

**UNIVERSIDADE FEDERAL DO PARANA**

**CRISTIANE VIEIRA PROENÇA MIAMOTO**

**REFINAMENTO DE UM DIAGRAMA ENTIDADE-RELACIONAMENTO:  
ESTUDO DE CASO EM UM SISTEMA ERP**

**CURITIBA**

**2012**

**CRISTIANE VIEIRA PROENÇA MIAMOTO**

**REFINAMENTO DE UM DIAGRAMA ENTIDADE-RELACIONAMENTO:  
ESTUDO DE CASO EM UM SISTEMA ERP**

Monografia apresentada como requisito parcial para conclusão do Curso de Especialização em Informática com ênfase em Análise Orientada à Objetos da Universidade Federal do Paraná – UFPR.

Orientadora: Prof<sup>a</sup>. Carmem Satie Hara

**CURITIBA**

**2012**

A DEUS, meu esposo Hilosi, meus pais  
Josias e Mercedes e meus irmãos  
Joslaine, J Manoel, Junior e Tyago.

"Feliz o homem que encontrou a sabedoria, o homem  
que alcançou o entendimento."

(Provérbios 3:13)

## RESUMO

Com o crescimento do mercado de software nota-se a crescente utilização de técnicas de desenvolvimento informais, que acarretam problemas na manutenção e uso desses programas. A engenharia reversa ajuda na manutenção ou geração da documentação de um sistema para facilitar o seu entendimento em atividades de manutenção. Dentro deste contexto, foi realizado um estudo que resolvesse parcialmente o problema de documentação de um banco de dados em um sistema ERP industrial legado que possui 22 anos de vida. Ao contrário de projetos novos, nos quais a etapa de desenvolvimento de um diagrama entidade-relacionamento (DER) é executada antes da criação do modelo relacional, nesta monografia é apresentado o processo inverso, no qual a partir de um esquema relacional é extraído um diagrama entidade-relacionamento com as cardinalidades dos relacionamentos. Para realizar esta tarefa, foram estudados e comparados dois artigos: *Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering* (Alhajj, 2003) e *Extracting the extended entity-relationship model from a legacy relational database* (Andersson, 1994).

Palavras-chave: Entidade-relacionamento. Engenharia Reversa. Cardinalidade.

## **ABSTRACT**

With the growth of the software market we can notice an increasing use of informal development techniques, which cause problems in the maintenance and use of these programs. Reverse engineering helps in the maintenance or generation of documentation to facilitate their understanding in updating activities. Within this context, we report an experience for solving part of the problem of documenting a legacy database of an ERP system that has 22 years of life. Unlike new projects, in which the design of an entity-relationship diagram is executed before the creation of the relational model, in this monograph we present the inverse process, in which a model is extracted from a relational schema. To accomplish this task two papers were studied and compared: Extracting Entity Relationship Schema from a Relational Database through Reverse Engineering (Alhaji, 2003) and Extracting the extended entity-relationship model from a legacy relational database (Andersson, 1994).

Keywords: Entity-relationship. Reverse Engineering. Cardinality.

## LISTA DE ABREVIATURAS E SIGLAS

- DBMS - *Database Management System*
- DER - Diagrama Entidade-Relacionamento
- ERC - *Enterprise Resource Center*
- ERP - *Enterprise Resource Planning*
- ER - Entidade Relacionamento
- SGBD - Sistema Gerencial de Banco de Dados

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>9</b>
<b>2 PROPOSTAS DE ENGENHARIA REVERSA APLICADA A BANCO DE DADOS</b> .....	<b>11</b>
2.1 EXTRAÇÃO DE UM ESQUEMA DE RELACIONAMENTO ENTRE ENTIDADES DE UM BANCO DE DADOS RELACIONAL ATRAVÉS DA ENGENHARIA REVERSA .....	11
2.2 EXTRAINDO A EXTENSÃO DO RELACIONAMENTO DA ENTIDADE-MODELO A PARTIR DE UM BANCO DE DADOS RELACIONAL LEGADO.....	12
2.2.1 Passos da abordagem proposta .....	13
2.3 COMPARAÇÃO.....	17
<b>3 ESTUDO DE CASO</b> .....	<b>18</b>
3.1 ENGENHARIA REVERSA DE DADOS .....	18
3.2 ANALISE DE RESULTADOS.....	19
<b>4 CONCLUSÃO</b> .....	<b>21</b>
<b>REFERÊNCIAS</b> .....	<b>22</b>
<b>APÊNDICE - COMANDOS SQL PARA EXTRAIR AS CARDINALIDADES</b> .....	<b>23</b>

