

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GIOVANI LUCENA RIZZIERI

**Reengenharia de Base e Estudo de Dados
Estatísticos de Clínica Médica – Clínica do
Sono**

Trabalho de Graduação.

Prof. Dr. Leandro Krug Wives
Orientador

Prof. Dr. Dênis Martinez
Co-orientador

Porto Alegre, dezembro de 2009.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do CIC: Prof. João César Netto

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço mais esta vitória em minha vida às seguintes pessoas:

À meus pais e minha irmã, pela sapiência, tranquilidade, companheirismo e amor incondicional.

À minha baixinha, por tudo. Simplesmente não tenho palavras para descrever o sentimento de ter ao meu lado.

À meu orientador, professor Leandro Wives, pela paciência, atenção e conselhos no decorrer desta etapa muito importante.

Finalmente, agradeço ao meu avô, meu melhor amigo, pela amizade mais especial que já tive, bem como o companheirismo e os ensinamentos passados ao longo de minha vida.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS.....	6
LISTA DE FIGURAS	7
LISTA DE TABELAS	8
RESUMO	9
ABSTRACT.....	10
1 INTRODUÇÃO	11
2 ANÁLISE DA BASE DE DADOS EXISTENTE.....	13
2.1 Descrição das tabelas que compõem a base atual	14
2.2 Etapa 1: Reconhecimento de entidades irrelevantes ou que devem integrar outras entidades 16	
2.2.1 Tabela exercício.....	16
2.2.2 Tabela E-MAIL.....	17
2.2.3 Tabela SCL90	17
2.2.4 Tabela atenas+sono1	17
2.2.5 Tabela alfa-delta sono leve e dor	18
2.3 Etapa 2: Verificação de entidades, relacionamentos ou atributos redundantes	19
2.4 Etapa 3: Atributos com nomes diferentes, porém mesma finalidade	20
2.5 Etapa 4: Atributos que devem se tornar registros de tabelas.....	22
2.6 Etapa 5: Entidades que não apresentam o devido relacionamento.....	23
2.7 Etapa 6: Repetição de chave em tabelas.....	26
2.8 Etapa 7: Adequação da base para atender a aplicação web	28
3 DETALHAMENTO DA SOLUÇÃO.....	29
3.1 Descrição e significado das tabelas presentes no novo modelo de dados	34

3.1.1	Tabela CS_BI_PARAMETROS	34
3.1.2	Tabela CS_BI_PARAMETROS_ALTERNATIVAS	34
3.1.3	Tabela CS_BI_MEDICOS_INDICACOES	35
3.1.4	Tabela CS_BI_MEDICAMENTOS	35
3.1.5	Tabela CS_BI_IDENTIFICACAO	35
3.1.6	Tabela CS_BI_USUARIO	36
3.1.7	Tabela CS_BI_QUESTOES	36
3.1.8	Tabela CS_BI_QUESTIONARIOS	37
3.1.9	Tabela CS_BI_EXAME_LABO	37
3.1.10	Tabela CS_BI_CONVENIOS	37
3.1.11	Tabela CS_BI_DADOS_CLIN	37
3.1.12	Tabela CS_BI_CONSULTAS_PAC	38
3.1.13	Tabela CS_BI_USUARIO_PERMISSAO	38
3.1.14	Tabela CS_BI_ALTERNATIVAS	38
3.1.15	Tabela CS_BI_RESPOSTAS_PAC	38
3.1.16	Tabela CS_BI_DIAGNOSTICOS	38
3.1.17	Tabela CS_BI_FINANCEIRO	39
3.1.18	Tabela CS_BI_EXAME_LABO_PAC	39
3.1.19	Tabela CS_BI_DADOS_CLIN_PAC	39
3.1.20	Tabela CS_BI_DADOS_CLINICOS_DIAG	39
3.1.21	Tabela CS_BI_MEDICAMENTOS_PAC	40
3.1.22	Tabela CS_BI_QUESTIONARIO_QUESTOES	40
3.2	Análise dos dados e cargas na nova base	41
3.2.1	Seleção na base Access das informações necessárias	41
3.2.2	Elaboração dos scripts de carga	42
3.2.3	Inserção na base MySQL	43
4	LEVANTAMENTOS DE DADOS ESTATÍSTICOS	45
5	CONCLUSÕES.....	50
	REFERÊNCIAS.....	51
	GLOSSÁRIO.....	52
	CONCEITOS RELACIONADOS.....	52
	ANEXO A - SCRIPTS DE CRIAÇÃO DAS TABELAS NOVAS.....	54

LISTA DE ABREVIATURAS E SIGLAS

BI	<i>Business Intelligence</i>
MS-Access	Microsoft Access

LISTA DE FIGURAS

FIGURA 2.1- APRESENTAÇÃO SUPERFICIAL DA BASE DE DADOS EM ACCESS.....	13
FIGURA 2.2 - TABELA EXERCICIO.....	16
FIGURA 2.3 - TABELA E-MAIL.	17
FIGURA 2.4 - TABELAS SCL 90 E 16 MIL NUMEROS EM SEQUENCIA.	177
FIGURA 2.5 - TABELA ATENAS+SONO1.	18
FIGURA 2.6 - TABELAS COM ATRIBUTOS INCORRETOS	179
FIGURA 2.7 - PROBLEMA DE REDUNDÂNCIA DE ENTIDADES, RELACIONAMENTOS OU ATRIBUTOS.	20
FIGURA 2.8 - PROBLEMA DO MESMO TIPO DE INFORMAÇÃO EM ATRIBUTOS COM NOMES DIFERENTES	21
FIGURA 2.9 - ATRIBUTOS QUE DEVEM SE TORNAR REGISTROS EM TABELAS.....	23
FIGURA 2.10 - AUSÊNCIA DE RELACIONAMENTO ENTRE DUAS TABELAS.....	24
FIGURA 2.11 - AUSÊNCIA DE RELACIONAMENTO ENTRE DUAS TABELAS.....	25
FIGURA 2.12 - AUSÊNCIA DE RELACIONAMENTO ENTRE DUAS TABELAS.....	26
FIGURA 3.1 - MODELAGEM DA NOVA BASE.....	32
FIGURA 3.2 - CONSULTA REALIZADA NO MODO DESIGN DO ACCESS.....	41
FIGURA 3.3 - ELABORAÇÃO NO MS-EXCEL DOS SCRIPTS DE CARGAS.....	43
FIGURA 3.4 - RESULTSET DE UMA CONSULTA NA BASE MYSQL.....	44
FIGURA 4.1 - TABELA DE IDENTIFICAÇÃO.....	46
FIGURA 4.2 - TABELA DE QUESTÕES.....	46
FIGURA 4.3 - TABELA DE ALTERNATIVAS.....	46
FIGURA 4.4 - TABELA DE RESPOSTAS.....	46
FIGURA 4.5 - RESULTSET DA TABELA DE QUESTÕES.....	47
FIGURA 4.6 - RESULTSET DA TABELA DE ALTERNATIVAS – QUESTÃO 10	47
FIGURA 4.7 - RESULTSET DA TABELA DE ALTERNATIVAS – QUESTÃO 9.....	48
FIGURA 4.8 - RESULTSET DA CONSULTA DE CONTAGEM.....	49

LISTA DE TABELAS

TABELA 2.1 – DESCRIÇÃO DE TABELAS ACCESS	14
TABELA 2.2 – CHAVES DUPLICADAS.	287
TABELA 2.3 - RESULTADO DE PACIENTES COM MESMA CHAVE.	28

RESUMO

O presente trabalho apresenta como foco principal a reengenharia de uma base de dados existente em uma clínica médica da cidade de Porto Alegre, a Clínica do Sono.

A base de dados foi migrada de uma plataforma MS-Access para uma MySQL. Além disso, realizou-se uma avaliação prévia dos dados presentes na base Access, assinalando todos os problemas existentes tanto sobre o ponto de vista de modelagem, quanto sobre o dos tipos de dados presentes nas tabelas.

Após esta análise, é apresentada uma estratégia para a migração da maioria dos dados presentes na base antiga, para a nova base. Esta migração não é realizada na sua completude porque existe uma certa dependência de análise do usuário para a validação de alguns dados.

Por fim, encontra-se um exemplo de elaboração de uma pesquisa sobre a nova base para a obtenção de dados estatísticos relacionados aos pacientes, métricas e métodos de tratamentos. Esta pesquisa visa uma futura elaboração de uma ferramenta de BI para uso da clínica também.

Palavras-Chave: Clínica, sono, reengenharia, base de dados.

Re-engineering of a Database and Research Over Statistics Data from a Medical Clinic – Clínica do Sono.

ABSTRACT

The current project has as main goal the re-engineering of the existing database at Clínica do Sono, which is a medical clinic in Porto Alegre.

The database was migrated from MS-Access to a MySQL platform. Besides that, it was performed a pre-evaluation of the data available in the MS Access database, pointing all the existing problems related to the model and to the data types used on that database.

After a detailed description of the analysis performed, its presented the strategy proposed to migrate most of the data to the new database. This migration was not complete because of the needed dependency on the analysis from the main user to validate some of the data.

Finally, there is provided a possible search to be used over the new database to generate statistic information related to patients, metrics and methods of treatment. As future work it is planned the design of a BI tool to be used at the clinic.

Keywords: Clinic, sono, re-engineering, data base.

1 INTRODUÇÃO

A Clínica do Sono é um consultório voltado para a pesquisa e o tratamento de doenças relacionadas aos distúrbios do sono. Atualmente, a mesma conta com um cadastro de aproximadamente 15.000 pacientes. Este encontra-se armazenado em uma base de dados criada pelo proprietário da clínica, e co-orientador deste trabalho, Dr. Dênis Martinez.

A base de cadastro da clínica está sobre a plataforma Access. Devido ao elevado crescimento da base de dados, e de sua complexidade, o Dr. Dênis solicitou sua reformulação. Além disso, a base apresenta diversos problemas relacionados com a redundância de dados, replicação de entidades, falta de relacionamentos, dados agrupados de maneira incoerente, entre outros fatores.

Para a reengenharia da base, como o objetivo era deixar a mesma o mais genérica possível para que suportasse todos os tipos de casos impostos pela real necessidade do usuário, foi proposta uma solução que fizesse com que a base apresentasse uma normalização na terceira forma normal. Além disso, como o estado inicial da base da clínica apresentava-se extremamente confuso, com diversos problemas, e a proposta deste trabalho visa também a reorganização dos dados presentes, foi adotada esta forma. No capítulo 3 será detalhado melhor a questão do porquê se adotou esta solução.

Para tanto, neste trabalho propõe-se uma solução sobre a plataforma MySQL, por se tratar de um banco de dados robusto, de software livre, multiusuário, além de o usuário já possuir uma base na mesma plataforma que alimenta o site da clínica. Este último fator é muito importante pois, se em algum momento o usuário necessitar uma integração entre a aplicação intranet e a internet, o processo será facilitado, pois ambas as bases estarão sobre a mesma plataforma.

Além da reengenharia, foi realizada uma pesquisa e uma reformulação dos dados sobre a base atual em Access. Esta reformulação realizou a eliminação de redundâncias, entre outros problemas descritos neste trabalho, a fim de se obter uma base de dados fiel e correta com a realidade da clínica.

Outro problema levantado pelo usuário foi com relação à obtenção de dados estatísticos, visto que a base de dados em Access encontrava-se muito complexa e incorreta para que fosse possível a obtenção de tais informações. Foi com este propósito que também foi elaborada uma série de consultas visando exclusivamente a obtenção de dados estatísticos sobre os diversos casos de pacientes, que futuramente poderão ser utilizados também para a elaboração de uma ferramenta de *business intelligence*. O objetivo destas consultas é principalmente proporcionar a possibilidade de tomar determinadas decisões sobre a administração da clínica, não somente sobre o ponto de vista financeiro, mas também a respeito de métricas, metodologias, atividades

relacionadas aos pacientes e exames realizados pelos mesmos, a partir dos dados levantados nas consultas. Logo, o trabalho proposto nesta monografia apresenta como foco principal a reengenharia de uma base de dados existente em uma clínica média e, adicionalmente, a elaboração de pesquisas relacionadas à realidade da mesma, visando obter dados estatísticos sobre os pacientes, para que se possa obter um determinado direcionamento na administração da clínica, a fim de se chegar a um padrão de atendimento, consultas, exames e pesquisa ideal.

Neste trabalho, serão apresentados os diversos casos de problemas encontrados na base atual, além da solução proposta para a reengenharia da base, e finalmente as consultas relacionadas à extração das informações para dados estatísticos.

O presente trabalho apresenta-se organizado da seguinte forma. No capítulo 2 deste trabalho, será apresentada uma análise completa sobre a base de dados existente na plataforma Access. Nesse capítulo serão apresentados todos os problemas verificados na base, além da apresentação completa da base, com suas entidades e relacionamentos. Ao final deste capítulo será possível perceber que a maneira com que foi modelada a base é que na realidade ela era voltada exclusivamente ao suporte dos formulários impostos pelo usuário aos pacientes. Porém, este paradigma será alterado a fim de se atender não somente os formulários, mas também uma melhor organização dos dados e também a facilitação na obtenção de dados, principalmente para pesquisas estatísticas.

No capítulo 3, será apresentada a sugestão da nova modelagem da base. Para tal, será utilizada uma ferramenta de modelagem de dados a ser descrita mais adiante. Esta nova modelagem apresentará as entidades reorganizadas sobre uma nova perspectiva, se comparada a modelagem em Access. Esta perspectiva é voltada para atender não somente os formulários presentes na clínica, mas também a nova aplicação web que foi desenvolvida e a facilitação na obtenção de dados estatísticos, fazendo com que fique mais fácil o cruzamento de informações e a análise das mesmas.

No capítulo 4 e último deste trabalho, encontra-se um exemplo de consulta sobre a nova base para que o usuário possa utilizar para obter dados estatísticos de seus pacientes e assim adotar determinadas decisões perante a administração de sua clínica em qualquer âmbito, porém o foco principal é sobre os resultados obtidos com seus formulários. Estes dados levantados nas consultas visam, principalmente, fazer com que o usuário tenha algum parâmetro de validação da eficácia de seus formulários e metodologias de tratamentos perante seus pacientes. Com posse destes dados, o usuário poderá alterar, reformular, excluir algumas perguntas a fim de que possa obter o máximo de informação relevante para o bom andamento de seus tratamentos, entre outros fatores.

