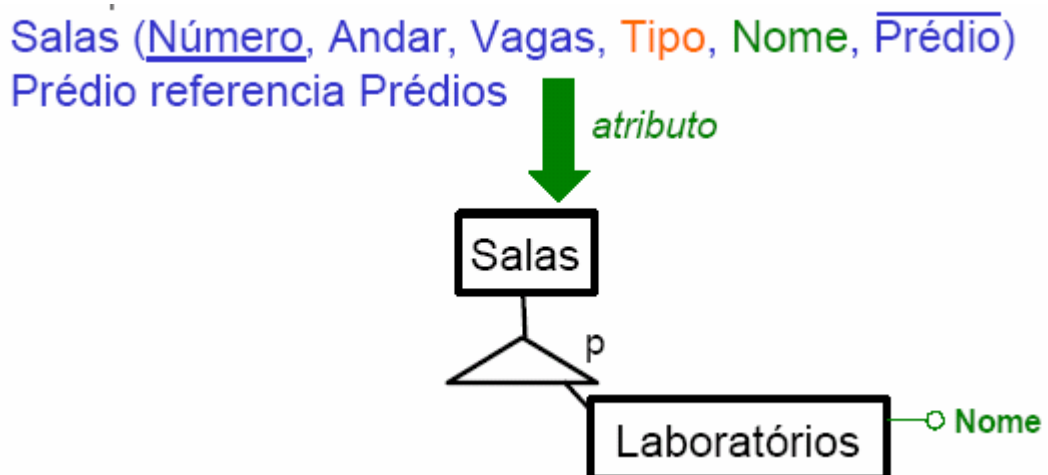


Figura 2-19 : Atributo pertencente a uma Entidade Especializada

- **Regra 3** : O Atributo pertence à entidade, quando nenhuma das duas alternativas anteriores se aplica (ver figura 2.20).

Centros (Sigla, Nome, Localização)
Prédio referencia Prédios



Figura 2-20 : Atributo pertencente à Entidade

4. Definição de identificadores e entidades

Neste último passo são definidos os identificadores das entidades e dos relacionamentos. A regra para definição dos identificadores é a seguinte:

- **Coluna da chave primária que não é chave estrangeira**
Toda coluna que faz parte da chave primária e que não é chave estrangeira corresponde a um atributo identificador da entidade ou relacionamento. Ver figuras 2.21 e 2.22.

Estantes (Corredor, Número,
NroPrateleiras)



Figura 2-21 : Identificador de Entidade

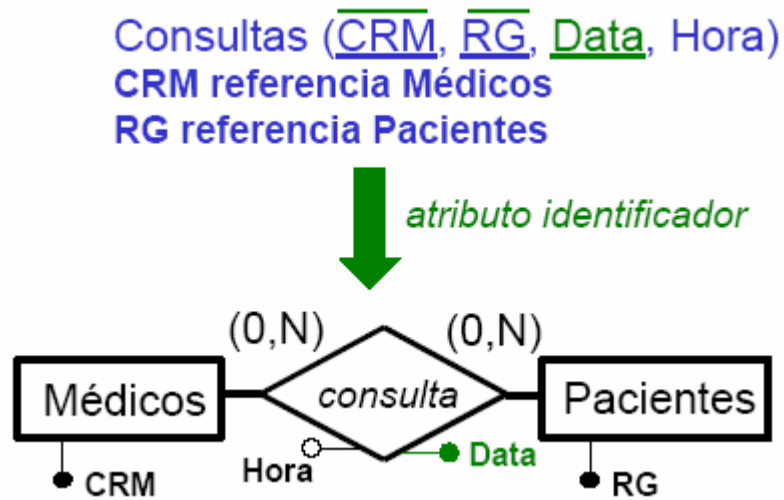


Figura 2-22 : Identificador de Relacionamento

- **Coluna da chave primária que é chave estrangeira**
 Toda coluna que faz parte da chave primária e que é chave estrangeira corresponde a um identificador externo da entidade. Ver figura 2.23.