## 1 INTRODUÇÃO

Uma das grandes dificuldades encontradas por diversas organizações é a falta de documentação das aplicações, principalmente dos sistemas desenvolvidos utilizando um banco de dados. Isto dificulta a manutenção do sistema, que envolve a alteração de *scripts* e procedimentos de manipulação dos dados, para adaptá-los às necessidades dos clientes ou correção de erros detectados nas aplicações.

Quando não há uma análise e projeto de software no início do processo de desenvolvimento, os prejuízos são percebidos posteriormente, devido a falta de padrões, manuais e não-uniformidade na implementação de rotinas e módulos que o compõe. Estas propriedades conduzem a um sistema inflexível e de alto custo de manutenção.

Nesse contexto, a engenharia reversa faz-se necessária para recuperar as informações erroneamente registradas ou perdidas durante o desenvolvimento do sistema, buscando-se documentar o real estado do banco de dados.

Isso possivelmente resultará em uma visão organizada e formal. A Engenharia Reversa de Software é uma técnica utilizada para recuperar informações a partir dos documentos do software relativos ao produto ou código-fonte, visando a obtenção de sua representação em nível mais alto de abstração (Alhajj, 2003). Ela se destina a criar “visões” do sistema em diferentes níveis de abstração, facilitando seu entendimento, com o principal objetivo de ajudar na manutenção do sistema.

Com o crescimento do mercado de software, há também uma crescente utilização de técnicas de desenvolvimento informais que acarretam em problemas na manutenção e uso desses programas.

Além disso, modificações e adições de novas características produzem efeitos colaterais inesperados. A documentação existente torna-se então incompleta, não refletindo o software existente e dificultando o gerenciamento de seu uso e manutenção.

A redocumentação visa criar novas visões do sistema através da análise do banco de dados, com o objetivo de melhorar sua compreensão. A criação dessas visões adicionais, geralmente gráficas, tem como objetivo recriar a documentação que já existiu ou que deveria ter existido sobre o sistema.

O projeto visa recuperar todas as informações necessárias para se compreender melhor o que o sistema faz, como ele o faz e porque ele o faz.

De acordo com (Andersson, 1994):

Sistemas legados são caracterizados por uma arquitetura ultrapassada, por falta de documentação e também pela não-uniformidade do desenvolvimento, resultado de numerosas extensões. A engenharia reversa pode ajudar através da extração conceitual, especificação de uma implementação independente que fornecerá uma base para uma evolução futura de um Sistema de Informação.

A Engenharia reversa também auxilia no reajuste e implementação de componentes de um sistema para melhorar a sua funcionalidade e desempenho.

Conforme (Alhajj, 2003), a Engenharia Reversa se faz necessária nos sistemas tradicionais para manter e ajustar o esquema entidade-relacionamento de um banco de dados legado existente.

O restante deste trabalho é organizado como segue. A seção 2 apresenta o resumo de dois artigos sobre a Engenharia Reversa, e descreve a metodologia do artigo selecionado para o desenvolvimento deste trabalho. A Seção 3 apresenta a descrição de um sistema real de ERP que é utilizado como estudo de caso, o projeto do experimento e análise dos resultados. Finalmente, a Seção 4 fornece uma visão global do que foi apresentado no trabalho, conclusões e direcionamento para trabalhos futuros.

## 2 PROPOSTAS DE ENGENHARIA REVERSA APLICADA A BANCO DE DADOS.

Neste capítulo são apresentadas duas abordagens de engenharia reversa para sistemas gerenciadores de banco de dados. A abordagem escolhida para o desenvolvimento deste trabalho é o descrito por (Alhajj, 2003), cujas etapas são apresentadas em detalhe na Seção 2.2.