2 ANÁLISE DA BASE DE DADOS EXISTENTE

Primeiramente, será apresentada uma overview da base de dados existente na clínica, que é uma base sobre a plataforma MS-Access. A Figura 2.1 apresenta todas as tabelas que compõem a mesma.



Figura 2.1- Apresentação superficial da base de dados em Access.

2.1 Descrição das tabelas que compõem a base atual

A seguir é listada uma breve descrição sobre cada tabela presente na base e o que a mesma apresenta de dados.

A partir da listagem, será possível perceber que a organização da base é voltada principalmente para atender aos questionários realizados pelo usuário aos seus pacientes.

Para a listagem que segue, as tabelas serão agrupadas por tipo de utilização que as mesmas apresentam na base de dados Access.

Tabelas de dados cadastrais	
<ul style="list-style-type: none"> • IDENTIFICAÇÃO 2006 	<p>Descrição: estas tabelas apresentam dados cadastrais dos pacientes principalmente. Dados como nome, idade, documento de identidade, data de nascimento, sexo, dentre outros.</p> <p>Problema: nestas tabelas encontram-se muitos dados redundantes, inclusive chaves (PCOD). Além disso, existem atributos que deveriam ser relacionados à outra entidade, como data de consulta, numero de cigarros que fuma, data que começou a fumar, etc.</p>
Tabelas de questionários	
<ul style="list-style-type: none"> • alfa-delta sono leve e dor • Atenas • atenas+sono1 • auriculoterapia apnéia leve 18-50 anos • Berlin • BOA NOITE • BOM DIA • CONTRAN TOTAL • Epworth 	<p>Descrição: tabelas utilizadas para armazenar dados levantados através de questionários respondidos pelos pacientes. Apresentam atributos que correspondem à questões como: fadiga, pensamentos, despertares, entre outros.</p> <p>Problema: nestas tabelas, existem atributos que na realidade devem ser dados de tabelas. O mapeamento realizado para estes tipos de atributos será apresentado nos capítulos seguintes.</p>
Tabelas de dados clínicos, exames e diagnósticos	
<ul style="list-style-type: none"> • TABELA DADOS CLÍNICOS • NARCOLEPSIA • ECOCARDIO • OBESIDADE • PARALISIA DO SONO • Sonambulismo • PSG TOTAL • CONTRAN TOTAL_1 	<p>Descrição: tabelas utilizadas para armazenar dados referentes a informações clínicas dos pacientes como diagnósticos, cataplexia, alucinações, paralisia, etc.</p> <p>Problema: apresenta atributos que na verdade devem ser valores de tabelas.</p>
Tabelas de dados diversos	
<ul style="list-style-type: none"> • MEDICOS NATAL • 16 MIL NUMEROS EM SEQUENCIA • CONVENIO • DADOS RETORNO 	<p>Descrição: tabelas utilizadas para outras finalidades que não sejam atender aos questionários, nem aos dados clínicos dos pacientes ou dados de identificação.</p> <p>Problema: algumas destas tabelas não apresentam relevância alguma, ou não são necessárias perante a nova modelagem, como a tabela de 16 mil números em sequência, por exemplo.</p>

Tabela 2.1 - Descrição de tabelas Access

A partir da análise da base atual, percebeu-se que existem diversos problemas na mesma. Dentre os problemas analisados, podemos citar como principais:

1. Redundância de informações: a redundância de informação é um problema grave para a base de dados visto que este fato significa que uma mesma informação encontra-se em mais de um local sem que nenhum relacionamento entre as tabelas exista. Além disso, este problema acarreta o fato de que a base pode vir a ficar incoerente e incorreta, pois, em um dado instante, o usuário age sobre uma tabela e em um momento seguinte, age sobre outra tabela que contém as mesmas informações.

Por exemplo, supondo-se que exista o campo idade na tabela de identificação e na tabela de dados_clínicos. Na entrada do paciente, é registrada sua idade e então ele fica um ano sem ir novamente à clínica. No ano seguinte, como ele já encontra-se cadastrado, ele não irá atualizar seus dados. No momento da consulta então, o paciente informa sua idade (agora acrescida de 1 unidade) e é gravado na tabela de dados_clínicos a nova idade. Ao realizar uma consulta na base, utilizando a tabela de identificação, o resultado será uma consulta incorreta visto que a idade que encontra-se cadastrada é a do ano anterior.

A partir do exemplo apresentado, percebe-se a importância de se ter os dados sempre atualizados e em um único local, caso partirmos do pressuposto de que não exista qualquer relacionamento entre entidades.

2. Presença de entidades irrelevantes sob o ponto de vista estrutural da base: existem algumas tabelas presentes no banco Access que, na verdade, deveriam ser atributos ou até mesmo nem deveriam existir na base pois são irrelevantes para o mapeamento da realidade em que se encontra a clínica. O maior problema da existência de tipos como este é a presença de desperdício de recurso, sem contar com que prejudica a manutenibilidade da base.

Por exemplo, supondo-se que exista a tabela e_mails. Nesta, existem dois atributos que seriam o cd_pessoa e o ds_email. Para cada alteração de um e-mail, baseado no nome de um paciente, deveria ser realizado uma junção com a tabela de identificação de pacientes, para que se descubra o código do paciente e posteriormente a realização da atualização. Esta operação é mais custosa do que simplesmente se realizar uma atualização diretamente na tabela de identificação buscando pelo nome do paciente.

3. Existência de algumas entidades que não condizem com a realidade da clínica: a base apresenta algumas tabelas que não apresentam um relacionamento bem embasado sobre a realidade da clínica. Com isso, acarreta-se um desperdício de recurso, bem com se abre margem para a utilização incorreta desta entidade. Por exemplo, um administrador do sistema que não tenha muito conhecimento da finalidade da tabela, acaba utilizando a mesma para armazenar uma informação que não deveria ser salva nessa tabela.
4. Atributos com mesma informação, porém com nomes diferentes: existem alguns atributos de tabelas que possuem o mesmo tipo de informação, porém os nomes dos mesmos são diferentes. Este fato prejudica muito as consultas, ou qualquer que seja o tipo de manipulação sobre os dados existentes. Na modelagem existente, podemos apresentar um exemplo claro deste problema. O mesmo será explicitado nos próximos tópicos.

5. Repetição de chaves em tabelas: em algumas tabelas, durante a realização de cargas de migração, principalmente na tabela de identificação, percebeu-se a existência de dois pacientes diferentes com um mesmo código. Trata-se de um erro extremamente grave pois a replicação de chave faz com que se perca o identificador único de uma entidade. Este problema, em um banco de dados relacional, dificilmente acontece, o que mostra que a base atual apresenta um certo nível de relacionamento, porém não pode ser considerada como tal.

A gravidade deste problema é tamanha pois, por exemplo, existindo a situação em que existem dois pacientes em um sistema de um hospital, A e B, ambos com um mesmo código. O paciente A apresenta uma determinada doença que requer um certo medicamento. Já o paciente B não necessita de remédios. Então o médico vai levantar a ficha do paciente B para verificar a necessidade de prescrever algum novo medicamento e então acaba recuperando a ficha do paciente A pois os dois possuem o mesmo código de paciente. Após prescrever tal medicamento, o setor de farmácia delega para o paciente B um medicamento que irá acarretar um problema grave para o mesmo.

Para uma melhor verificação dos problemas relatados acima, foi realizada uma abordagem em etapas, a serem descritas nas sub-seções que seguem.

2.2 Etapa 1: Reconhecimento de entidades irrelevantes ou que devem integrar outras entidades

Nesta etapa, foram listadas as entidades que aparentemente devem compor outra entidade, ou que são irrelevantes para a base. As imagens abaixo apresentam estes casos.

2.2.1 Tabela exercício

Nesta tabela, encontram-se armazenadas informações de código de paciente, exercício e o tempo despendido realizando o mesmo. Esta entidade tem por finalidade armazenar o tempo em que um paciente pratica alguma atividade física e o tipo de atividade. Nesta, a pesquisa sobre um determinado exercício físico fica impossibilitada devido ao tipo de dado que é armazenado. Esta tabela deveria estar integrada a alguma outra de dados clínicos do paciente.

A figura seguinte apresenta a tabela em questão.



exercício	
pcod	
exercício	
há quanto	

Figura 2.2 - Tabela exercício.

2.2.2 Tabela E-MAIL

Semelhante ao problema anterior, a tabela de e-mail apresenta um desperdício de recurso sobre o ponto de vista de modelagem. A mesma deve ser anexada à tabela de identificação de paciente. Adicionalmente, a busca em uma tabela como esta seria mais lenta pois para tal fato, deveria se criar uma junção com a tabela de identificação para que se obtenha o e-mail desejado de um determinado paciente. Em contrapartida, se esta informação estiver na tabela de identificação, então a busca por um e-mail será mais simples e menos custosa para o SGBD.

A figura seguinte apresenta a tabela em questão.



Figura 2.3 - Tabela E-MAIL.

2.2.3 Tabelas SCL90 e 16 MIL NÚMEROS EM SEQUENCIA

Ambas foram criadas com o intuito de geração de números sequenciais. Devem ser excluídas pois não existe a necessidade de se ter uma entidade para gerar *sequences*. Caso necessário, pode-se criar na base uma *sequence*, sem que se crie uma tabela para atender exclusivamente esta dependência.

A imagem a seguir apresenta a tabela em questão.

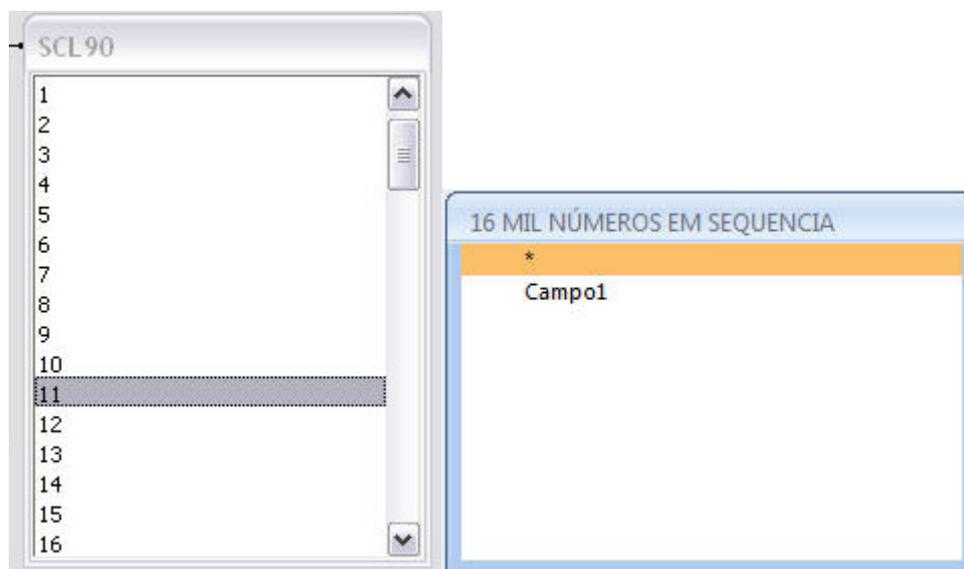


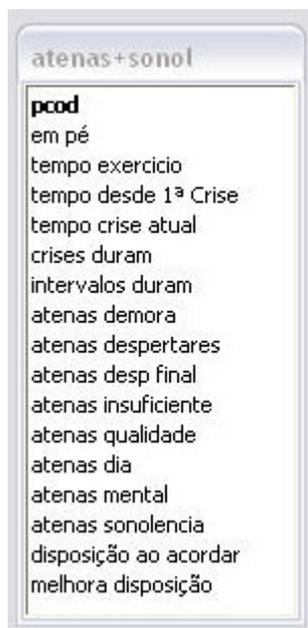
Figura 2.4 - Tabelas SCL 90 e 16 MIL NUMEROS EM SEQUENCIA.

2.2.4 Tabela atenas+sono1

A tabela foi criada com a finalidade de integrar informações provenientes de outras duas. Para esta questão, não existe a necessidade de se existir tal entidade. Será criada

uma consulta específica para se integrar tais dados, ou então pode-se resolver esta questão criando uma view na base, que englobe os atributos necessários para retornar essa consulta sempre. A adoção de um ou outro método depende da usabilidade e da necessidade de dados íntegros. Por exemplo, se o usuário necessitar retirar um relatório em qualquer horário e sempre que necessitar, então é instaurada a possibilidade de uma consulta montada para atender tal demanda. Caso o usuário necessite de relatórios diários, ou seja, ele vai retirar tal relatório em um determinado horário, todos os dias, uma única vez, então seria adotada a metodologia de views. Todos os dias, é rodada uma consulta que atualiza os dados da view. Por praticidade e como o usuário não necessita de desempenho, adotou-se a consulta que pode ser executada sempre que houver necessidade.

A imagem a seguir apresenta a tabela em questão.



pcod
em pé
tempo exercicio
tempo desde 1ª Crise
tempo crise atual
crises duram
intervalos duram
atenas demora
atenas despertares
atenas desp final
atenas insuficiente
atenas qualidade
atenas dia
atenas mental
atenas sonolencia
disposição ao acordar
melhora disposição

Figura 2.5 - Tabela atenas+sono1.

2.2.5 Tabela alfa-delta sono leve e dor

Esta tabela apresenta alguns atributos que devem integrar outros atributos. Por exemplo, os campos peso e altura devem fazer parte da tabela referente a dados clínicos. Além desta, existem outras tabelas que apresentam o mesmo problema de dados, a saber: BERLIN, TABELA DADOS CLINICOS.

A figura 2.2.5 seguir apresenta os problemas listados neste item.

alfa-delta sono leve e dor	BERLIN	TABELA DADOS CLINICOS
num PSG	PCOD	PCOD
PSG Total_PCOD	NPSG	IDADE
CODP	NMVV	DIAG1
PENSAMENTOS	DATA	DIAG2
DESPERTARES	PESO MUDOU	DIAG3
FADIGA	RONCA	DIAGDEF1
QUALIDADE	RONCA INTENSIDADE	DIAGDEF2
SONHO	RONCO FREQUÊNCIA	DIAGDEF3
SONOLÊNCIA	RONCO PERTURBA	MÉDICO
per	APNÉIA FREQUÊNCIA	HDA
GRAU	CANSAÇO	EXAME CLÍNICO_data
ttd	CANSAÇO DIA	PLANO
es	COCHILA	SONOLÊNCIA
I2	HAS	SONO_TEMPO
Is	PESO	HORA
Ir	ALTURA	VENDOTV
pe2	COLARINHO	AULAS
pe34	CINTURA	LENDO
TABELA DADOS CLINICOS_PCOD	PA SIST	VIAJANDO
SEVO	PA DIAST	DIRIGINDO
IDADE	DIAS UTEIS	TRAB
OMEVA	ENC DE SEMANA	ESCRIVE

Figura 2.6 - Tabelas com atributos incorretos.