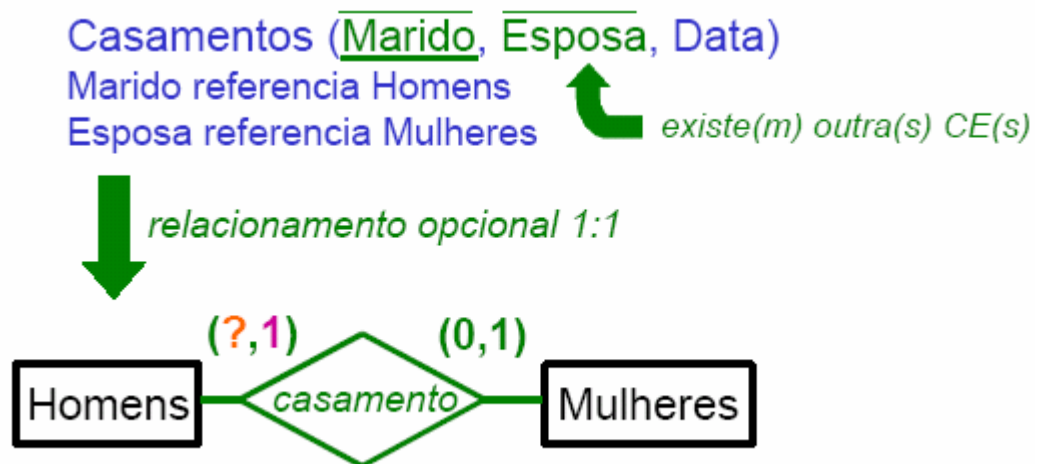


Figura 2-23 : Identificador Externo – Chave Estrangeira

Cabe lembrar que a ferramenta não conseguirá representar entidades associativas ou agregação. Esta representação pode substituída pela constituição de um relacionamento 1:N ou N:N.

3. DESCRIÇÃO DA FERRAMENTA

A ferramenta proposta neste trabalho de conclusão de curso foi implementada para auxiliar a execução da metodologia de engenharia reversa apresentada no capítulo anterior. A ferramenta foi implementada na linguagem de programação Java. A opção por esta linguagem está relacionada a grande disseminação que esta linguagem possui atualmente e o seu alto nível de segurança.

O ambiente de desenvolvimento Java, ou também chamado de IDE, utilizado neste trabalho foi o *Eclipse Platform* (www.eclipse.org). O Eclipse é uma plataforma de código aberto que tem seu foco no auxílio ao desenvolvimento de *softwares*. O Eclipse prove recursos como ferramentas e *frameworks* que facilitam e auxiliam durante o ciclo de vida de construção de um *software*. Alguns destes recursos são: Suporte a Modelagem, Ambiente para desenvolvimento utilizando linguagens Java, C, C++ dentre outras, testes e análise de performance, *business intelligence*, etc. Eclipse é a comunidade que criou a Plataforma Eclipse. Ela é formada por universidades, institutos de pesquisas e pesquisadores. Todos eles complementam e dão suporte a Plataforma Eclipse.

Para um melhor entendimento das funcionalidades e recursos, da ferramenta desenvolvida, abaixo elaboramos uma tabela comparativa entre as ferramentas existentes mais famosas e a ferramenta implementada neste trabalho de conclusão de curso.

Características	DBDesigner	ERWin	G2ER
Produzida por:	Fabulous Force Database tools	Computer Associates	Trabalho de conclusão de curso
Disponível em:	www.fabforce.net	www.ca.com	www.inf.ufsc.br/~caon/g2er (em breve)
Software:	Livre	Pago	Livre
Plataformas:	Multi-plataforma	Windows	Multi-plataforma
Engenharia Reversa:	Automática	Automática	Semi-automática
Características do ER:	Simples diagrama com o relacionamento entre	Simples diagrama com o relacionamento	Ótimo visual dispendo de todas as características expostas por Peter Chen

	as tabelas	entre as tabelas	exceto entidade associativa
Suporte a SQL:	MySQL, Oracle, SQL Server, todo que suporte acesso ODBC	SQL Server, Oracle, Dbase e outras	MS SQL Server
Persistência:	Salva arquivos em xml	Salva arquivos em xml	Salva arquivos em xml

Observando a tabela acima com um comparativo da ferramenta desenvolvida neste trabalho com 2 outras ferramentas muito populares podemos destacar alguns pontos importantes:

A ferramenta G2ER provê somente suporte ao SQL do SQL Server isto devido ela estar em seu estado inicial de desenvolvimento, enquanto o DBDesigner e o ERWin dão suporte a grande maioria de linguagens SQL Esta dificuldade pode ser resolvida através da criação de novos *parsers* que interpretem os diferentes tipos de linguagens SQL.

Dois pontos fortes também observados na tabela acima em favor da ferramenta G2ER é a parte de Engenharia Reversa, que ocorre de forma semi-automática favorecendo dando muita vantagem ao usuário, visto que, nas outras ferramentas elas geraram o modelo ER de forma automática de acordo com os padrões previamente implementados nas mesmas, enquanto que na G2ER a geração do modelo ER se diferencia de acordo com o domínio do problema, isto ocorre no momento que existem questões com mais de uma alternativa, é onde ocorre o questionamento ao usuário, tornando o processo de engenharia reversa mais lento que os outros, mas muito mais agradável ao usuário.

O outro ponto importante é a parte gráfica da ferramenta que enquanto nas outras o modelo ER somente demonstra os relacionamentos com nomes entre as tabelas com os seus respectivos atributos, na ferramenta G2ER a modelagem ER é quase que completa segundo o modelo apresentado por Peter Chen, somente não ocorrendo a representação da entidade associativa, que pode ser substituída por um relacionamento ternário. Isto com certeza facilita muito o entendimento do modelo pelo usuário.

Agora iremos detalhar todo o funcionamento da ferramenta G2ER exposta acima.

A ferramenta funciona da seguinte forma: O usuário passa para a ferramenta um *script SQL*. O script SQL é o código de implementação do projeto físico de um banco de dados relacional. Este script é a fonte da informação necessária para que a ferramenta possa então gerar o esquema E-R.