### 2.1 EXTRAÇÃO DE UM ESQUEMA DE RELACIONAMENTO ENTRE ENTIDADES DE UM BANCO DE DADOS RELACIONAL ATRAVÉS DA ENGENHARIA REVERSA

Nesta seção é apresentada a abordagem de extração de dependências funcionais a partir diferentes fontes para a elaboração de um diagrama entidade-relacionamento que foi proposta em (Andersson, 1994).

Esta abordagem não supõe que as chaves primárias e estrangeiras das relações são conhecidas. Assim estas informações são extraídas de outras fontes, tais como: do código fonte das aplicações, das informações, dos nomes de tabelas, dos campos das tabelas e índices. Este artigo propõe a extração dessas informações a partir da declaração de manipulação de dados incorporados nos formulários e relatórios. Tal estudo enfatiza o uso da declaração de manipulação de dados em consultas e visões.

A análise das informações é feita para extrair os dados. Após isso o esquema é transformado num diagrama entidade-relacionamento e um conjunto de regras, divididas em três grupos:

- Regras Nó: traduz em amarração os relacionamentos entre uma tabela e outra;
- Regras Link: tiram as conclusões de cardinalidades mínimas e máximas presentes no diagrama;
- Refinamento para gerar o diagrama final e otimizar o gráfico.

## 2.2 EXTRAINDO A EXTENSÃO DO RELACIONAMENTO DA ENTIDADE- MODELO A PARTIR DE UM BANCO DE DADOS RELACIONAL LEGADO

Para transformar um esquema relacional em um diagrama entidade-relacionamento (DER) refinado, foi escolhida a metodologia desenvolvida por (Alhajj, 2003). Nesta seção é apresentada uma breve abordagem do artigo, composto por três etapas:

- a) procura de chaves primárias;
- b) procura de chaves estrangeiras;
- c) procura de cardinalidades.

A procura por cardinalidades será explicada em detalhes nas próximas seções.

As informações sobre chaves candidatas e chaves estrangeiras são necessárias e suficientes para desenhar um gráfico consistente com um DER (Alhajj, 2003).

Por conseguinte, a busca por estas chaves é um passo fundamental para a geração do DER. A maioria das pesquisas feitas neste campo é baseada na utilização do esquema relacional como insumo básico, visando e extraindo as informações semânticas de uma análise exaustiva das relações individuais no esquema, juntamente com seus atributos essenciais e não essenciais. É realmente um processo informal, que exige muito do envolvimento do usuário. As duas principais etapas envolvidas nesta abordagem são as identificações de tabelas e também identificar os diferentes tipos de relacionamentos entre elas. As chaves primárias são identificadas procurando por índices únicos, e as chaves estrangeiras são utilizadas para relacionamentos de integridade referencial. O último passo no processo envolve (quando possível), melhorar o projeto original, identificando relacionamentos com atributos, de cardinalidades muitos-para-muitos e relacionamentos n-ários. Estas etapas de refinamento são o foco deste trabalho.

A abordagem proposta por (Alhajj, 2003) tem como entrada:

- Um conjunto de tabelas, atributos e a base de dados (real), sem informação de nenhuma restrição de integridade tais como: chave e chave estrangeira.

E produz como saída:

- Diagrama ER refinado com entidades, atributos e relacionamento com cardinalidade que pode ser um-para-um (1:1), um-para-muitos (1:n) e muitos-para-muitos (m:n).

### 2.2.2 Passos da abordagem proposta

Os primeiros passos de proposta consistem da extração de informações de chave primária e chave estrangeira.

- Procura de Chave Primária: O algoritmo recebe como entrada a combinação de todos os atributos da tabela e fornece como saída uma tabela com os campos que tem maior probabilidade de ser chave primária. Para se chegar à conclusão que tais atributos podem compor uma chave primária é executada uma consulta para determinar a quantidade de valores distintos de um conjunto de atributos;
- Procura de Chave Estrangeira: O algoritmo recebe como entrada as chaves candidatas fornecidas pelo passo anterior e fornece como saída os campos que tem maior probabilidade de ser chave estrangeira. As entradas do algoritmo consistem de uma determinada relação e as chaves candidatas fornecidas pelo algoritmo de procura de chave primária. O algoritmo considera que as chaves estão em ordem ascendente de número de atributos. A partir do resultado destes dois passos, é possível gerar um diagrama entidade-relacionamento simplificado, no qual cada tabela é representada por uma entidade e cada chave estrangeira corresponde a um relacionamento 1:n.