2.3 Etapa 2: Verificação de entidades, relacionamentos ou atributos redundantes

Nesta etapa, foram listadas todas as entidades, relacionamentos ou atributos que apresentam replicação de informação em outras. Podemos destacar principalmente a presença de duas tabelas exatamente iguais – IDENTIFICAÇÃO 2006 e IDENTIFICAÇÃO 2006_1. A razão da presença destas duas tabelas foi devido a uma tentativa do usuário realizar uma atualização dos dados cadastrais de todos os pacientes, através de uma re-digitação das fichas cadastrais que o mesmo possui arquivado em sua clínica.

O maior problema da redundância de informações, consiste na manipulação de seus dados. A probabilidade de que você fique com dados diferentes, para um mesmo objeto, em dois locais diferentes é muito grande.

Adotando o ponto de vista de usuário, é compreensível que se crie tal redundância, visto que o usuário não necessitará realizar qualquer junção com outra tabela para obter a informação necessária. Tomemos como exemplo o atributo NOME. Este, encontra-se presente em diversas tabelas pois, a fim de obter as informações relativas a um exame, questionário ou dado clínico, o usuário replicou o nome de seu paciente, para que não fosse necessário o acesso através de uma junção à tabela de identificação.

A figura 2.3 apresenta uma visão geral da base sinalizando os problemas de redundância de dados, para que se tenha uma idéia mais próxima da quantidade de dados, entidades e relacionamentos redundantes na base.

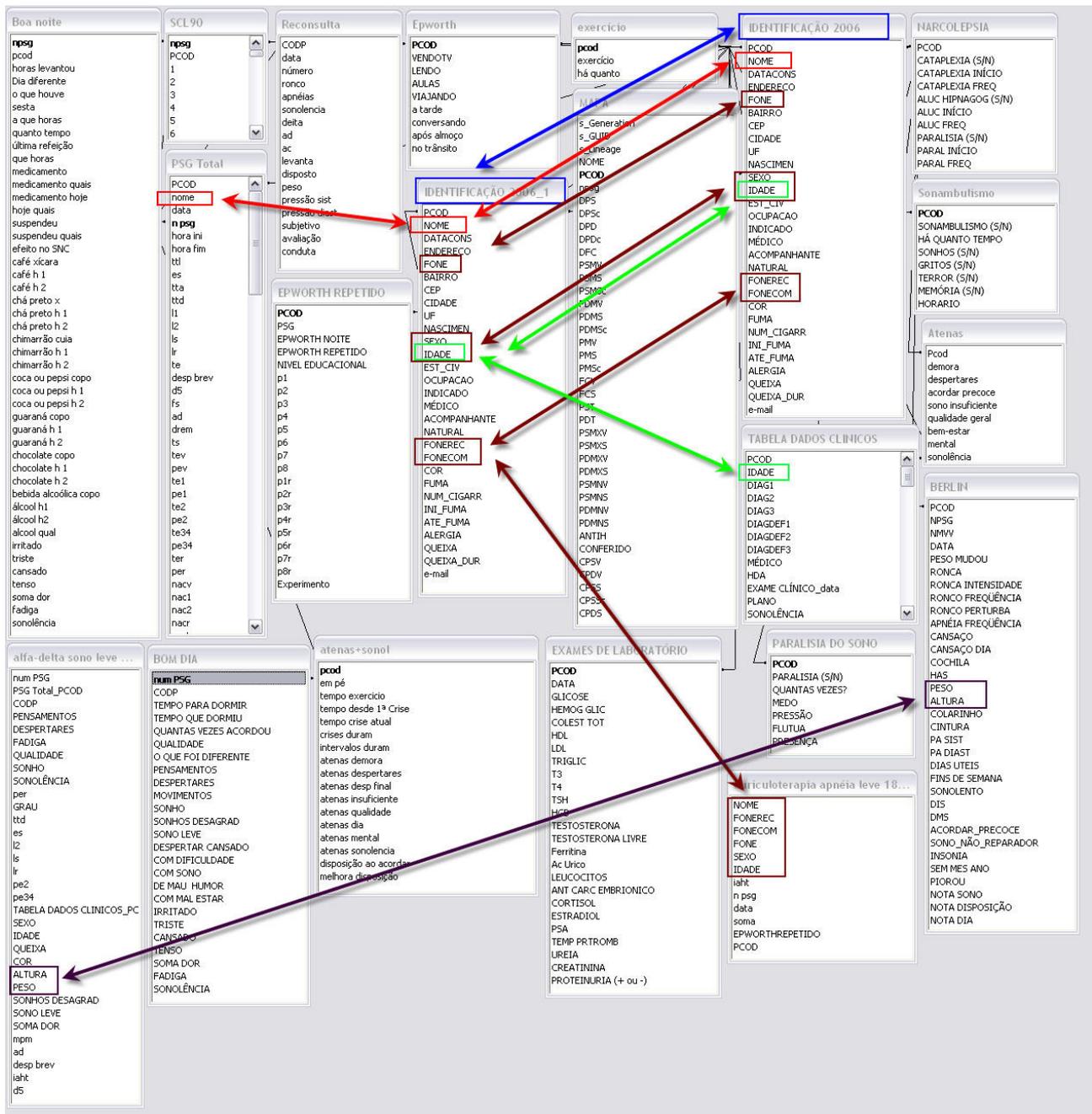


Figura 2.7 - Problema de redundância de entidades, relacionamentos ou atributos.

2.4 Etapa 3: Atributos com nomes diferentes, porém mesma finalidade

O atributo PCOD, que corresponde ao identificador único de um paciente, aparece em algumas entidades com o nome de CODP.

Trata-se de um grave problema pois, além de os atributos estiverem com nomes diferentes, o que dificulta as consultas e manipulações de dados, este campo é considerado uma chave primária na tabela de identificação e subentende-se que o mesmo seja uma chave estrangeira de outras tabelas.

Imaginemos a situação em que o usuário deseja realizar uma consulta na tabela de Reconsulta. Se o mesmo for realizar esta pesquisa através da montagem de uma

consulta SQL, a chance de ele buscar pela informação digitando esta chave como PCOD é grande pois ela compõe a chave da tabela de IDENTIFICAÇÃO 2006. Consequentemente a consulta não funcionará pois na tabela de Reconsulta este campo apresenta o alias de CODP.

Este tipo de problema é comum em bases Access criadas por usuários comuns, visto que suas consultas são montadas em modo design, aonde o usuário simplesmente seleciona a tabela que deseja consultar e arrasta os campos que deseja visualizar os valores. Após realizada a montagem, ele simplesmente executa a consulta. Para o usuário, é transparente este erro pois ele vai obter a informação sem que seja apresentado qualquer erro.

A figura a seguir explicita o problema relatado, destacando os campos que apresentam o mesmo.

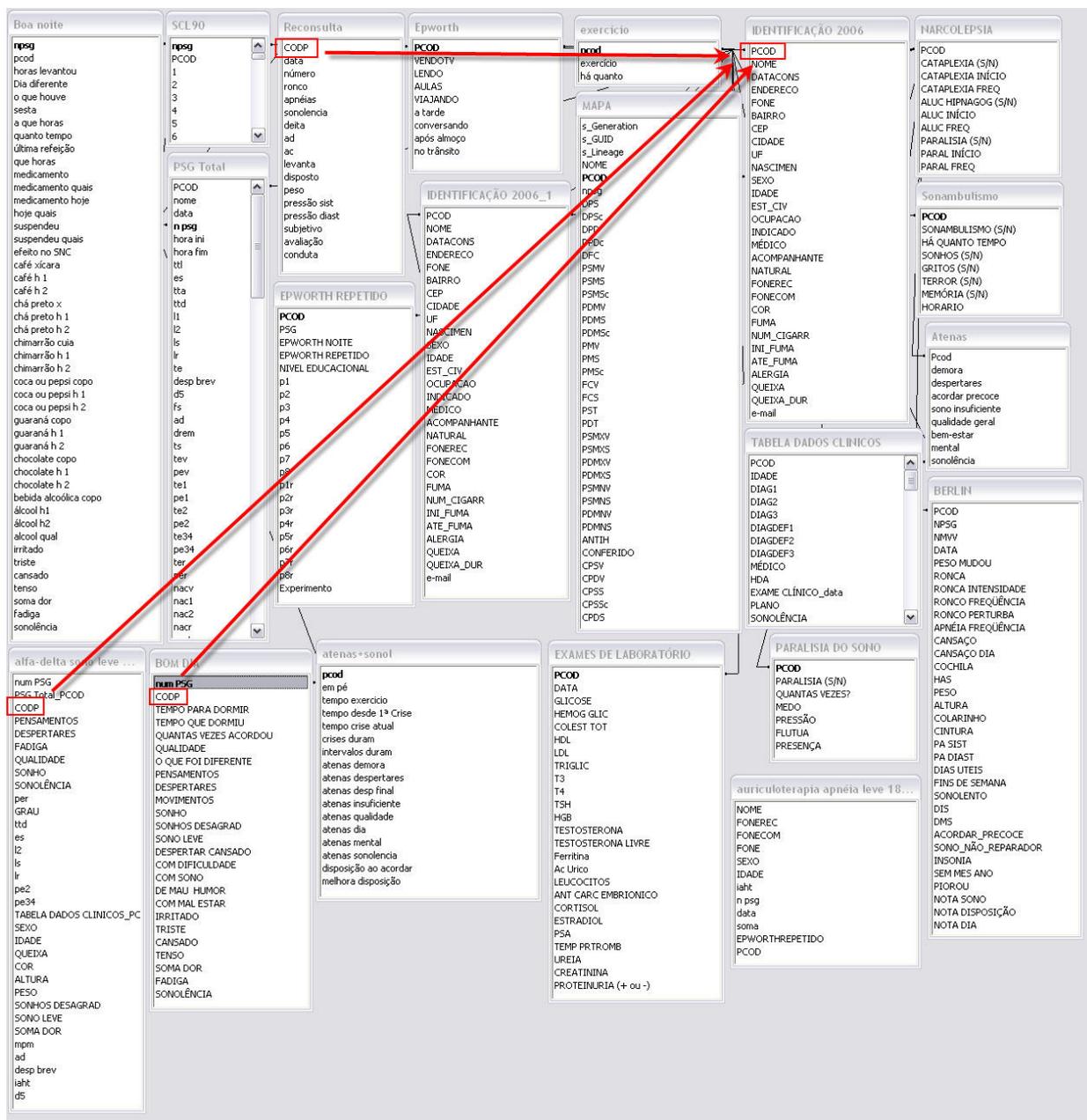


Figura 2.8 - Problema do mesmo tipo de informação em atributos com nomes diferentes

2.5 Etapa 4: Atributos que devem se tornar registros de tabelas

Alguns atributos possuem uma nomenclatura muito específica. A idéia é criar esse tipo de informação a partir de outras entidades (tabela de parametrização ou de questões).

Nesta etapa, serão listadas as tabelas presentes na base de dados que serão alteradas e transformadas em questionários elaborados pela aplicação. Este passo apresenta um dos principais problemas existente na base. Isto se deve ao fato que, para cada novo questionário existente, o usuário criava uma nova tabela, com as questões relacionadas ao mesmo. O objetivo aqui é fazer com que exista uma única fonte de questões e que estas sejam agrupadas por assunto ou que a própria aplicação se encarregue de elaborar os questionários no momento de sua execução. Esta alteração fará com que a base fique mais enxuta e menos susceptível à geração de dados redundantes em mais de uma entidade, dentre outros fatores.

Por se tratar de um problema que ocorre em diversas entidades da base, será apresentada uma imagem com as tabelas destacadas, indicando quais sofrerão tal tipo de alteração. Em algumas destas tabelas, pode-se perceber também a presença de redundância de informações, mas este assunto será tratado em um tópico futuro. A seguir serão apresentadas as tabelas que sofrerão tal alteração, inclusive explicitando os campos que serão transformados em dados de questões, ao invés de serem considerados atributos da base.

Abaixo encontram-se os campos com que apresentam esse problema. Todos os atributos assinalados, serão transformados em valores de tabelas.