Uma vez de posse deste *script*, a ferramenta interage com o usuário sempre que necessário, para poder gerar o esquema E-R correspondente. Para que essa geração seja possível, ela efetua o tratamento das instruções SQL que foram previamente reconhecidas por analisadores léxico e sintático.

Uma vez reconhecidas as tabelas e atributos do *script*, cabe então a ferramenta, no que for possível, implementar de forma automática as etapas apresentadas na metodologia de engenharia reversa. Uma vez definido o esquema conceitual, este é apresentado graficamente ao usuário.

A figura 3.1 abaixo mostra as principais etapas da ferramenta:

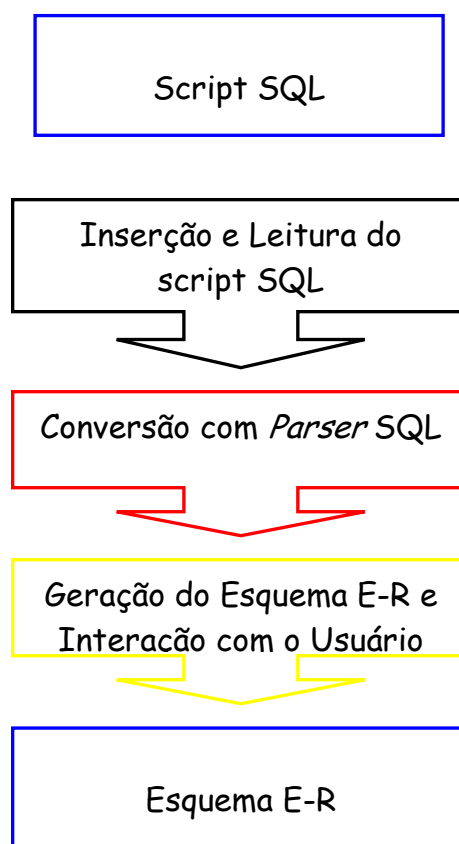


Figura 3-1 : Etapas da Ferramenta

Estas etapas são detalhadas nas seções a seguir.

3.1. Inserção e Leitura do Script SQL

Este passo é caracterizado pela abertura do *script SQL* na interface da ferramenta em uma área que funciona como um editor de texto. Neste editor, as palavras reservadas, que são as palavras que possuem relevância para a concretização do esquema E-R, aparecem em cores diferentes das demais.

Nesta primeira versão, a ferramenta é capaz de ler somente scripts SQL do SGBD Microsoft SQL Server. Optamos inicialmente por este padrão por este ser o SGBD utilizado nas disciplinas de Banco de Dados do curso de Sistemas de Informação da UFSC. Atualmente, a ferramenta é capaz de analisar apenas instruções de criação de tabelas. Outras instruções são ignoradas por ela.

A Figura 3.2 mostra a interface da ferramenta para inserção e visualização de scripts SQL.