A figura 1 apresenta um exemplo de diagrama entidade-relacionamento simplificado, no qual relacionamentos são representados com ligações entre as entidades (que contém atributos) e a cardinalidade é expressa com o símbolo de chave no “lado 1” e o símbolo de infinito no “lado n”. Este diagrama é dito “simplificado” porque a cardinalidade mínima não é representada e todos os relacionamentos são considerados do tipo 1:n.

Para refinar a informação sobre cardinalidade do relacionamento, Alhajj (2003) propõe um passo adicional, que é descrito na seqüência. A entrada para o algoritmo consiste de uma tabela com informações de Chave Estrangeira e a saída é

composta pela cardinalidade mínima e máxima dos relacionamentos representados no diagrama simplificado. O algoritmo é apresentado na figura 2.

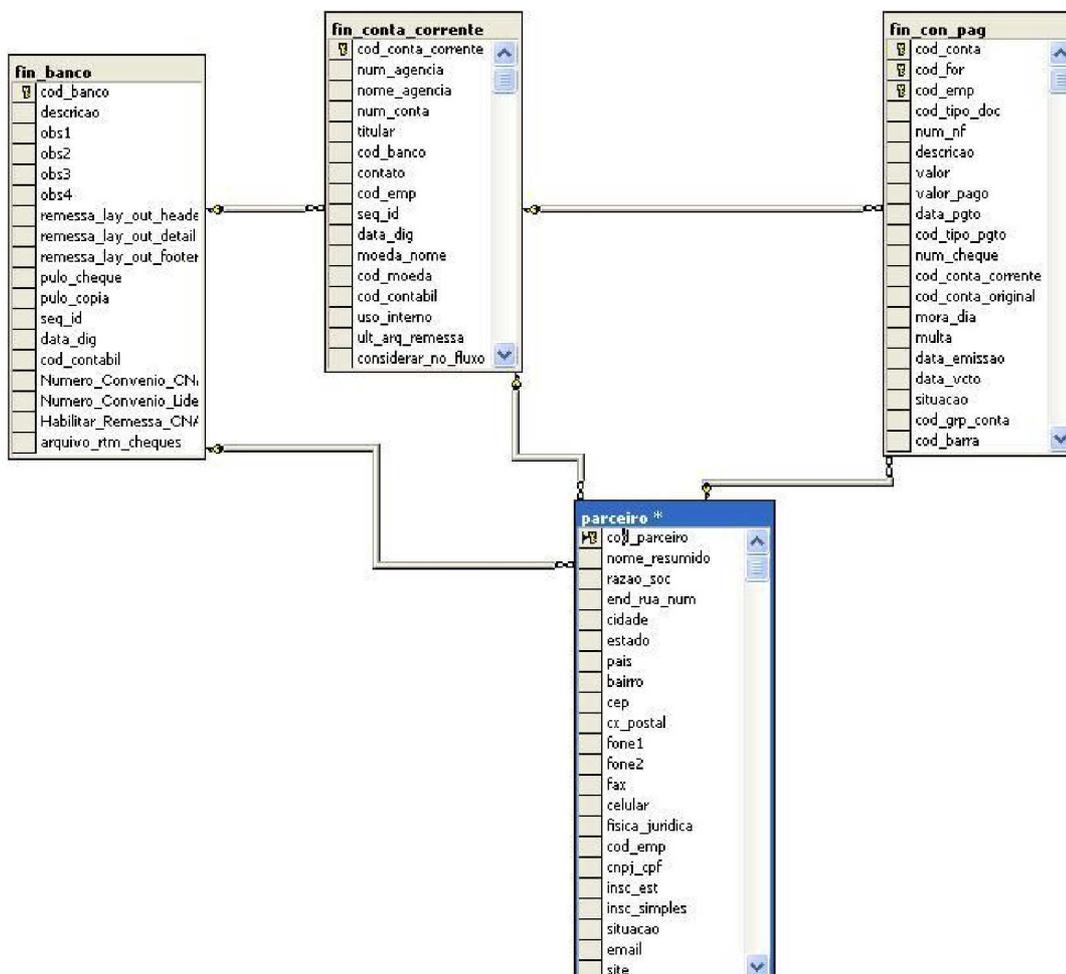


FIGURA 1 - DER SIMPLIFICADO

Para se obter as cardinalidades o algoritmo recebe como entrada a tabela Chave Estrangeira e fornece como saída a cardinalidade existente entre os relacionamentos. O algoritmo é apresentado na figura 2.