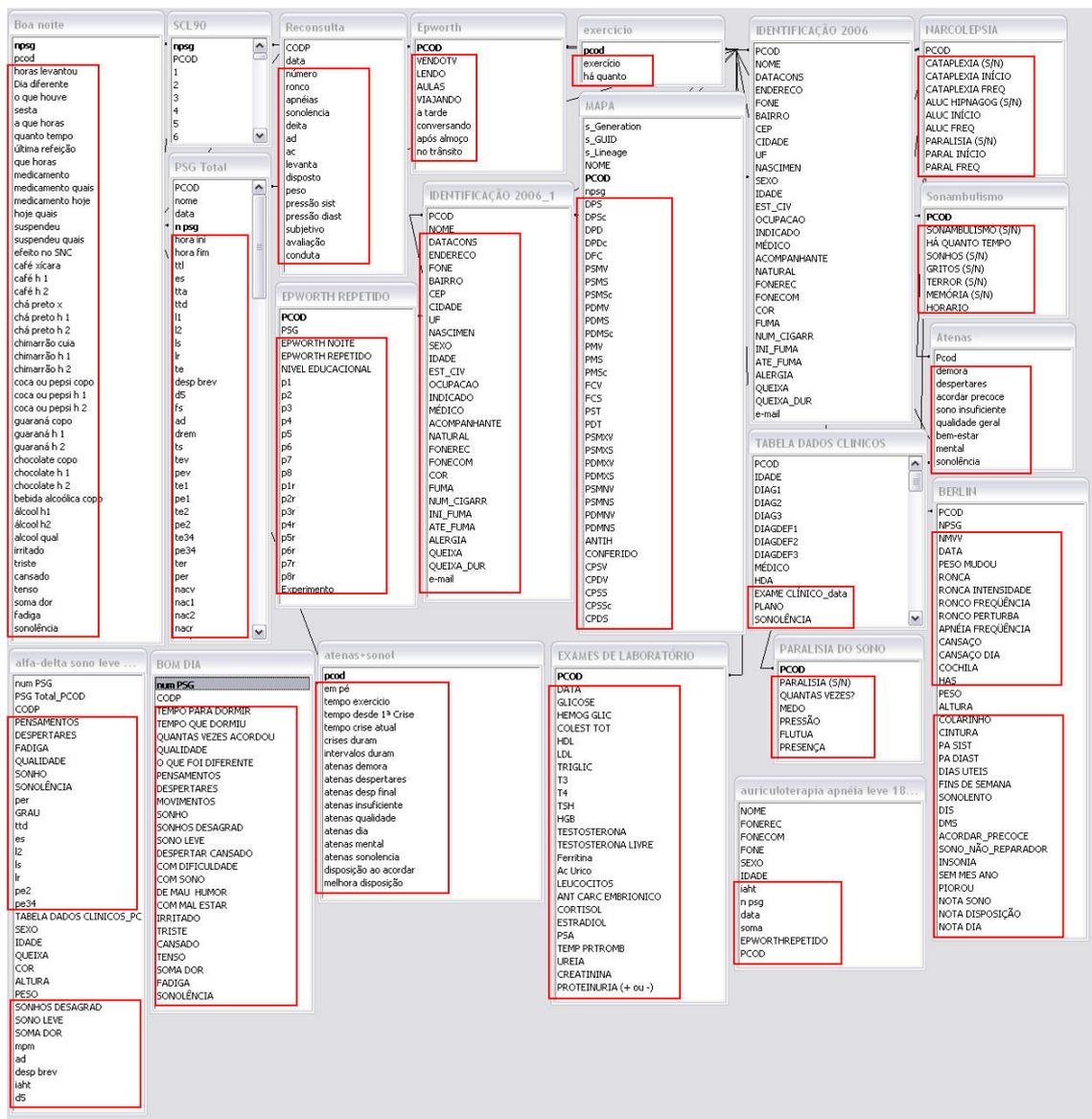


Figura 2.9 - Atributos que devem se tornar registros em tabelas.

2.6 Etapa 5: Entidades que não apresentam o devido relacionamento

Nesta etapa, verificou-se que algumas entidades não apresentam o devido relacionamento, como ocorre, por exemplo, entre as tabelas alfa-delta sono leve e dor e IDENTIFICAÇÃO 2006 (para a chave PCOD).

Sob o ponto de vista de modelagem relacional, este erro é considerado grave pois elimina o devido relacionamento entre duas entidades que deveriam apresentar o mesmo. O problema de não existir o relacionamento é que, para a tabela de alfa-delta sono leve e dor não existirá a garantia de que a informação para um determinado paciente com um certo código seja o mesmo paciente com mesmo código presente na tabela de IDENTIFICAÇÃO 2006.

Verificou-se a ocorrência desse problema entre as tabelas de IDENTIFICAÇÃO 2006 e:

- alfa-delta sono leve e dor

- atenas + sono1
- auriculoterapia apnéia leve 18-50 anos

Na imagens que seguem, estão apresentadas as ocorrências desse problema.

alfa-delta sono leve e dor	IDENTIFICAÇÃO 2006
num PSG	PCOD
PSG Total_PCOD	NOME
CODP	DATACONS
PENSAMENTOS	ENDERECO
DESPERTARES	FONE
FADIGA	BAIRRO
QUALIDADE	CEP
SONHO	CIDADE
SONOLÊNCIA	UF
per	NASCIMEN
GRAU	SEXO
ttd	IDADE
es	EST_CIV
I2	OCUPACAO
Is	INDICADO
Ir	MÉDICO
pe2	ACOMPANHANTE
pe34	NATURAL
TABELA DADOS CLINIC	FONEREC
SEXO	FONECOM
IDADE	COR
QUEIXA	FUMA
COR	NUM_CIGARR
ALTURA	INI_FUMA
PESO	ATE_FUMA
	ALERGIA
	QUEIXA
	OUFIXA DUR

Figura 2.10 - Ausência de relacionamento entre duas tabelas.

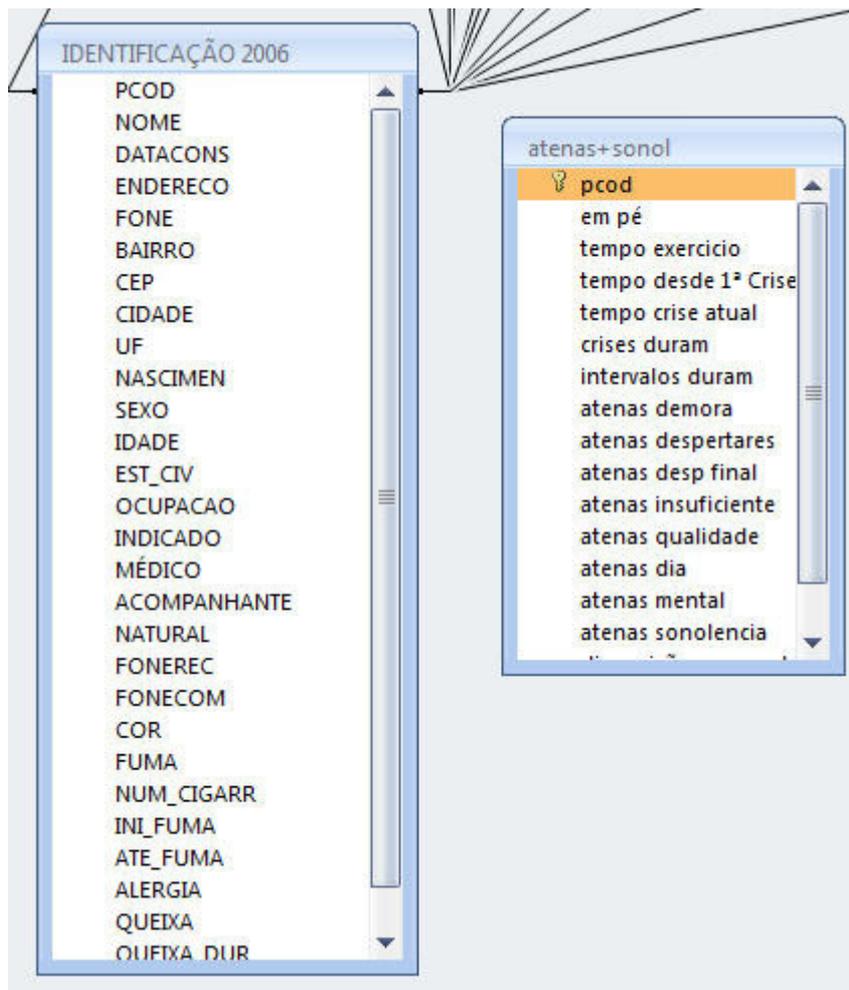


Figura 2.11 - Ausência de relacionamento entre duas tabelas.

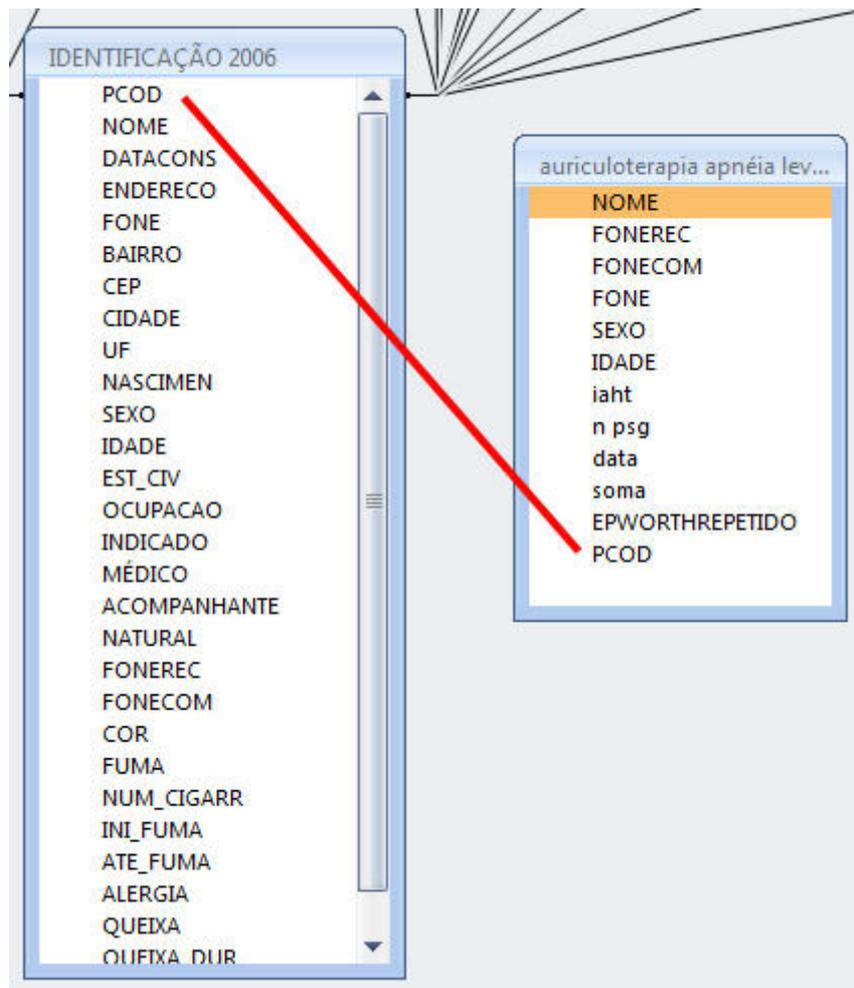


Figura 2.12 - Ausência de relacionamento entre duas tabelas.

2.7 Etapa 6: Repetição de chave em tabelas

A repetição de chave em uma mesma tabela é um grande problema não somente para o modelo relacional, mas para qualquer tipo de base de dados. A chave primária é caracterizada por ser um identificador único em uma tabela, o que garante a atomicidade de uma determinada informação. A partir do momento que este fato não é garantido, abre-se margem a ocorrências de diversos erros, como foi listado em um dos exemplos explanados no início do tópico 3.

Outro problema encontrado foi a dificuldade em se tratar a duplicidade de chave devido ao relacionamento que esta possuía com outras tabelas. Por exemplo, a tabela IDENTIFICAÇÃO 2006, que é uma das que apresenta este problema, possui relacionamento com outras tabelas, como a de PARALISIA DO SONO. Agora, como saber que os dados na tabela de paralisia para o código de paciente 1 pertence ao paciente que apresenta a primeira ocorrência de 1 na tabela de identificação ou a segunda ocorrência desse mesmo valor na mesma tabela. Não tem como. A solução encontrada será descrita nos capítulos seguintes, os quais irão tratar das soluções para a remodelagem desta base.

As tabelas a seguir explicitam a ocorrência deste problema. Foi possível adquirir esta informação através da seguinte consulta executada no banco Access.

```

SELECT [IDENTIFICAÇÃO 2006].PCOD, COUNT([IDENTIFICAÇÃO 2006].PCOD)
FROM [IDENTIFICAÇÃO 2006]
GROUP BY [IDENTIFICAÇÃO 2006].PCOD
ORDER BY 2 DESC;

```

Os códigos que apresentaram tal problema são os apresentados a seguir:

Consulta	
PCOD	Expr1
14508	3
13380	3
12062	2
15176	2
13720	2
14140	2
13608	2
13712	2
13708	2
13611	2
15251	2
14634	2
5379	2
13682	2
13633	2
13668	2
13666	2
11969	2
13979	2
13654	2
13431	2
13874	2
14160	2
14707	2
15042	2
12292	2
14376	2
14207	2

Tabela 2.2 - Chaves duplicadas

Podemos perceber esta ocorrência executando a seguinte consulta para um dos códigos apresentados acima:

```
SELECT [IDENTIFICAÇÃO 2006].PCOD, [IDENTIFICAÇÃO 2006].NOME
FROM [IDENTIFICAÇÃO 2006]
WHERE ((([IDENTIFICAÇÃO 2006].PCOD)=14508));
```

A partir desta consulta obtiveram-se os seguintes resultados:

Consulta	
PCOD	NOME
14508	JOAO DA SILVA
14508	PAULA DA SILVA
14508	JOÃO DA SILVA

Tabela 2.3 - Resultado de pacientes com mesma chave.

Este fato comprova a existência de mais de uma pessoa para uma mesma chave.

2.8 Etapa 7: Adequação da base para atender a aplicação web

Em conjunto a modelagem da nova base, foi desenvolvida uma aplicação web para atender a necessidade do usuário em possuir os formulários utilizados na clínica em ambiente web. Assim, o paciente pode preencher os questionários através do acesso à internet, o que facilitaria o trabalho de alguns setores da clínica bem como agiliza o processo de preenchimento de dados.

A fim de tornar compatível a nova base com a aplicação proposta é que a modelagem de dados da nova base foi baseada no formato que a aplicação web se comporta. Ainda não foi realizada uma integração entre as partes mas acredita-se que, com a conclusão deste trabalho, as alterações na nova base serão mínimas a fim de se atender na totalidade a aplicação web.

As demandas impostas pela aplicação são o controle de acesso de usuários, possibilidade de edição de questões que compõe os formulários, necessidade em se atender a criação de novos valores de alternativas para as questões ou dados utilizados nos formulários dinâmicos, possibilitar a integração de um grupo de questões junto a um determinado questionário, entre outros fatores.

3 DETALHAMENTO DA SOLUÇÃO

A primeira etapa deste trabalho apresenta um novo diagrama entidade-relacionamento a fim de realizar uma reorganização dos dados atuais. Além disso, foi realizada a criação de uma nova base de dados sobre um sistema de gerenciamento de base de dados novo, a ser descrito no decorrer deste trabalho.

Para a obtenção deste diagrama, foram executados os passos básicos para uma modelagem de uma base de dados: elaboração de um modelo conceitual, de um modelo lógico e de um modelo físico.

O modelo conceitual representa as regras de negócio, sem limitações tecnológicas ou de implementação. Neste modelo temos a visão geral do negócio, a facilitação do entendimento entre o usuário e desenvolvedores, possui somente as entidades e atributos principais, pode conter relacionamentos n para m.