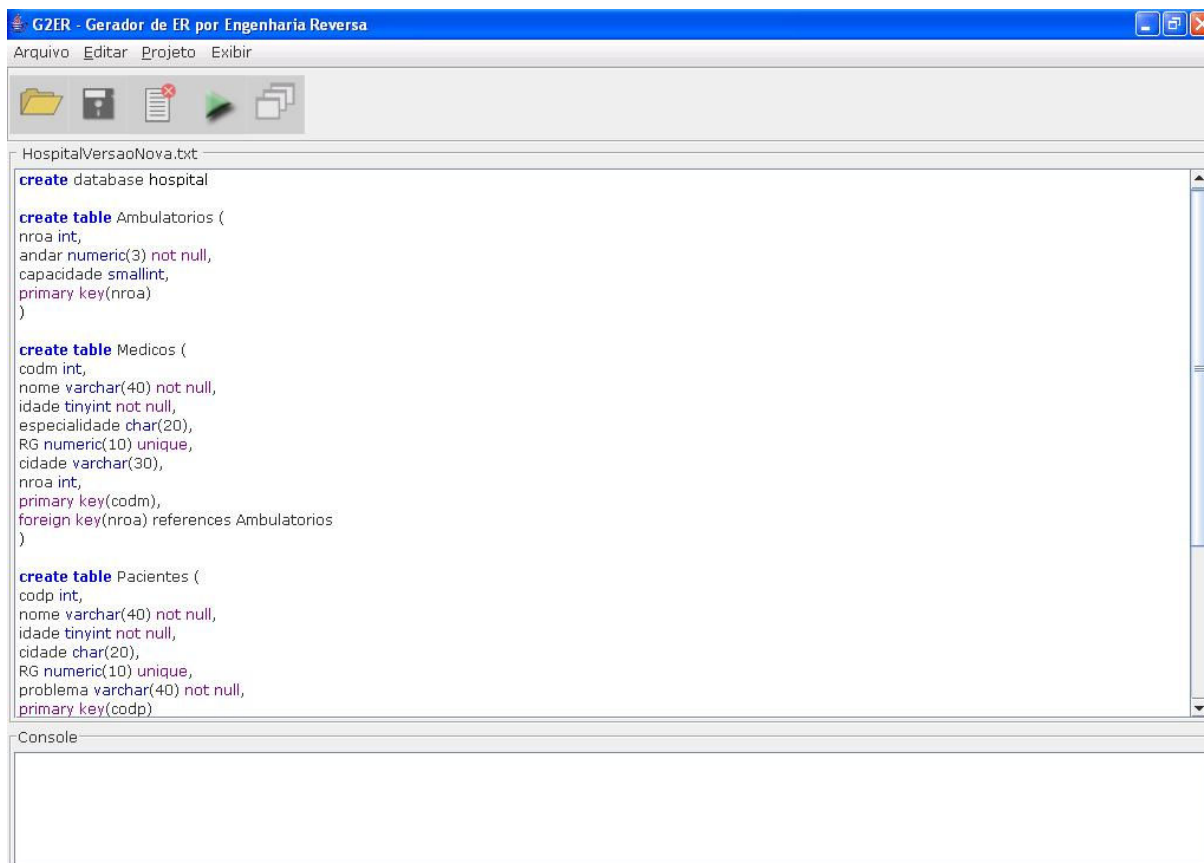


Figura 3-2 : Tela de Inserção e Leitura do Script SQL

Visando facilitar o entendimento do funcionamento da ferramenta utilizamos como exemplo de uso o esquema físico (script SQL) mostrado a seguir:

create database hospital

```
create table Ambulatorios (  
  nroa int,  
  andar numeric(3) not null,  
  capacidade smallint,  
  tipo varchar(40),  
  primary key(nroa)  
)
```

```
create table Medicos (  
  codm int,  
  nome varchar(40) not null,  
  idade tinyint not null,  
  especialidade char(20),  
  RG numeric(10) unique,  
  cidade varchar(30),  
  nroa int,  
  primary key(codm),  
  foreign key(nroa) references Ambulatorios  
)
```

```
create table Pacientes (  
  codp int,  
  nome varchar(40) not null,  
  idade tinyint not null,  
  cidade char(20),  
  RG numeric(10) unique,  
  problema varchar(40) not null,  
  primary key(codp)  
)
```

```
create table Consultas (  
  codm int,
```

```
codp int,  
data datetime,  
hora datetime,  
primary key(codm,codp,data),  
foreign key(codm) references Medicos,  
foreign key(codp) references Pacientes  
)
```

3.2. Conversão dos Dados Lidos usando um *Parser* SQL

A ferramenta deve entender o que significa cada “palavra” do script SQL. Por isso, foi criado um *Parser* SQL.

Este *Parser* SQL reconhece cada “palavra” do script SQL e principalmente possui a coleção de ações a serem tomadas pela ferramenta quando cada uma destas “palavras” é interpretada por ela.

O *Parser* SQL foi implementado na linguagem Java, como a ferramenta, e pode ser mais bem entendido como um “Tradutor” que informa a ferramenta o que ela deve fazer conforme vai “lendo ou interpretando” o script SQL. Abaixo escrevemos mais especificamente sobre o *Parser* SQL.

3.2.1. *Parser* SQL

O *parser* SQL foi implementado através de uma ferramenta chamada GALS (Gerenciador de Analisador Léxico e Sintático). O GALS é um ambiente para a geração de analisadores léxicos e sintáticos, desenvolvido por Carlos Eduardo Gesser como trabalho de conclusão de curso do Curso de Bacharelado em Ciências da Computação, da Universidade Federal de Santa Catarina, sendo desenvolvido sob a orientação do Prof. Olinto José Varela Furtado (2003).

A função de um analisador léxico é ler o código fonte, no caso desta ferramenta, um arquivo de script SQL, e identificar os caracteres em itens lógicos chamados de “tokens”. Um *token* é representado por identificadores, palavra reservadas ou constantes. Neste trabalho, um exemplo disso seria um “create” ou uma “table”. O analisador léxico também tem a função de

detectar e diagnosticar possíveis erros léxicos como, por exemplo, símbolos inválidos. Outra função do analisador léxico é ignorar elementos sem valor sintático, como por exemplo, uma instrução de consulta SQL, que é irrelevante para a ferramenta.

O analisador sintático tem a função de agrupar em estruturas sintáticas os *tokens* identificados pelo analisador léxico. O comando “create table” é um exemplo disto, ou seja, a estrutura sintática “create table” tem uma função dentro da SQL. Outras funções do analisador sintático são verificar se a ordem das estruturas sintáticas da linguagem está sendo respeitada no programa analisado e identificar e diagnosticar possíveis erros sintáticos.

Cabe ressaltar que a seção de definição da linguagem SQL (DDL) é muito ampla e complexa. Em função disto, diversas expressões foram suprimidas da gramática desta linguagem pelo *parser* SQL, como a definição de *triggers* e *procedures* e instruções de atualização e remoção de definições de tabelas.

Assim sendo, o escopo da gramática está atualmente limitado ao reconhecimento do comando *create table* e seus relacionados, pois ele é o responsável pela criação das tabelas com seus atributos e chaves primárias e estrangeiras.