Primeiramente a variável R1 recebe o nome da relação que contém a chave primária, a variável R2 recebe o nome da relação que contém a chave estrangeira (FK). A variável R recebe a projeção de R2 sobre os atributos FK e a variável P recebe o mesmo conjunto sem valores duplicados (linhas 2 a 4). O restante do algoritmo é dividido em duas partes principais:

- a) Obtenção da cardinalidade mínima e máxima do lado de R1 (linhas 5 à 9): a cardinalidade do lado das relações que são chaves primárias é obtida utilizando a seguinte verificação: “Se tamanho(P)=tamanho(R1)”, (linha 5), ou seja, se a quantidade de registros em P for igual a quantidade de registros em R1, conclui-se que todos os registros de R1 estão relacionados com elementos em R2 e portanto a participação é obrigatória. Ou seja, a cardinalidade mínima e máxima é de 1 do lado da chave primária (linha 6). Se a quantidade de registros em P for diferente da quantidade de registros em R1, ocorre uma cardinalidade 0 ou 1, ou seja a participação é opcional do lado R1 (linha 8);
- b) Obtenção da cardinalidade mínima e máxima do lado de R2 (linhas 10 à 23): a cardinalidade do lado das relações que são chaves estrangeiras é obtida utilizando a seguinte verificação: “Se tamanho(P)<tamanho (R) ”, (linha 10), ou seja, se a quantidade de valores distintos de chaves estrangeiras for menor que a quantidade de registros em R2 com valores de FK diferente de nulo indica a cardinalidade máxima de “muitos”; caso contrário a cardinalidade máxima é 1. “Se tamanho (R)=tamanho (R2)” (linha 11), ou seja, se não há registros em R2 com valores de FK iguais a nulo, conclui-se que a cardinalidade mínima é de 1 ou seja o relacionamento é obrigatório para R2 (linha 12); caso contrário, a cardinalidade mínima é 0 ou seja o relacionamento é opcional para o lado R2 (linha 14).

QUADRO 1 - ALGORITMO PROCURA DE CARDINALIDADES

Entrada: A relação da tabela com as chaves estrangeiras
Saída: A cardinalidade entre os relacionamentos
Etapas:
1. Para cada aresta (R2, R1) no gráfico faça
2.       Seja FK a chave estrangeira em R2 que referencia a chave primária de R1
3.       Let R= select fk from R2 where the value of fk is not null
4.       Let P = select distinct fk from R
<b>/*Decide a mínima e máxima cardinalidade do lado R1</b>
5.       Se tamanho(P)=tamanho(R1) então
6.             Ligação (R2,R1) é obrigatória do lado R1, com cardinalidade máxima igual a 1
7.       Senão
8.             Ligação (R2,R1) é opcional do lado R1, com cardinalidade máxima igual a 1
9.       FimSe
<b>/*Decide a mínima e máxima cardinalidade do lado R2*/</b>
10.       Se tamanho(P)<tamanho (R) então
11.             Se tamanho (R)=tamanho (R2) então
12.                 Ligação (R2,R1) é obrigatória do lado R2, com cardinalidade máxima igual a
M
13.       Senão
14.             Ligação (R2,R1) é opcional do lado R2, com cardinalidade máxima igual a
M
15.       FimSe
16.       Senão Se tamanho (P)=tamanho (R) então
17.             Se tamanho (R2) = tamanho(R)
18.                 então
19.                     Ligação (R2,R1) é obrigatória do lado R2, com cardinalidade máxima igual a
1
20.       Senão
21.             Ligação (R2,R1) é opcional do lado R2, com a cardinalidade máxima igual a
1
22.       FimSe
23.       FimSe

FONTE: (Alhaji, 2003).