No modelo lógico, são levados em conta os limites impostos por algum tipo de tecnologia de banco de dados. Suas características apresentam a derivação do modelo conceitual e representação do negócio, apresentação de entidades associativas no lugar dos relacionamentos n:m, definição das chaves primárias das entidades, normalização até a terceira forma normal, adequação ao padrão de nomenclatura e a documentação de entidades e atributos.

Para a modelagem física, são levados em consideração os limites impostos pelo SGBD e pelos requisitos não funcionais dos programas que acessam os dados. Esta fase apresenta como características a elaboração a partir do modelo lógico, algumas variações segundo o SGBD, a presença de tabelas físicas e de colunas físicas. O objetivo dessa remodelagem de dados consistiu não somente na organização dos dados atuais, presentes na base Access, mas também em atender as demandas impostas pelo ambiente web que compõe o novo sistema da clínica, bem como se tratar de uma possível solução futura para a elaboração de uma ferramenta de BI.

Além disso, buscou-se por uma modelagem que facilitasse o entendimento e manipulação de dados, fazendo com que a mesma fosse o mais genérica e maleável possível.

As etapas executadas para a remodelagem da base de dados foram as descritas abaixo:

1. Instalação e configuração do MySQL Server.
2. Análise das entidades e relacionamentos presentes na base atual em MS-Access.
3. Análise dos dados existentes, bem como seus tipos e relevância.

4. Modelagem da nova base de dados, utilizando a ferramenta DBDesigner, explicitando os novos atributos, relacionamentos, entre outros fatores essenciais para a criação de uma nova base de dados.
5. Criação das novas tabelas na base de dados existente, a partir dos scripts em SQL gerados a partir da ferramenta DBDesigner.

Para o desenvolvimento desse projeto utilizaram-se os seguintes softwares:

MySQL Server: banco de dados MySQL.

SQL Manager Lite: aplicação para a manipulação da base de dados. Permite o gerenciamento da base em uma interface gráfica, mais amigável, sem necessidade de se acessar a linha de comando do servidor de banco de dados.

Para a elaboração do modelo conceitual, foram levadas em considerações algumas colocações feitas pelo usuário durante reuniões. Dentre as principais, podemos destacar a eliminação de redundâncias de dados, de entidades e de atributos. Além disso, as características da clínica foram desmembradas em 6 grupos, a serem listados abaixo:

1. Dados clínicos: grupo que engloba todas as questões referentes a exames e dados clínicos relacionados aos pacientes.
2. Questionários: grupo que integra o conjunto de questões, alternativas e respostas fornecidas pelos pacientes, aos diversos questionários que a clínica apresenta. Estes questionários são de suma importância, visto que a avaliação de métodos de tratamentos, diagnósticos, entre outros fatores, envolvem a avaliação das respostas dos pacientes a estes questionários.
3. Financeiro: grupo que engloba dados referentes a faturamentos de consultas, exames, dentre outros que envolvem questões financeiras relacionadas à clínica.
4. Identificação: entidades que serão responsáveis pelos dados cadastrais dos pacientes.
5. Convênios: entidades que contém informações referentes aos convênios que os pacientes apresentam.
6. Médicos: cadastro de médicos que indicam a clínica para a realização de exames e tratamentos. É de grande interesse do usuário manter este cadastro a fim de apresentar um bom relacionamento da clínica perante os demais médicos.

Com base nessas informações levantadas pelo usuário é que se realizou uma modelagem conceitual do que seria a nova base de dados. Na modelagem conceitual foram determinadas as seguintes entidades:

1. Identificação (cd_paciente, nm_paciente).
2. Questoes (cd_questao, ds_questao).
3. Questionarios (cd_questionario, ds_questionario).
4. Diagnosticos (cd_diagnostico, ds_diagnostico).
5. Medicos (cd_medico, nm_medico).
6. Financeiro (dt_faturamento, vl_cobranca).

7. Consultas (dt_consulta, ds_consulta).

8. Dados Clinicos (cd_dado_clinico, ds_dado_clinico).

Com posse dessas entidades prévias, é que se realizou a modelagem lógica da base. Nesta, apresentam-se todas as entidades e todos os atributos, bem como os relacionamentos que serão criados.

Para a modelagem lógica da base, foi utilizada a ferramenta DBDesign. Trata-se de uma ferramenta livre, com uma interface muito amigável e de fácil manuseio. Além disso, a ferramenta apresenta a opção de exportar o modelo criado para um arquivo do tipo txt ou SQL, com os scripts de criação das tabelas apresentadas. Esta opção foi de grande valia visto que poupou trabalho de digitação dos scripts, fato que poderia ocasionar diversos erros.

Além de facilitar a modelagem, a ferramenta propicia uma visualização muito boa do modelo. Este fator foi utilizado diversas vezes nas reuniões tanto com o usuário, quanto com as demais partes envolvidas no projeto.

Pode-se verificar a remodelagem da nova base a partir do diagrama apresentado abaixo.

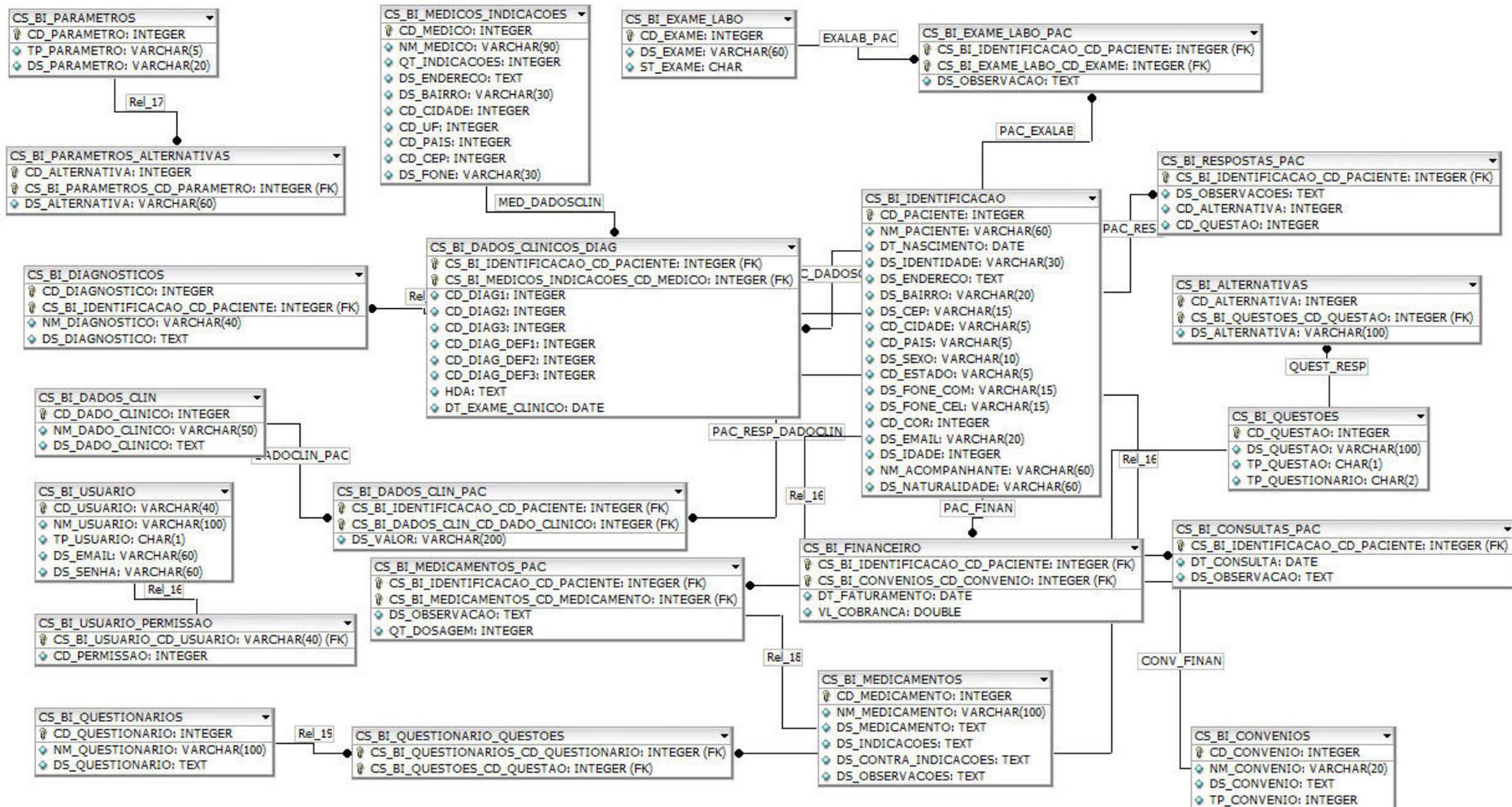


Figura 3.1 - Modelagem da nova base.

Nesta etapa, percebe-se que foram criadas diversas novas entidades, atributos e relacionamentos, a fim de que o modelo apresentasse a maior flexibilidade possível para atender as necessidades do usuário e também da aplicação web desenvolvida para atuar sobre a base.

O modelo proposto visa estar na terceira forma normal. Uma relação encontra-se na terceira forma normal se a mesma encontrar-se na segunda forma normal e todos os atributos não chave forem dependentes não transitivos da chave primária. Para se chegar na terceira forma normal, adotam-se os seguintes procedimentos:

1. Identificar todos os atributos que são funcionalmente dependentes de outros atributos não chave;
2. Removê-los e criar uma nova entidade com os mesmos.

A chave primária desta nova entidade será o atributo para o qual os demais atributos removidos são dependentes.

Após esta normalização, as estruturas de dados estão projetadas para eliminar inconsistências e redundâncias dos dados, eliminando desta forma qualquer problema de atualização e operacionalização do sistema.

Para o desenvolvimento da nova base, a abordagem de desenvolvimento seguiu os seguintes passos:

1. Listagem de entidades: nesta etapa, deve-se listar as entidades que farão parte da coleção de dados. Cada uma pode ser considerada como entidade única que corresponde a um assunto ou tema.
2. Listagem dos atributos que cada entidade possui: neste momento, deverão ser listados os atributos relacionados a cada entidade. A importância dessa fase consiste no fato de você verificar se todos os fatores relacionados às entidades, ou seja, todas as informações que cada entidade contém, deverão compor a base de dados. Há dados que podem ser inúteis ou redundantes e que não devem ser incluídos na estrutura do banco de dados.
3. Transformação das entidades em tabelas: as entidades descritas na etapa 1 devem tornar-se tabelas e os atributos, colunas. É importante ressaltar que um domínio é um conjunto de valores que a coluna pode conter. Um domínio possui propriedades física e lógica.
4. Transformação de cada atributo das entidades em colunas da tabela;
5. Determinação das relações entre as entidades: para determinar as relações entre as entidades, deve-se escolher uma delas e observar como a mesma se relaciona a outra. Nem sempre toda a relação existente entre entidades é importante. Aquelas que forem consideradas importantes permitem que se modele o banco de dados de maneira a corresponder às situações do mundo real a que ele representa.
6. Determinação das chaves: uma chave corresponde a um identificador único em uma tabela. Pode ser o RG, número de matrícula ou qualquer combinação de caracteres que seja exclusiva. Nenhuma outra linha na tabela pode conter o valor da coluna chave. É importante ressaltar que não se deve utilizar uma determinada coluna como chave se existir a possibilidade da mesma possuir duplicidades.

7. Avaliação da estrutura modelada: trata-se de uma revisão da modelagem, procurando todas as falhas do projeto. Recomenda-se utilizar as seguintes questões:
 - a. Cada tabela possui um único assunto?
 - b. Cada tabela tem a sua coluna chave?
 - c. Os nomes das colunas ou domínios são únicos?
 - d. Existem relações entre as tabelas?
 - e. A tabela é fácil de usar?
8. Criação do banco de dados.

3.1 Descrição e significado das tabelas presentes no novo modelo de dados

A seguir, encontram-se descritos os significados dos campos, bem como o que cada tabela significa para a modelagem.

3.1.1 Tabela CS_BI_PARAMETROS

Tabela que apresenta parâmetros que são utilizados para parametrizar o sistema. Esta tabela é muito importante pois ela, juntamente com a de questões, substitui a grande maioria dos atributos da base atual.

- a. CD_PARAMETRO: chave primária da tabela;
- b. TP_PARAMETRO: tipo de parâmetro que está relacionado (cor, ocupação, sexo, etc). A partir desta informação, é possível selecionar os valores relacionados a cada tipo de dado que se necessita, ou qualquer outra informação que se deseja extrair do paciente nos formulários presentes na tanto aplicação web quanto nas consultas que são utilizadas para a obtenção de dados estatísticos.
- c. DS_PARAMETRO: nome do parâmetro (Sexo, Cor, etc).

3.1.2 Tabela CS_BI_PARAMETROS_ALTERNATIVAS

Tabela que possuirá as alternativas para a tabela de parâmetros.

- a. CD_ALTERNATIVA: chave primária da tabela;
- b. CS_BI_PARAMETROS_CD_PARAMETRO: chave estrangeira;
- c. DS_ALTERNATIVA: descrição que será apresentada para o dado parâmetro.

3.1.3 Tabela CS_BI_MEDICOS_INDICACOES

Tabela que irá armazenar informações sobre os médicos que realizam indicações da clínica aos seus pacientes.

- a. CD_MEDICO: chave primária da tabela;
- b. NM_MEDICO: nome do médico que indica o paciente;
- c. QT_INDICACOES: quantidade de indicações que o médico realizou;
- d. DS_ENDERECO: endereço do consultório;
- e. DS_BAIRRO: bairro;
- f. CD_UF: código da unidade federal;
- g. CD_PAIS: código do país;
- h. CD_CEP: cep;
- i. DS_FONE: telefone para contato.

3.1.4 Tabela CS_BI_MEDICAMENTOS

Tabela que irá armazenar informações sobre os medicamentos utilizados.

- a. CD_MEDICAMENTO: chave primária da tabela;
- b. NM_MEDICAMENTO: nome do medicamento;
- c. DS_MEDICAMENTO: breve descrição do medicamento. Resumo da bula;
- d. DS_INDICACOES: medicamento indicado para que finalidade;
- e. DS_CONTRA_INDICACOES: medicamento contra-indicado em que casos;
- f. DS_OBSERVACOES: demais observações a respeito do medicamento;

3.1.5 Tabela CS_BI_IDENTIFICACAO

Uma das principais tabelas da base. Possui informações cadastrais dos pacientes. Esta é a primeira tabela que é populada no momento da entrada de um novo paciente na clínica.