A integração do *Parser* SQL gerado pelo GALS com a ferramenta desenvolvida neste trabalho é bem simples. O GALS gera as classes em linguagem Java do *Parser* e as mesmas são integradas junto ao *pool* de classes da ferramenta. Os métodos existentes nas classes do *Parser* são executados de acordo com as chamadas feitas pela classe de controle da ferramenta.

Por exemplo: A ferramenta executa o método de leitura do script SQL, palavra por palavra, e então todas as palavras são passadas via parâmetro para as classes do *Parser* SQL que então retornara a ferramenta o significado desta palavra e conseqüentemente também com a ação que deve ser tomada pela ferramenta.

3.3. Geração do Esquema E-R e Interação com o Usuário

Após a análise do *script* SQL pelo *parser* e corrigidos todos os possíveis erros, inicia-se a geração do esquema E-R correspondente às tabelas do script SQL. Durante esta geração podem ocorrer diversas interações com o usuário. Estas interações ocorrem somente quando

existe mais de uma opção para determinada conversão de um conceito a nível modelo lógico¹ para um conceito a nível conceitual. A escolha por uma opção ou outra deve levar em conta a semântica do domínio em questão e consideramos que somente o usuário tem capacidade de determinar exatamente esta semântica.

A seguir temos telas da ferramenta exemplificando a interação com o usuário para a geração do esquema E-R de do domínio hospitalar utilizado por nós como exemplo.

A Interação abaixo (figura 3.4) foi gerada devido a ferramenta executar o seguinte script:

```
create table Medicos (  
  codm int,  
  nome varchar(40) not null,  
  idade tinyint not null,  
  especialidade char(20),  
  RG numeric(10) unique,  
  cidade varchar(30),  
  nroa int,  
  primary key(codm),  
  foreign key(nroa) references Ambulatorios  
)
```

Nota-se que a chave primária da tabela não é composta por várias ou por uma única chave estrangeiras, isto então, leva a possibilidade do usuário ter mais de uma representação para a tabela acima. Neste caso optamos por transformar a tabela em uma entidade forte o que ocorre em 90% dos casos e é a situação que mais se encaixa a nossa necessidade.

¹ Apesar de estarmos lidando com uma especificação em SQL (uma implementação física), chamamos os conceitos de lógicos, pois estamos na verdade convertendo conceitos lógicos do modelo relacional implementados na SQL para conceitos do modelo E-R.

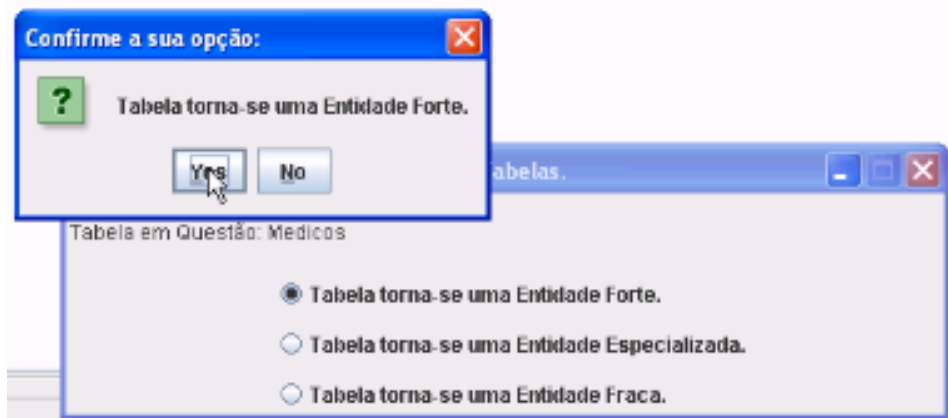


Figura 3-3 : Interação com o Usuário

Neste outro caso de interação (figura 3.4) ele ocorrerá sempre que existir um relacionamento entre tabelas. Esta interação serve para definir a cardinalidade do relacionamento e também o nome desse relacionamento. Observar script SQL abaixo.

```

create table Ambulatorios (
...
primary key(nroa)
)

create table Medicos (
...
primary key(codm),
foreign key(nroa) references Ambulatorios
)

```

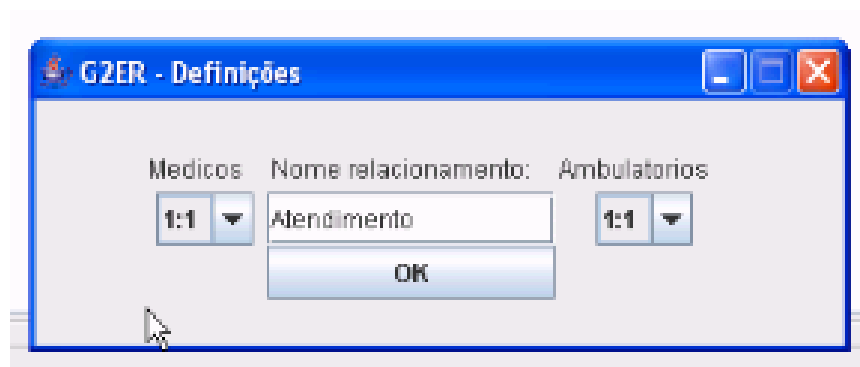


Figura 3-4 : Interação com o Usuário

Após responder a todas as interações exibidas pela ferramenta podemos então gerar o esquema ER do script em questão.