## 2.3 COMPARAÇÃO

O artigo extração de um esquema de relacionamento entre entidades de um banco de dados relacional através da engenharia reversa de (Andersson, 1994), apresenta um método para extração de um esquema conceitual de uma base de dados relacional. Este método é baseado sobre uma análise de declaração de manipulação de dados no código fonte de uma aplicação. Os atributos representam referências entre tabelas no esquema relacional e possíveis chaves são determinadas por uma análise de condições associadas nas consultas e visões. Conhecimento sobre quais atributos que ligam as tabelas é usado para investigar a extensão da base de dados de uma forma seletiva. Quando as chaves não podem ser precisamente determinadas, possíveis soluções são geradas pelo sistema sob a orientação do usuário.

De acordo com (Alhaji, 2003), o grau de sucesso na extração de um diagrama entidade-modelo a partir de um banco de dados relacional legado, depende do esforço colocado no processo de extração e do conhecimento do software. Durante o processo de extração, as características do banco de dados legado são investigadas para obter e verificar os principais e possíveis candidato(s) de cada relação, a fim de decidir sobre a sua chave primária. A outra decisão é a comparação entre os domínios e os valores do atributo(s) em uma chave com candidatos em todos os outros atributos e relações existentes, para se descobrir qual chave candidata de uma relação de chave estrangeira tem dado representante(s) em outras relações. As informações sobre o candidato e as chaves estrangeiras são necessárias e suficientes para desenhar um gráfico consistente com o diagrama ER. O último passo envolve determinar sobre os relacionamentos com atributos, de cardinalidade n:m e relacionamentos n-ários.

### 3 ESTUDO DE CASO

A manutenção de um banco de dados depende da compreensão de suas características e é um trabalho complexo. A situação torna-se pior quando a documentação relacionada é inexistente ou desatualizada. Em geral, estas propriedades conduzem a sistemas inflexíveis e de alto custo de manutenção.

Quando há necessidade de desenvolver um novo módulo na aplicação e apresentar a estrutura a um novo cliente ou funcionário, surge uma grande dificuldade em expor como as tabelas estão relacionadas e a utilidade de cada uma no sistema. A aplicação na qual o caso de uso reportado neste capítulo foi implementado é um sistema ERP (*Enterprise Resource Planning*), que é um sistema de gestão industrial constituído dos seguintes módulos: materiais, industrial, compras, comercial, financeiro, custo, gerencial, correio, exportador e sistema. A base de dados é gerenciada pelo SGBD SQL Server.

Para a execução deste estudo foi feita uma cópia da estrutura das tabelas e também dos dados.

O sistema ERP consiste de 251 tabelas, cada uma contendo de 4 à 70 atributos. Para este estudo de caso foram consideradas 22 tabelas e aproximadamente os 30 primeiros atributos de cada uma.

As tabelas e suas respectivas descrições e atributos são apresentados no Apêndice A.

#### 3.1 ENGENHARIA REVERSA DE DADOS

Nesta seção é apresentada a Engenharia Reversa de Dados do sistema seguindo a abordagem proposta por (Alhadj, 2003). Foi considerado o módulo de ERP Industrial.

A Figura 3 ilustra o resultado obtido com a extração das cardinalidades dos relacionamentos entre as tabelas do módulo comercial, especificamente o pedido. As tabelas são: *fin\_banco*, *fin\_conta\_corrente*, *parceiro* e *fin\_con\_pag*, as cardinalidades são representadas entre os relacionamentos. O campo *cod\_banco* é a chave primária da tabela *fin\_banco*. O campo *cod\_conta\_corrente* é a chave primária da tabela *fin\_conta\_corrente*. O campo *cod\_parceiro* é a chave primária da

tabela parceiro. O campo `cod_conta` é a chave primária do campo `fin_con_pag`. O lado chave estrangeira dos relacionamentos é indicado pelo símbolo de infinito.