- a. CD_PACIENTE: chave primária da tabela;
- b. NM_PACIENTE: nome do paciente;
- c. DT_NASCIMENTO: data de nascimento do paciente;
- d. DS_IDENTIDADE: número do documento de identidade;
- e. DS_ENDERECO: endereço;
- f. DS_BAIRRO: bairro;

- g. DS_CEP: cep;
- h. CD_CIDADE: código da cidade;
- i. CD_PAIS: código do país;
- j. DS_SEXO: sexo do paciente;
- k. CD_ESTADO: código do estado;
- l. DS_FONE_COM: telefone comercial;
- m. DS_FONE_CEL: telefone celular;
- n. CD_COR: código da cor/raça;
- o. DS_EMAIL: endereço eletrônico;
- p. DS_IDADE: idade do paciente;
- q. NM_ACOMPANHANTE: nome do acompanhante do paciente. Algumas consultas ou exames exigem que o mesmo vá acompanhado de alguma pessoa;
- r. DS_NATURALIDADE: naturalidade.

3.1.6 Tabela CS_BI_USUARIO

Tabela que irá realizar o controle de acessos. Esta tabela será utilizada principalmente pela aplicação web.

- a. CD_USUARIO: chave primária;
- b. NM_USUARIO: nome do usuário;
- c. TP_USUARIO: tipo de usuário. Utilizado para permissões;
- d. DS_EMAIL: endereço eletrônico do usuário;
- e. DS_SENHA: senha de login.

3.1.7 Tabela CS_BI_QUESTOES

Tabela que possui o banco de questões utilizadas para os questionários/formulários presentes na clínica.

- a. CD_QUESTAO: chave primária;
- b. DS_QUESTAO: descrição da questão, enunciado;
- c. TP_QUESTAO: tipo da questão. Será utilizado pela aplicação para definir se a questão apresenta uma combo de seleção, ou um campo texto, dentre outras opções;
- d. TP_QUESTIONARIO: tipo de questionário que engloba a questão.

3.1.8 Tabela CS_BI_QUESTIONARIOS

Tabela que possui informações a respeito do questionário. Ela é a interface de um novo questionário/formulário.

- a. CD_QUESTIONARIO: chave primária;
- b. NM_QUESTIONARIO: título do questionário (Berlin, Atenas, etc);
- c. DS_QUESTIONARIO: breve descrição a respeito do questionário; do que ele trata; é utilizado para que finalidade; etc.

3.1.9 Tabela CS_BI_EXAME_LABO

Tabela que possui informações a respeito de exames laboratoriais.

- a. CD_EXAME: chave primária;
- b. DS_EXAME: descrição do exame;
- c. ST_EXAME: status do exame; se está ativo ou não.

3.1.10 Tabela CS_BI_CONVENIOS

Tabela que possui informações a respeito dos convênios que os pacientes apresentam.

- a. CD_CONVENIO: chave primária;
- b. NM_CONVENIO: nome do convênio;
- c. DS_CONVENIO: descrição a respeito do convênio;
- d. TP_CONVENIO: tipo de convênio (público ou privado);

3.1.11 Tabela CS_BI_DADOS_CLIN

Tabela que possui informações a respeito dos diversos dados clínicos (peso, altura, medidas, etc).

- a. CD_DADO_CLINICO: chave primária;
- b. NM_DADO_CLINICO: título do dado clínico;
- c. DS_DADO_CLINICO: descrição do dado clínico.

3.1.12 Tabela CS_BI_CONSULTAS_PAC

Tabela que possui informações a respeito de consultas dos pacientes.

- a. CS_BI_IDENTIFICACAO_CD_PACIENTE: chave estrangeira;
- b. DT_CONSULTA: data da consulta;
- c. DS_OBSERVACAO: observações realizadas a respeito da consulta.

3.1.13 Tabela CS_BI_USUARIO_PERMISSAO

Tabela que possui informações a respeito das permissões dos usuários perante a aplicação web.

- a. CS_BI_USUARIO_CD_USUARIO: chave estrangeira;
- b. CD_PERMISSAO: código da permissão concedida ao usuário.

3.1.14 Tabela CS_BI_ALTERNATIVAS

Tabela que possui as alternativas vinculadas a cada questão.

- a. CD_ALTERNATIVA: chave primária;
- b. CS_BI_QUESTOES_CD_QUESTAO: chave estrangeira;
- c. DS_ALTERNATIVA: descrição da alternativa.

3.1.15 Tabela CS_BI_RESPOSTAS_PAC

Tabela que possui todas as respostas dos pacientes aos diversos questionário/formulários relacionados a clínica.

- a. CS_BI_IDENTIFICACAO_CD_PACIENTE: chave estrangeira;
- b. DS_OBSERVACOES: possíveis observações a serem adicionadas nas respostas;
- c. CD_ALTERNATIVA: código da alternativa selecionada;
- d. CD_QUESTAO: código da questão que está sendo respondida.

3.1.16 Tabela CS_BI_DIAGNOSTICOS

Tabela que possui os diagnósticos de cada paciente.

- a. CD_DIAGNOSTICO: chave primária;
- b. CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER: chave estrangeira;
- c. NM_DIAGNOSTICO: nome do diagnóstico;
- d. DS_DIAGNOSTICO: descrição do diagnóstico.

3.1.17 Tabela CS_BI_FINANCEIRO

Tabela que possui o controle financeiro dos pacientes.

- a. CS_BI_IDENTIFICACAO_CD_PACIENTE: chave estrangeira;
- b. CS_BI_CONVENIOS_CD_CONVENIO: chave estrangeira;
- c. DT_FATURAMENTO: data do faturamento;
- d. VL_COBRANCA: valor.

3.1.18 Tabela CS_BI_EXAME_LABO_PAC

Tabela que possui o vínculo de um paciente a um determinado exame.

- a. CS_BI_IDENTIFICACAO_CD_PACIENTE: chave estrangeira;
- b. CS_BI_EXAME_LABO_CD_EXAME: chave estrangeira;
- c. DS_OBSERVACAO: possíveis observações a respeito do exame.

3.1.19 Tabela CS_BI_DADOS_CLIN_PAC

Tabela que possui o vínculo de um paciente a um conjunto de dados clínicos.

- a. CS_BI_IDENTIFICACAO_CD_PACIENTE: chave estrangeira;
- b. CS_BI_DADOS_CLIN_CD_DADO_CLINICO: chave estrangeira;
- c. DS_VALOR: valor do dado clínico (por exemplo: 80 → corresponde ao peso em kg do paciente, para o dado clínico peso).

3.1.20 Tabela CS_BI_DADOS_CLINICOS_DIAG

Tabela que possui o conjunto de diagnósticos obtidos para um paciente durante um determinado tratamento. Esta tabela será usada durante o tratamento do paciente para que o responsável pelo tratamento possua um histórico da evolução do tratamento com o paciente. É uma espécie de um prontuário.

- a. CS_BI_IDENTIFICACAO_CD_PACIENTE: chave estrangeira;

- b. CS_BI_MEDICOS_INDICACOES_CD_MEDICO: médico que indicou paciente;
- c. CD_DIAG1: código do diagnóstico 1;
- d. CD_DIAG2: código do diagnóstico 2;
- e. CD_DIAG3: código do diagnóstico 3;
- f. CD_DIAG_DEF1: código do diagnóstico definitivo 1;
- g. CD_DIAG_DEF2: código do diagnóstico definitivo 2;
- h. CD_DIAG_DEF3: código do diagnóstico definitivo 3;
- i. HDA;
- j. DT_EXAME_CLINICO: data do exame clínico.

3.1.21 Tabela CS_BI_MEDICAMENTOS_PAC

Tabela que possui o vínculo de um paciente aos medicamentos consumidos pelo mesmo.

- a. CS_BI_IDENTIFICACAO_CD_PACIENTE: chave estrangeira;
- b. CS_BI_MEDICAMENTOS_CD_MEDICAMENTO: chave estrangeira;
- c. DS_OBSERVACAO: possíveis observações sobre o medicamento;
- d. QT_DOSAGEM: dosagem consumida pelo paciente.

3.1.22 Tabela CS_BI_QUESTIONARIO_QUESTOES

Tabela que possui o vínculo de um questionário ao conjunto de questões que compõe o mesmo.

- a. CS_BI_QUESTIONARIOS_CD_QUESTIONARIO: chave estrangeira;
- b. CS_BI_QUESTOES_CD_QUESTAO: chave estrangeira.

Com posse das informações apresentadas acima é que foi gerado o script para a geração das tabelas. O software DBDesigner apresenta um facilitador que, a partir do momento que se apresenta a base definida, pode-se exportar para um arquivo de extensão .txt os scripts de criação das tabelas. Este, pode ser verificado no anexo 1, presente neste trabalho.

3.2 Análise dos dados e cargas na nova base

Para a implantação da nova base, após a criação das tabelas, foram executadas diversas consultas na base existente em Access.

Conforme a necessidade que uma tabela apresentava no novo modelo é que se executava uma determinada consulta. A seguir encontram-se listadas as consultas e uma prévia dos dados retornados que foram utilizados para as cargas na nova base. Também será apresentada a forma com que as cargas foram executadas.

Devido ao fato da ausência de uma ferramenta eficiente e principalmente ao fato de que a base presente na plataforma Access não apresentar uma modelagem reaproveitável do ponto de vista relacional é que se foi utilizada uma metodologia de migração de dados bastante manual.

Para a migração dos dados, foram executados os passos a serem descritos nos próximos sub-tópicos. Neste trabalho não serão descritos todas as migrações pois este ficaria extremamente extenso. Logo, serão apresentados com detalhes todos os passos que são adotados em um determinado caso. Como exemplo, será utilizada a migração dos dados presentes principalmente na tabela IDENTIFICAÇÃO 2006.

Como será possível perceber, os dados presentes na tabela citada no parágrafo anterior irão compor mais de uma tabela, dependendo do tipo de informação que o dado fornece.

A seguir os passos executados na migração dos dados.

3.2.1 Seleção na base Access das informações necessárias

A seleção dos dados na base em Access foram realizadas com o auxílio da interface de design de consulta e posteriormente realizando uma verificação na consulta gerada, a fim de se certificar de que a consulta realmente condiz com a real necessidade.

Um exemplo de consulta realizada nesta tabela, utilizando o design, pode ser verificado na imagem a seguir.

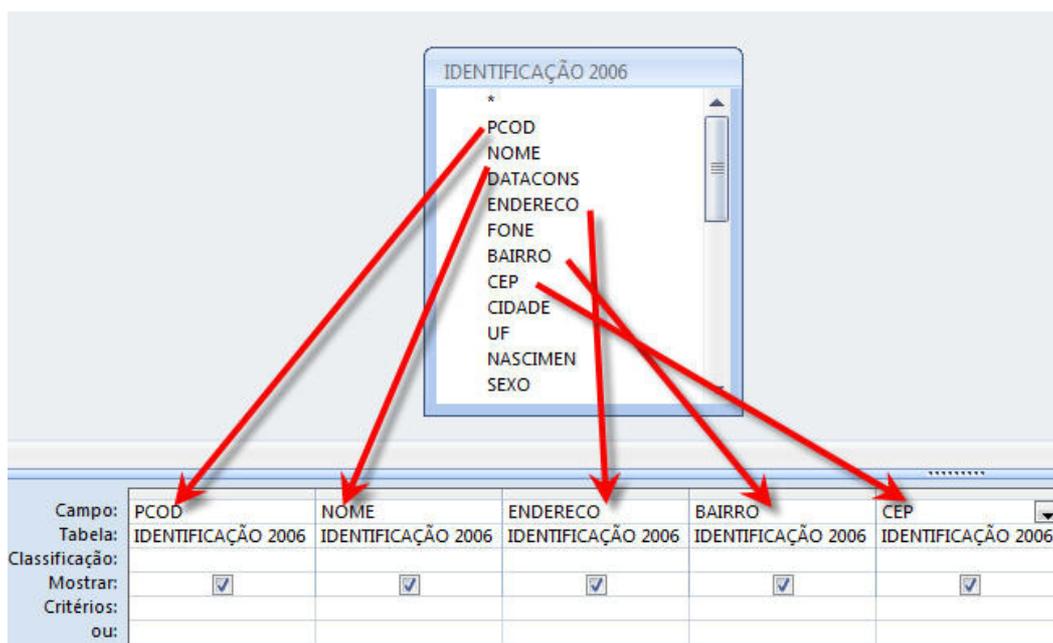


Figura 3.2 - Consulta realizada no modo design do Access

Esta consulta, conseqüentemente gera o seguinte script SQL:

```
SELECT [IDENTIFICAÇÃO 2006] PCOD,
       [IDENTIFICAÇÃO 2006].NOME,
       [IDENTIFICAÇÃO 2006].ENDERECO,
       [IDENTIFICAÇÃO 2006].BAIRRO,
       [IDENTIFICAÇÃO 2006].CEP
FROM [IDENTIFICAÇÃO 2006];
```

O resultado da execução desta consulta retorna os seguintes dados (modificados para proteger a identidade dos pacientes):

Consulta				
PCOD	NOME	ENDERECO	BAIRRO	CEP
2	JOAO DA SILVA	FLORENCIO YGARTUA, 10	NONOAI	90000
3	CARLOS DA SILVA	PROTASIO ALVES 89		90000
4	PAULO DA SILVA	RUA SANTOS DUMONT, 11		90000
5	PEDRO DA SILVA	RUA SÃO LEOPOLDO, 33	MATHIAS VELHO	
6	GIOVANI DA SILVA	CARLOS P. DE BRUM 1	SANTO ANTONIO	90000
7	JULIANA DA SILVA	ACORES 53		
8	MARCOS SILVA JUNIOR	15 DE NOVEMBRO 66		
9	MARCO JOÃO	LILA RIPOL 30		
10	ADRIANA DA SILVA	BENTO GONCALVES 363		90000
11	CAROLINA BERJÓ	ASSIS BRASIL 4		
12	CARINE DA SILVA	FRANCISCO JOSE SOARES 16		
13	JAMILE SILVA SÓ	PROF IVO CORSEIL 42 /10		90000
14	MARIANA SILVA	RUA BARAO DO AMAZONAS, 4 AP. 1		90000

Tabela 3.1 - Exemplo de resultset da consulta a tabela de identificação.