3.4. Apresentação do esquema E-R gerado

Uma vez gerado o esquema E-R, ele é apresentado graficamente ao usuário.

Para a geração do diagrama E-R foi utilizada uma biblioteca de recursos gráficos do Eclipse. Ela se chama *draw2d* e é utilizada para construção de componentes gráficos com o SWT. Através destes componentes gráficos é que foi possível criar as entidades, relacionamentos e outros conceitos necessários a representação gráfica de um esquema E-R.

O *draw2d* tem capacidade de gerenciamento de layout, ou seja, ele controla de o posicionamento das figuras geradas de modo que elas não fiquem umas sobre as outras ou ainda amontoadas em determinado local da tela. Combinando as bordas e as figuras corretas pode-se criar qualquer tipo de relacionamento de gráficos como o de Entidade-Relacionamento usado nesta ferramenta.

Por fim o esquema E-R gerado poderá ser armazenado em um arquivo no formato ainda a ser definido.

A ferramenta possui algumas peculiaridades com respeito ao esquema E-R gerado (ver figura 3.5).

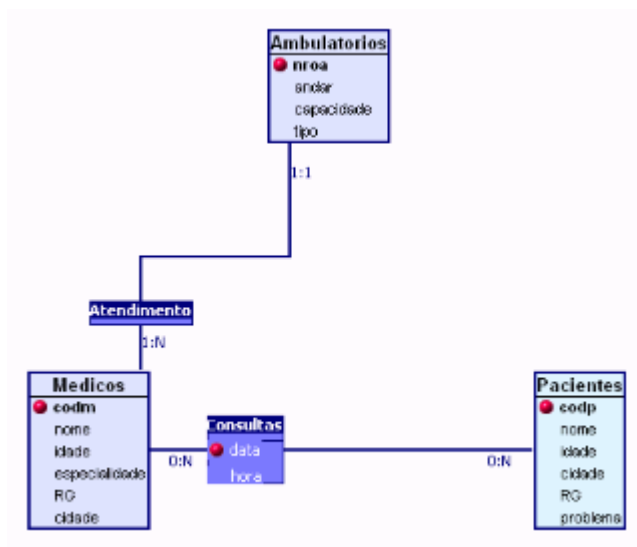


Figura 3-5 : Esquema E-R gerado

Para representar as entidades, a ferramenta assume o mesmo padrão de outras ferramentas muito conhecidas, o DBDesigner e o ERWin.

As entidades são representadas por retângulos onde no topo aparece o nome da Entidade, logo abaixo aparecem os atributos chaves, ou chaves primárias as quais possuem uma bolinha vermelha na frente do seu nome para também ajudar na sua identificação além de terem seus nomes também escritos em negrito. Para finalizar os outros atributos, não chaves, aparecem escritos.

A representação dos relacionamentos é feita por um retângulo ligado a cada entidade que se relaciona. Dento desse retângulo é escrito o nome do relacionamento atribuído pelo usuário quando da interação como também os possíveis atributos que possam existir pertencentes ao relacionamento. Nas extremas das ligações dos relacionamentos ficam representados as cardinalidades dos mesmos, que também são informadas pelo usuário através das interações que ocorrem previamente.

Para a representação da Especialização é usado um triângulo ligado às entidades, tipo de representação amplamente usada (ver figura 3.6).

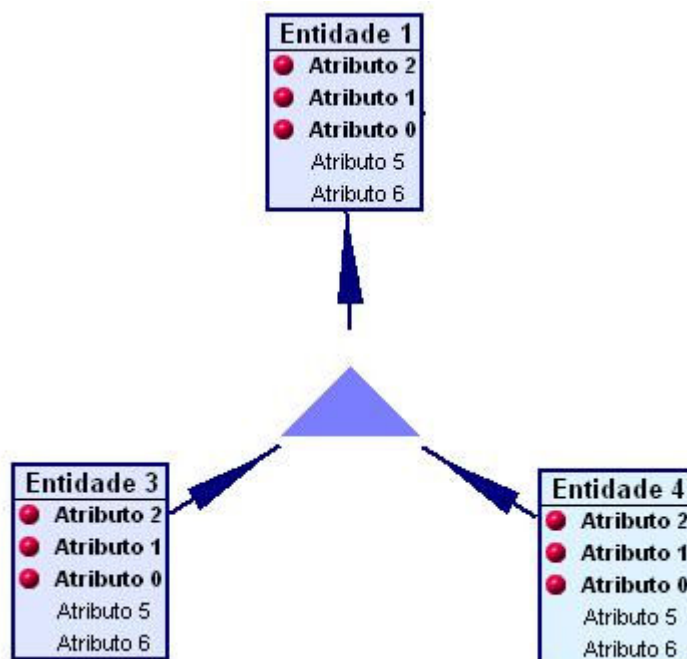


Figura 3-6 : Representação Entidade Especializada

A representação das entidades fracas é feita da mesma forma de um relacionamento normal, mas a linha usada é muito mais espessa, como se fosse um negrito, forma de representação também amplamente usada (ver figura 3.7).