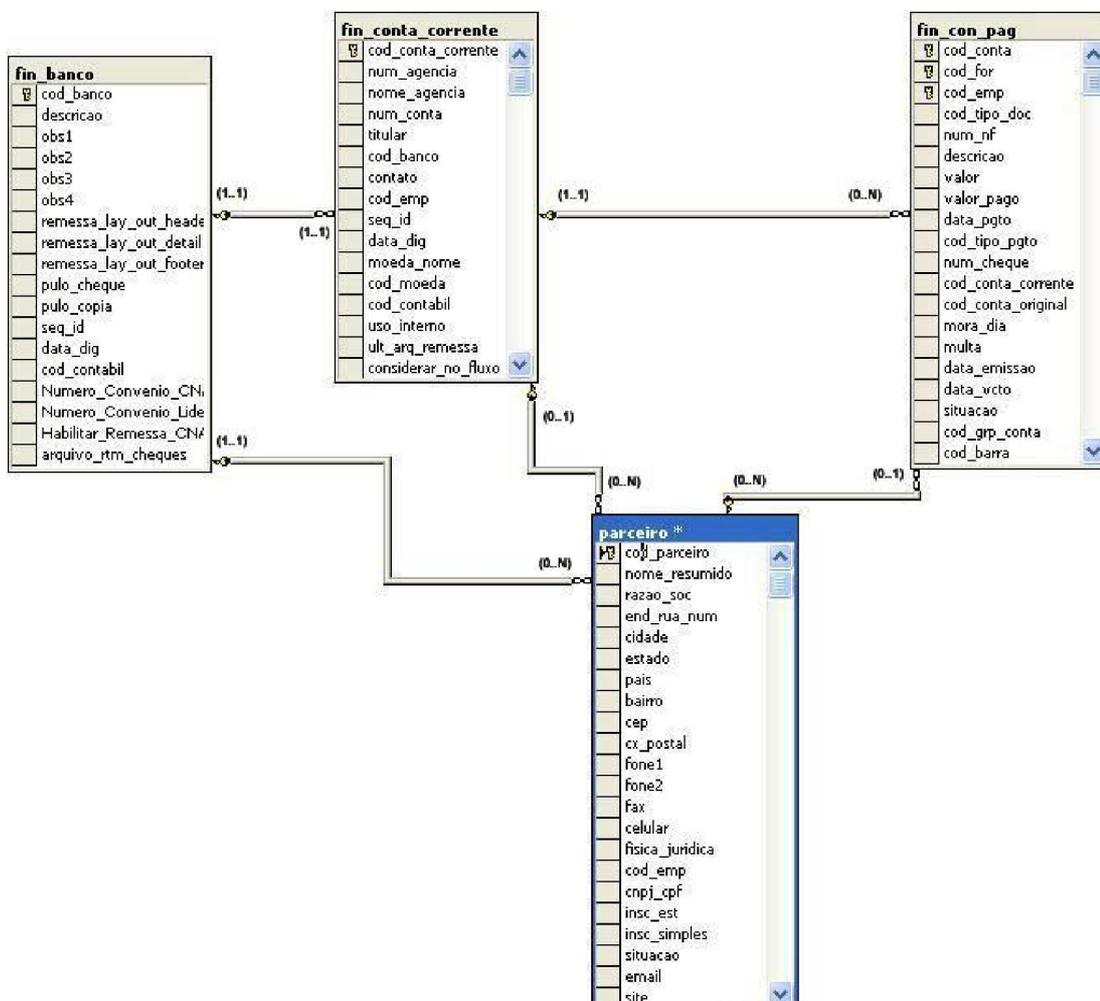


FIGURA 2 - EXTRAÇÃO DAS CARDINALIDADES DOS RELACIONAMENTOS DA ENTIDADE-MODELO

### 3.2 ANÁLISE DE RESULTADOS

A implementação teve sucesso quanto a geração das cardinalidades. O resultado foi a definição da participação (opcional ou obrigatória) de cada entidade envolvida no relacionamento 0 ou 1 e sua cardinalidade máxima, que pode ser 1 ou N. Os resultados são consistentes com o conhecimento prévio de aplicação. Isso

mostra que a abordagem proposta por (Alhajj, 2003) foi eficaz no refinamento das cardinalidades no estudo de caso.

## 4 CONCLUSÃO

Este trabalho teve como motivação tirar da informalidade as informações do sistema ERP legado utilizado no estudo de caso. Os resultados obtidos foram satisfatórios para novas implementações e manutenções nas aplicações, além de proporcionar clareza e objetividade para desenvolvedores, usuários e futuros clientes.