Para se garantir que a vinculação de todos os dados presentes nas tabelas da base que existe no Access fossem corretamente vinculados aos seus respectivos pacientes, a chave PCOD (presente no Access) e CD_PACIENTE (presente no MySQL) foi mantida. A diferença existe no fato de que, no momento da migração dos dados, foi garantida a unicidade deste código, setando na tabela de identificação do MySQL a propriedade de unicidade deste atributo.

3.2.2 Elaboração dos scripts de carga

Com posse do resultado da consulta, foi elaborada uma planilha em MS-Excel para que se realizassem as cargas na base em MySQL. Um exemplo de elaboração dos scripts de carga podem ser percebidos na imagem a seguir:

		PCOD	NOME	END	NUM	COMP
2	INSERT INTO CS_IDENTIFICACAO (CD_PACIENTE, NM_PACIENTE, DS_ENDERECO, DS_BAIRRO, DS_CEP) VALUES (TRIM(10000), TRIM(''), TRIM(''), TRIM(''), TRIM('')) ;	10000				
3	INSERT INTO CS_IDENTIFICACAO (CD_PACIENTE, NM_PACIENTE, DS_ENDERECO, DS_BAIRRO, DS_CEP) VALUES (TRIM(10001), TRIM(''), TRIM(''), TRIM(''), TRIM('')) ;	10001				
4	INSERT INTO CS_IDENTIFICACAO (CD_PACIENTE, NM_PACIENTE, DS_ENDERECO, DS_BAIRRO, DS_CEP) VALUES (TRIM(10002), TRIM(''), TRIM(''), TRIM(''), TRIM('')) ;	10002				
5	INSERT INTO CS_IDENTIFICACAO (CD_PACIENTE, NM_PACIENTE, DS_ENDERECO, DS_BAIRRO, DS_CEP) VALUES (TRIM(10003), TRIM(''), TRIM(''), TRIM(''), TRIM('')) ;	10003				
6	INSERT INTO CS_IDENTIFICACAO (CD_PACIENTE, NM_PACIENTE, DS_ENDERECO, DS_BAIRRO, DS_CEP) VALUES (TRIM(10004), TRIM(''), TRIM(''), TRIM(''), TRIM('')) ;	10004				
7	INSERT INTO CS_IDENTIFICACAO (CD_PACIENTE, NM_PACIENTE, DS_ENDERECO, DS_BAIRRO, DS_CEP) VALUES (TRIM(10005), TRIM(''), TRIM(''), TRIM(''), TRIM('')) ;	10005				
8	INSERT INTO CS_IDENTIFICACAO (CD_PACIENTE, NM_PACIENTE, DS_ENDERECO, DS_BAIRRO, DS_CEP) VALUES (TRIM(10006), TRIM(''), TRIM(''), TRIM(''), TRIM('')) ;	10006				

Figura 3.3 - Elaboração no MS-Excel dos scripts de cargas.

Após o procedimento de retirada dos dados do Access e lançamento na planilha, o script foi então exportado para um arquivo .sql, resultando no seguinte conjunto de comandos SQL:

```

/*
DE-PARA ACCESS.IDENTIFICACAO 2006 --> MYSQL IDENTIFICACAO
CAMPOS:
    CD_PACIENTE,
    NM_PACIENTE,
    DS_ENDERECO,
    DS_BAIRRO,
    DS_CEP
*/
cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep)
/* COD 1 ATÉ 4999 */

INSERT INTO cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep) VALUES (TRIM( 2
),TRIM(' TESTE1 '), TRIM(' FLORENCIO YGARTUA, XXXAP. XXX '), TRIM(' NONOAI '), TRIM(' 90000
')) ;

INSERT INTO cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep) VALUES (TRIM( 3
),TRIM(' TESTE2 '), TRIM(' PROTASIO ALVES XXX '), TRIM(' '), TRIM(' 90000 '))
;

INSERT INTO cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep) VALUES (TRIM( 4
),TRIM(' TESTE3 '), TRIM(' RUA SANTOS DUMONT, XXX'), TRIM(' '), TRIM(' 90000 '))
;

INSERT INTO cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep) VALUES (TRIM( 5
),TRIM(' TESTE4 '), TRIM(' RUA SÃO LEOPOLDO, XXX'), TRIM(' MATHIAS VELHO '), TRIM('
')) ;

INSERT INTO cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep) VALUES (TRIM( 6
),TRIM(' TESTE5 '), TRIM(' CARLOS P. DE BRUM XXX'), TRIM(' SANTO ANTONIO '), TRIM(' 90000
')) ;

INSERT INTO cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep) VALUES (TRIM( 7
),TRIM(' TESTE6 '), TRIM(' ACORES XXX '), TRIM(' '), TRIM(' '))
;

INSERT INTO cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep) VALUES (TRIM( 8
),TRIM(' TESTE7 '), TRIM(' 15 DE NOVEMBRO XXX '), TRIM(' '), TRIM(' '))
;

INSERT INTO cs_bi_identificacao (cd_paciente, nm_paciente, ds_endereco, ds_bairro, ds_cep) VALUES (TRIM( 9
),TRIM(' TESTE8 '), TRIM(' LILA RIPOL XXX '), TRIM(' '), TRIM(' '))
;

```

3.2.3 Inserção na base MySQL

Tendo em mãos o arquivo de script SQL, foi realizada a carga na base MySQL. Principalmente para o caso da tabela de identificação, por possuir aproximadamente 15000 registros, as cargas foram realizadas em blocos de 5000 registros para que a

execução fosse mais ágil e que a correção de eventuais inconsistências fossem facilmente corrigidas.

Primeiramente foi realizada a carga para uma base local de testes. Após a carga ser realizada com sucesso, o script de inserção de dados foi devidamente corrigido e então foi realizada a gravação dos dados na tabela presente no servidor da clínica.

Como a tabela de identificação possui muitas colunas, como ocorre também em outras tabelas da base Access, então as cargas foram realizadas aos poucos, selecionando algumas colunas, também pela razão de facilitar eventuais correções e com isso agilizar o processo.

Cada script executado foi salvo com o intuito de possuir um backup das cargas iniciais da base na plataforma MySQL.

Após a gravação realizada com sucesso, podemos verificar os dados carregados na base nova do MySQL. A seguir uma imagem apresentando um resultset da tabela CS_BI_IDENTIFICACAO após a carga dos dados:

CD_PACIENTE	NM_PACIENTE	DT_NASCIMENTO	DS_IDENTIDADE	DS_ENDERECO
2	TESTE 1	10/10/1990	76857665476	
3	TESTE2	Null	Null	Null

Figura 3.4 - Resultset de uma consulta na base MySQL.

4 LEVANTAMENTOS DE DADOS ESTATÍSTICOS

Após a migração dos dados, iniciou-se um processo de elaboração de um conjunto de consultas que o usuário pode utilizar para a obtenção de dados estatísticos. Esta etapa visa principalmente facilitar ao usuário o levantamento de dados que o mesmo considerar relevantes para alterações em metodologias de exames, na utilização de questionários, bem como em decisões a serem tomadas para uma melhor administração da clínica.

O ponto principal desta etapa é propiciar ao usuário a possibilidade de traçar um determinado perfil para cada paciente. A partir deste perfil traçado, o usuário poderá elaborar um conjunto de questionários e exames diferentes dos que existem, ou então alterar os que já se encontram, a fim de melhorar o atendimento e a eficiência dos exames, avaliações, diagnósticos e questionários hoje existentes.

Além disso, como a clínica é utilizada por diversos bolsistas todos os anos, este capítulo visa auxiliar os mesmos na obtenção de dados estatísticos de uma maneira mais rápida e fácil. Além disso, a modelagem nova dos dados está mais flexível à alterações, o que faz com que as consultas abranjam todos os casos possíveis, sem muito impacto.

A apresentação de todas as possibilidades de consultas para a obtenção de dados faria com que este trabalho ficasse extremamente extenso. Por isso que, neste caso, será apresentado um exemplo de implementação e como será repassado ao usuário a manipulação dos dados, para que ele possa obter estes dados enquanto não se implementa na aplicação um módulo para a obtenção destes dados.

Utilizando a tabela de identificação, de questões, de alternativas e de respostas dos pacientes, vamos verificar quantos pacientes roncam, com qual intensidade e se o ganho de peso apresenta alguma relação com esse dado. Logo, a partir das informações obtidas, podemos obter uma relação entre o ganho de peso e a questão de roncar ou não.

Primeiro apresentar-se-ão 4 tabelas envolvidas na pesquisa:

Tabela CS_BI_IDENTIFICACAO

CS_BI_IDENTIFICACAO	
PK	CD_PACIENTE: INTEGER
	NM_PACIENTE: VARCHAR(60)
	DT_NASCIMENTO: DATE
	DS_IDENTIDADE: VARCHAR(30)
	DS_ENDERECO: TEXT
	DS_BAIRRO: VARCHAR(20)
	DS_CEP: VARCHAR(15)
	CD_CIDADE: VARCHAR(5)
	CD_PAIS: VARCHAR(5)
	DS_SEXO: VARCHAR(10)
	CD_ESTADO: VARCHAR(5)
	DS_FONE_COM: VARCHAR(15)
	DS_FONE_CEL: VARCHAR(15)
	CD_COR: INTEGER
	DS_EMAIL: VARCHAR(20)
	DS_IDADE: INTEGER
	NM_ACOMPANHANTE: VARCHAR(60)
	DS_NATURALIDADE: VARCHAR(60)

Figura 4.1 - Tabela de identificação.

Tabela CS_BI_QUESTOES

CS_BI_QUESTOES	
PK	CD_QUESTAO: INTEGER
	DS_QUESTAO: VARCHAR(100)
	TP_QUESTAO: CHAR(1)
	TP_QUESTIONARIO: CHAR(2)

Figura 4.2 - Tabela de questões.

Tabela CS_BI_ALTERNATIVAS

CS_BI_ALTERNATIVAS	
PK	CD_ALTERNATIVA: INTEGER
	CS_BI_QUESTOES_CD_QUESTAO: INTEGER (FK)
	DS_ALTERNATIVA: VARCHAR(100)

Figura 4.3 - Tabela de alternativas.

Tabela CS_BI_RESPOSTAS_PAC

CS_BI_RESPOSTAS_PAC	
	CS_BI_IDENTIFICACAO_CD_PACIENTE: INTEGER (FK)
	DS_OBSERVACOES: TEXT
	CD_ALTERNATIVA: INTEGER
	CD_QUESTAO: INTEGER

Figura 4.4 - Tabela de respostas.

Verificando a tabela de questões, através da consulta abaixo, obtém-se o seguinte resultset e consequentemente percebe-se que a questão sobre o ronco é a de código 10.

```
select *
from `cs_bi_questoes`
order by cd_questao
```

CD_QUESTAO	DS_QUESTAO	TP_QUESTAO	TP_QUESTIONARIO
1	Vendo TV	C	C
2	Lendo	C	C
3	Aulas	C	C
4	Viajando	C	C
5	a Tarde	C	C
6	Conversando	C	C
7	Após almoço	C	C
8	no Trânsito	C	C
9	Seu peso mudou?	C	C
10	Você ronca?	C	C
11	Qual a intensidade do seu ronco?	T	T
12	Qual frequência do seu ronco?	T	T
13	Seu ronco perturba?	C	C

Figura 4.5 - Resultset da tabela de questões.

A partir da obtenção do número da questão, executa-se a consulta para descobrir o valor da alternativa que corresponde à opção Sim. A consulta executada é a que segue.

```
select *
from `cs_bi_alternativas`
where cs_bi_alternativas.`CS_BI_QUESTOES_CD_QUESTAO` = 10
order by cd_alternativa
```

Esta consulta apresenta o seguinte resultset:

CD_ALTERNATIVA	CS_BI_QUESTOES_CD_QUESTAO	DS_ALTERNATIVA
E		10 Não sabe.
N		10 Não.
S		10 Sim.

Figura 4.6 - Resultset da tabela de alternativas – questão 10.

Queremos obter as pessoas que responderam a opção ‘S’.

Para isto, deve-se buscar na tabela de respostas dos pacientes, os que responderam ‘S’ para a questão de código 10. Porém, antes disso, devemos realizar as duas primeiras consultas novamente para descobrir quais valores serão necessários para buscarmos pelos pacientes que apresentaram ganho de peso. As consultas são exatamente as

mesmas, porém alteramos os parâmetros. Com isso, obtivemos as duas consultas abaixo:

```
select *
  from `cs_bi_questoes`
 order by cd_questao
```

```
select *
  from `cs_bi_alternativas`
 where cs_bi_alternativas.`CS_BI_QUESTOES_CD_QUESTAO` = 9
 order by cd_alternativa
```

Com isto, percebe-se que a alternativa que buscamos é a que possui o código '1', conforme imagem abaixo:

CD_ALTERNATIVA	CS_BI_QUESTOES_CD_QUESTAO	DS_ALTERNATIVA
1	9	Sim, aumentou.
2	9	Sim, diminuiu.
3	9	Não.

Figura 4.7 - Resultset da tabela de alternativas – questão 9.

Fazem-se algumas junções entre as tabelas para apresentarmos os nomes dos pacientes. A consulta resultante da pesquisa é a que segue abaixo. Pode-se perceber que é realizada uma contagem abrangendo os 3 tipos de respostas, para que se possa traçar uma porcentagem.

```
select c.`DS_ALTERNATIVA`,
       count(*)
  from `cs_bi_respostas_pac` as a,
       `cs_bi_questoes` as b,
       `cs_bi_alternativas` as c,
       `cs_bi_respostas_pac` as d,
       `cs_bi_questoes` as e,
       `cs_bi_alternativas` as f
 where b.`CD_QUESTAO`=c.`CS_BI_QUESTOES_CD_QUESTAO`
 and a.`CD_QUESTAO`=b.cd_questao
 and a.CD_ALTERNATIVA=c.CD_ALTERNATIVA
```

```

and e.`CD_QUESTAO`=f.`CS_BI_QUESTOES_CD_QUESTAO`
and d.`CD_QUESTAO`=e.cd_questao
and d.CD_ALTERNATIVA=f.CD_ALTERNATIVA
and          a.`CS_BI_IDENTIFICACAO_CD_PACIENTE`          =
d.`CS_BI_IDENTIFICACAO_CD_PACIENTE`
and e.cd_questao=9 -- questao é sobre peso
and f.`CD_ALTERNATIVA` = 1 --peso aumentou
and b.cd_questao=10 -- questao é sobre ronco
group by c.`DS_ALTERNATIVA`

```

Esta consulta conta o número de pacientes que relataram que o peso aumentou e apresenta as três possibilidades de ronco (sim, não e não sabe). Abaixo encontra-se o *resultset* desta consulta.