Figura 3-7 : Representação da Entidade Fraca

A ferramenta não tem a capacidade ainda de representação de uma entidade associativa, mas para que isto não ficasse sem uma possível representação no esquema ER, utilizamos um relacionamento ternário que possui o mesmo efeito.

3.5. Resultados e Contribuições

Depois de finalizada a ferramenta e tendo a usado algumas vezes podemos destacar alguns pontos interessantes como:

A ferramenta atende a expectativa do usuário no processo de realização de engenharia reversa de um BD relacional, visto que a ferramenta tem a capacidade de moldar o esquema ER de acordo com a necessidade e vontade do usuário, isto é possível através das interações implementadas no processo de montagem do esquema ER o que é um ponto muito peculiar

desta ferramenta em relação às outras existentes no mercado que fazem todo o processo de forma automática sem a participação do usuário.

A ferramenta assim provê facilidades ao usuário nas tarefas que requeiram engenharia reversa, como adaptações e aprimoramentos a serem realizados em um BD relacional ou a migração para um outro SGBD.

Tudo isto é possível apenas através da leitura do script SQL (esquema físico) do BD relacional já existente.

4. CONSIDERAÇÕES FINAIS

O processo de engenharia reversa de um banco de dados relacional é muito importante, pois corrige vários problemas anteriores de projeto.

Ele auxilia o usuário/desenvolvedor no processo de construção de uma documentação não existente da modelagem conceitual que é utilizada quando de possíveis alterações ou adaptações no banco de dados relacional ou ainda quando da migração do esquema do banco de dados para outro SGBD.

Este processo é muito importante também devido ao fato de que os bancos de dados de uma organização existem ou duram por muito tempo e não podemos garantir que a mesma pessoa que o desenvolveu será a mesma a trabalhar com ele indefinidamente. Por isso, o processo de especificação do projeto de um banco de dados é tão importante, fazendo com que cada usuário, desenvolvedor, programador, analista, etc possa em qualquer momento aprender e entender todo o processo de projeto e funcionamento do banco de dados.

No intuito então de alcançar as vantagens descritas acima é que foi desenvolvida esta ferramenta de auxílio à engenharia reversa de banco de dados relacionais. A ferramenta tenta automatizar ao máximo o processo para o usuário, sendo necessário ao mesmo somente informar o *script* SQL do banco de dados relacional e ainda interagir com a ferramenta em questões importantes relativas ao domínio do banco de dados em questão.

4.1. TRABALHOS FUTUROS

Como sugestões de trabalhos futuros descrevemos algumas melhorias a serem feitas na ferramenta proposta.

Primeiramente, a ferramenta pode ser estendida para interpretar outros tipos de *scripts* SQL além do padrão do Microsoft SQL Server, como os padrões do Oracle e do MySQL. Para isto deverão ser criados novos *parsers* específicos para estes padrões.

Outro trabalho futuro a fazer é armazenar os esquemas E-R gerados em algum tipo de arquivo, como por exemplo, arquivos XML.

A idéia do armazenamento no formato XML se justifica pelo fato deste ser o padrão mais utilizado atualmente para representação e transferência de dados e também um padrão entendido por ferramentas de projeto de banco de dados como DBDesigner e ERWin.

REFERÊNCIAS

BATINI, Carlo; Ceri, Stefano; Navathe, Shamkant B. *Conceptual Database Design: An Entity-Relationship Approach*, Benjamin/Cummnigs 1992.

CHEN, Peter. *The Entity-Relationship Model: Toward the unified view of data* ACM Press New York, NY, USA.

COUGO, Paulo. *Modelagem Conceitual e Projeto de Banco de Dados*, Campus 1997.

DATE, C. J., *Introdução a Sistemas de Bancos de Dados*, Campus 7 ed. 2000.

ECLIPSE, Open Source Community - www.eclipse.org, acessado em 06 de outubro de 2005.

ELMASRI, Ramez, Navathe B. Shamkant. *Fundamentals of Database Systems* 3 ed. Addison Wesley, 2000.

GALS, Gerador de Analisadores Léxicos e Sintáticos – www.sourceforge.net acessado em 15 de fevereiro de 2006.

HEUSER, Carlos Alberto. *Projeto de banco de dados* 4 ed., Sagra Luzzatto, 2001.

MACHADO, Felipe Nery Rodrigues; Abreu, Maurício Pereira de. *Projeto de banco de dados: uma visão prática* 5 ed. rev. , Érica 1999.

MELLO, Ronaldo dos Santos, *Apostilas aulas de Projeto de Banco de Dados*, ministradas na Universidade Federal de Santa Catarina, 2003.

SANTOS, Antonio Raimundo dos. *Metodologia Científica a construção do conhecimento*

SETZER, Valdemar W. *Bancos de dados: Conceitos, modelos, gerenciadores, projeto lógico e projeto físico* 3 ed. ver., Edgard Blücher 1989.