Neste trabalho foi estudado um algoritmo que procura por cardinalidades em um diagrama entidade-relacionamento a partir da base de dados do sistema. Foi realizado um estudo de caso utilizando quatro tabelas, com suas respectivas chaves primárias e chaves estrangeiras.

No decorrer do desenvolvimento deste trabalho, observaram-se, dentre outras, as seguintes oportunidades para trabalhos futuros:

- dar continuidade na extração de cardinalidades dos outros relacionamentos simplificados;
- gerar um arquivo dhtml ou xml com os relacionamentos existentes, para transpor o sistema na web.

## REFERÊNCIAS

ALHAJJ, R. **Extracting the extended entity-relationship model from a legacy relational database**. Information Systems, Oxford, Vol. 28, Num. 6, p. 597-618, Set. 2003.

ANDERSSON, M. **Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering**. Proceedings of the International Conference on Entity-Relationship Approach, Manchester, p.403-419, Dez. 1994.

HENRY, F. K.; ABRAHAM, S.S.S. **Sistema de Banco de Dados**. São Paulo: Makron Books, 1995.

MELLO, R. S. **Projeto de Banco de Dados**. Disponível em: <<http://www.inf.ufsc.br/~ronaldo/ine5613/12-ER.pdf>>. Acessado em: 01 Set. 2010.

PETIT, F. T.; BOULICAUT J.F.; KOULOUMDJIAN, J. **Towards the reverse engineering of denormalized relational databases**, Proceedings of the IEEE. International Conference on Data Engineering, New Orleans, p.350-364, 1996.

REZENDE, D.A. **Engenharia de Software e Sistemas de Informação**. Rio de Janeiro: Brasport, 2005.

TARJINO, R. **Modelo Entidade Relacionamento**. Disponível em: <[http://www.cos.ufrj.br/~targino/bd1/unidade2\\_6sl.pdf](http://www.cos.ufrj.br/~targino/bd1/unidade2_6sl.pdf)> Acesso em: 06 Jul. 2010.

UNIVERSIDADE FEDERAL DE SANTA CATARINA. **Extração de um ER de banco de dados**. Disponível em: <[http://projetos.inf.ufsc.br/arquivos\\_projetos/projeto\\_667/Extra%E7%E3o%20de%20ER.doc](http://projetos.inf.ufsc.br/arquivos_projetos/projeto_667/Extra%E7%E3o%20de%20ER.doc)> . Acesso em: 05 Set. 2010.

**APÊNDICE - COMANDOS SQL PARA EXTRAIR AS CARDINALIDADES**

--fin\_banco / fin\_conta\_corrente

select cod\_banco from fin\_conta\_corrente where cod\_banco is not null --

R select distinct cod\_banco from fin\_conta\_corrente --P

select count(cod\_banco) from fin\_banco --R1

select count(cod\_conta\_corrente) from fin\_conta\_corrente --R2

--parceiro / fin\_banco

select cod\_banco from parceiro where cod\_banco is not null --

R select distinct cod\_banco from parceiro --P

select count(cod\_banco) from fin\_banco --R1

select count(cod\_parceiro) from parceiro -- R2

--parceiro/fin\_cond\_pag

select cod\_cond\_pag from parceiro where cod\_cond\_pag is not null --

R select distinct cod\_cond\_pag from parceiro --P

select count(cod\_cond\_pag) from fin\_cond\_pag --

R1 select count(cod\_parceiro) from parceiro -- R2

--parceiro/fin\_conta\_corrente

select cod\_conta\_corrente from parceiro where cod\_conta\_corrente is not null --

R select distinct cod\_conta\_corrente from parceiro --P

select count(cod\_conta\_corrente) from fin\_conta\_corrente --

R1 select count(cod\_parceiro) from parceiro -- R2

--fin\_con\_pag/fin\_conta\_corrente

select cod\_conta\_corrente from fin\_con\_pag where cod\_conta\_corrente is not null -

- R

select distinct cod\_conta\_corrente from fin\_con\_pag --P

select count(cod\_conta\_corrente) from fin\_conta\_corrente --

R1 select count(cod\_conta) from fin\_con\_pag -- R2

[PDF to Word](#)