ANEXOS - Artigo da Monografia.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CTC – Centro Tecnológico
Florianópolis, fevereiro de 2006.

INE – Departamento de Informática e Estatística
Curso de Sistemas de Informação
Trabalho de Conclusão de Curso

Alunos: Marcelo Caon de Souza
Miguel Kojiio Nobre

“Ferramenta de apoio a Engenharia Reversa de um Banco de Dados Relacional”

RESUMO:

Grande parte da documentação dos Bancos de dados não existe. Este artigo apresenta de forma sintética os resultados obtidos no desenvolvimento de uma ferramenta para auxílio ao processo de engenharia reversa de um banco de dados relacional em um modelo entidade – relacionamento.

INTRODUÇÃO:

Os sistemas computacionais, permitiram que grandes quantidades de dados fossem manipulados a fim de que informação fosse produzida. Este dado, agora com valor agregado, deveria portanto ser persistido para utilização futura. Foi neste contexto que surgiram os Bancos de dados.

Apesar de praticamente todo sistema de médio e grande porte ter um Banco de dados, é muito fácil encontrá-los sem nenhum tipo de documentação.

É neste contexto que se insere a ferramenta que será apresentada, existem grandes massas de dados sem uma documentação que possa ajudar o entendimento do universo que a mesma abstrai.

DESENVOLVIMENTO:

O processo de engenharia reversa tem por objetivo, no caso estudado, partir de um esquema físico para um esquema conceitual. Neste caso o esquema físico será o script SQL que foi utilizado para implementar o Banco de Dados.

Por tratar-se de uma abstração de maior nível o esquema conceitual possui muita informação que não pode ser mais obtida no esquema físico.

Uma boa metodologia de engenharia reversa deve ser capaz de diminuir ao máximo a quantidade de decisões subjetivas que ficam a cargo do usuário da ferramenta.

Dentro desta abordagem a metodologia mais adequada dentre as pesquisadas foi a desenvolvida por Heuser, uma vez que ela permite que a maior parte do processo seja feito de forma automática pela ferramenta, reduzindo o número de decisões tomadas pelo usuário.

O desenvolvimento da ferramenta foi baseado em três grandes processos até o produto final, neste caso o esquema E-R:

Pré-análise: Ao iniciar o processo de engenharia reversa o script SQL é pré analisado a fim de que sejam encontrados os relacionamentos, chaves primárias e atributos. Tudo o que puder ser feito de forma automática, será reconhecido nesta fase. Observar figura 0-1.

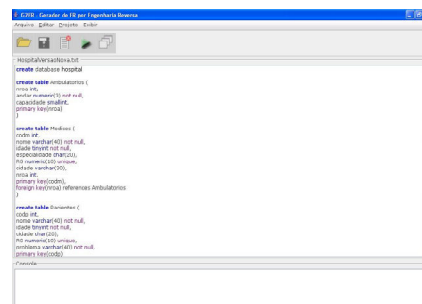


Figura 0-1 : Tela de Inserção e Leitura do Script SQL

Interação: Após a etapa de pré-análise do script, existem questões que não podem ser solucionadas de forma automática, nestes casos o usuário irá participar do processo de engenharia reversa ativamente, definindo qual será o caminho a ser tomado pela ferramenta. Observar figura 0-2.

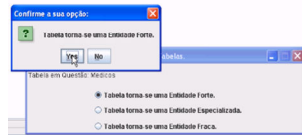


Figura Erro! Nenhum texto com o estilo especificado foi encontrado no documento.-1 : **Exemplo de Interação com o Usuário**

Visualização: Após as duas etapas de processamento do script o resultado final é apresentado na forma de um esquema E-R com a representação das entidades, atributos, relacionamentos e etc. Observar figura 0-3.

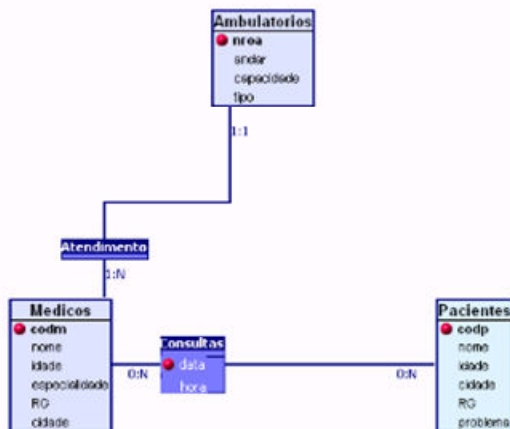


Figura Erro! Nenhum texto com o estilo especificado foi encontrado no documento.-2 : **Apresentação do Esquema ER gerado**

CONCLUSÃO:

O processo de engenharia reversa de um banco de dados é muito trabalhoso e toda ferramenta case que atue neste nicho vai encontrar um vasto campo a explorar.

A implementação desta solução não tem por objetivo esgotar o processo, existem ainda lacunas que podem vir a ser

preenchidas em trabalhos futuros como por exemplo a persistência do resultado obtido em arquivo XML, extensão do parser de interpretação do script SQL entre outras.