DS_ALTERNATIVA	count(*)
Não sabe.	14
Não.	104
Sim.	1635

Figura 4.8 - Resultset da consulta de contagem.

Pode-se verificar que 1635 pacientes alegam que o peso aumentou e que roncam.

Agora, realizar-se-á outra consulta para ver o total de pacientes que roncam. A consulta a seguir é executada para obter tal informação:

```

select count(*)
from cs_bi_respostas_pac
where cd_questao = 10
and CD_ALTERNATIVA='S'

```

O resultado da consulta retorna o número de 2462.

Logo, pode-se concluir que mais de 50% dos pacientes que relatam que aumentaram de peso roncam.

De repente, do ponto de vista da clínica, este levantamento é irrelevante. Porém, este exemplo apresenta o poder de obter dados de diversas formas e também apresenta a flexibilidade da base após a reestruturação, possibilitando que o usuário cruze diversos dados de várias maneiras.

5 CONCLUSÕES

Com este trabalho realizou-se a reengenharia de uma base de dados existente em uma clínica médica, a Clínica do Sono. A base que existia encontrava-se com diversos problemas. Problemas que foram acumulando a medida que a demanda na clínica aumentava, juntamente com a presença de novos casos, dentre outros fatores.

A partir da remodelagem da base, foi possível tornar a organização dos dados de uma maneira mais abrangente, fazendo com que o impacto perante a base de dados seja mínimo, quando acontecerem alterações nas metodologias da clínica como em exames, questionários, entre outros. Também, este novo modelo visa atender a aplicação web que foi desenvolvida para a geração de formulários dinâmicos, o que facilita e muito a administração dos questionários utilizados, que são as maiores prioridades do usuário.

A apresentação de dados estatísticos visa atender ao usuário, em um primeiro instante, na obtenção de diversa informações, com cruzamentos de dados de diversas maneiras, para que não somente o Dr. Dênis, mas também seus estagiários obtenham todo o tipo de informação que necessitem para fundo de pesquisas ou qualquer que seja a utilização dos dados. Além deste trabalho, foi elaborado um documento que será repassado ao Dr. Dênis para que o mesmo possa utilizar por enquanto, na obtenção desses dados.

Encontrou-se no desenvolvimento deste trabalho um pouco de dificuldade na visualização das entidades e dos relacionamentos presentes na base atual, devido aos termos específicos utilizados em alguns atributos e entidades. Também a redundância dos dados e a difícil localização de alguns atributos contribuíram para dificultar o entendimento das entidades e relacionamentos presentes. Porém, apesar das dificuldades apresentadas, atingiu-se o principal objetivo da modelagem e implantação, que visa por ter uma base o mais genérica possível para que, no futuro, a manutenção da base seja mais fácil e ágil.

Com este trabalho de conclusão de curso, foi possível perceber as dificuldades na extração de informações para com o cliente devido às diferenças de linguagens utilizadas. Também, o grande desafio neste trabalho foi a identificação de problemas e a reestruturação da base de dados, sem que ocorresse uma descaracterização dos objetivos descritos pelo usuário.

Por fim, a modelagem nova da base apresenta uma possibilidade de se desenvolver uma ferramenta de BI que facilite a extração de dados a fim de auxiliar o usuário na tomada de decisões para com a clínica. Ao final da elaboração desta ferramenta, o usuário não necessitará mais das consultas fornecidas previamente, visto que a ferramenta de BI será responsável pela abstração destes dados.

REFERÊNCIAS

Silberschatz, Korth e Sudarshan, **Sistemas de Bancos de Dados**, 5^ª edição, Campus, 2006.

Heuser, C.A. **Projeto de Banco de Dados**. 6^a edição. Ed. Bookman, Porto Alegre, 2009

Macoratti, José Carlos, Conceitos básicos de modelagem de dados. Disponível em <http://www.macoratti.net/cbmd1.htm>.

Wikipedia, Business Intelligence. Disponível em http://pt.wikipedia.org/wiki/Business_intelligence.

GLOSSÁRIO

CONCEITOS RELACIONADOS

Os conceitos relacionados nesse trabalho encontram-se listados a seguir:

1. BI: proveniente de *Business Intelligence*. Engloba o uso de ferramentas sofisticadas, que proporcionam informações mais acuradas, base de conhecimento, além de relatórios diversificados.
2. SGBD: sistema de gerenciamento de banco de dados. Representa a sigla do sistema, que fornece ao desenvolvedor, ferramentas que facilitam no gerenciamento de uma base de dados, entre outras funcionalidades.
3. MS-Access: corresponde ao SGBD da Microsoft. A base de dados atual da clínica encontra-se no formato do MS-Access (.mdb).
4. MySQL Server 5.0: SGBD livre. Neste será criada a nova base de dados, pois o mesmo permite conexão multiusuário, facilidade para implementação de aplicações web relacionadas à ele, entre outros fatores.
5. DBDesigner 4: ferramenta para a modelagem da base de dados. Possui uma interface amigável, que apresenta as entidades, atributos e relacionamentos da base. Permite também a geração de scripts na linguagem SQL que facilitam na criação da base de dados.
6. SQL Manager Lite: ferramenta para a manipulação do banco de dados. Permite a execução dos scripts SQL para a criação e manipulação do banco e dos dados. Também facilita a definição de permissões de acesso às tabelas e ao banco, criação e configuração de usuários que irão acessar a base, entre outros.
7. Entidade: trata-se do objeto ou evento do mundo real, distintamente identificado e tratado como uma categoria definida, sobre a qual armazenam-se os dados. Podem ser do tipo fundamental, associativa ou derivada de relacionamento, atributiva, fortes e fracas;
8. Atributos: são dados elementares que permitem descrever a entidade ou relacionamento. Podem ser monovalorados ou multivalorados;
9. Domínio: trata-se do conjunto de valores possíveis do atributo;

10. Tupla: é o elemento do conjunto de uma entidade. Corresponde a uma estrutura de atributos intimamente relacionados e interdependentes que residem em uma entidade específica;
11. Chave ou identificador: é o atributo contido na tupla que a personaliza e individualiza;
12. Chave primária: atributo ou agrupamento de atributos cujo valor identifica unicamente uma tupla dentre todas as outras de uma identidade. Deve ter conteúdo reduzido e valor constante no tempo;
13. Chave candidata: atributo ou agrupamento de atributos que possui a propriedade de identificação única;
14. Chave alternativa: trata-se da chave candidato que não é chave primária;
15. Chave estrangeira: quando um atributo de uma entidade é a chave primária de outra;
16. Chave composta ou concatenada: é formada pelo agrupamento de mais de um atributo;
17. Chave secundária: é o atributo que não possui a propriedade de identificação única;
18. Relacionamento: associação, com significado, de duas ou mais entidades;
19. Cardinalidade máxima: quantidade máxima de ocorrências de entidades que podem estar associadas a uma ocorrência de outra entidade;
20. Cardinalidade mínima: indica se a participação das ocorrências de entidades no relacionamento é obrigatória ou opcional;
21. Auto-relacionamento: representa associação entre ocorrências de uma mesma entidade;
22. Relacionamento “N”-ário: abstração de uma associação entre “N” entidades;

Além dos conceitos apresentados acima, pode ocorrer de existirem conceitos novos, que serão definidos no decorrer das demais etapas do trabalho.

ANEXO A - SCRIPTS DE CRIAÇÃO DAS TABELAS NOVAS

A seguir, apresentam-se os scripts gerados em SQL para a criação da nova base de dados e das tabelas com suas respectivas propriedades.

```
CREATE TABLE CS_BI_MEDICOS_INDICACOES (  
  CD_MEDICO INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  NM_MEDICO VARCHAR(90) NULL,  
  QT_INDICACOES INTEGER UNSIGNED NULL,  
  DS_ENDERECO TEXT NULL,  
  DS_BAIRRO VARCHAR(30) NULL,  
  CD_CIDADE INTEGER UNSIGNED NULL,  
  CD_UF INTEGER UNSIGNED NULL,  
  CD_PAIS INTEGER UNSIGNED NULL,  
  CD_CEP INTEGER UNSIGNED NULL,  
  DS_FONE VARCHAR(30) NULL,  
  PRIMARY KEY(CD_MEDICO)  
);
```

```
CREATE TABLE CS_BI_MEDICAMENTOS (  
  CD_MEDICAMENTO INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  NM_MEDICAMENTO VARCHAR(100) NULL,  
  DS_MEDICAMENTO TEXT NULL,  
  DS_INDICACOES TEXT NULL,  
  DS_CONTRA_INDICACOES TEXT NULL,  
  DS_OBSERVACOES TEXT NULL,  
  PRIMARY KEY(CD_MEDICAMENTO)  
);
```

```
CREATE TABLE CS_BI_IDENTIFICACAO (  
  CD_PACIENTE INTEGER UNSIGNED NOT NULL,
```

```
NM_PACIENTE VARCHAR(60) NULL,  
DT_NASCIMENTO DATE NULL,  
DS_IDENTIDADE VARCHAR(30) NULL,  
DS_ENDERECO TEXT NULL,  
DS_BAIRRO VARCHAR(20) NULL,  
DS_CEP VARCHAR(15) NULL,  
CD_CIDADE VARCHAR(5) NULL,  
CD_PAIS VARCHAR(5) NULL,  
DS_SEXO VARCHAR(10) NULL,  
CD_ESTADO VARCHAR(5) NULL,  
DS_FONE_COM VARCHAR(15) NULL,  
DS_FONE_CEL VARCHAR(15) NULL,  
CD_COR INTEGER UNSIGNED NULL,  
DS_EMAIL VARCHAR(20) NULL,  
DS_IDADE INTEGER NULL,  
NM_ACOMPANHANTE VARCHAR(60) NULL,  
DS_NATURALIDADE VARCHAR(60) NULL,  
PRIMARY KEY(CD_PACIENTE)  
);  
  
CREATE TABLE CS_BI_PARAMETROS (  
  CD_PARAMETRO INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  TP_PARAMETRO VARCHAR(5) NULL,  
  DS_PARAMETRO VARCHAR(20) NULL,  
  PRIMARY KEY(CD_PARAMETRO)  
);  
  
CREATE TABLE CS_BI_USUARIO (  
  CD_USUARIO VARCHAR(40) NOT NULL AUTO_INCREMENT,  
  NM_USUARIO VARCHAR(100) NULL,  
  TP_USUARIO CHAR(1) NULL,  
  DS_EMAIL VARCHAR(60) NULL,  
  DS_SENHA VARCHAR(60) NOT NULL,  
  PRIMARY KEY(CD_USUARIO)  
);  
  
CREATE TABLE CS_BI_QUESTOES (  
  CD_QUESTAO INTEGER NOT NULL AUTO_INCREMENT,  
  DS_QUESTAO VARCHAR(100) NULL,
```

```
TP_QUESTAO CHAR(1) NULL,  
TP_QUESTIONARIO CHAR(2) NULL,  
PRIMARY KEY(CD_QUESTAO)  
);
```

```
CREATE TABLE CS_BI_QUESTIONARIOS (  
  CD_QUESTIONARIO INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  NM_QUESTIONARIO VARCHAR(100) NULL,  
  DS_QUESTIONARIO TEXT NULL,  
  PRIMARY KEY(CD_QUESTIONARIO)  
);
```

```
CREATE TABLE CS_BI_EXAME_LABO (  
  CD_EXAME INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  DS_EXAME VARCHAR(60) NULL,  
  ST_EXAME CHAR NULL,  
  PRIMARY KEY(CD_EXAME)  
);
```

```
CREATE TABLE CS_BI_CONVENIOS (  
  CD_CONVENIO INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  NM_CONVENIO VARCHAR(20) NULL,  
  DS_CONVENIO TEXT NULL,  
  TP_CONVENIO INTEGER UNSIGNED NULL,  
  PRIMARY KEY(CD_CONVENIO)  
);
```

```
CREATE TABLE CS_BI_DADOS_CLIN (  
  CD_DADO_CLINICO INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  NM_DADO_CLINICO VARCHAR(50) NULL,  
  DS_DADO_CLINICO TEXT NULL,  
  PRIMARY KEY(CD_DADO_CLINICO)  
);
```

```
CREATE TABLE CS_BI_CONSULTAS_PAC (  
  CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER UNSIGNED NOT NULL,  
  DT_CONSULTA DATE NOT NULL AUTO_INCREMENT,  
  DS_OBSERVACAO TEXT NULL,  
  PRIMARY KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE),
```

```

INDEX CS_CONSULTAS_PAC_FKIndex1(CS_BI_IDENTIFICACAO_CD_PACIENTE),
FOREIGN KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE)
REFERENCES CS_BI_IDENTIFICACAO(CD_PACIENTE)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_USUARIO_PERMISSAO (
CS_BI_USUARIO_CD_USUARIO VARCHAR(40) NOT NULL,
CD_PERMISSAO INTEGER NULL,
PRIMARY KEY(CS_BI_USUARIO_CD_USUARIO),
INDEX CS_USUARIO_PERMISSAO_FKIndex1(CS_BI_USUARIO_CD_USUARIO),
FOREIGN KEY(CS_BI_USUARIO_CD_USUARIO)
REFERENCES CS_BI_USUARIO(CD_USUARIO)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_ALTERNATIVAS (
CD_ALTERNATIVA INTEGER NOT NULL AUTO_INCREMENT,
CS_BI_QUESTOES_CD_QUESTAO INTEGER NOT NULL,
DS_ALTERNATIVA VARCHAR(100) NULL,
PRIMARY KEY(CD_ALTERNATIVA, CS_BI_QUESTOES_CD_QUESTAO),
INDEX CS_ALTERNATIVAS_FKIndex1(CS_BI_QUESTOES_CD_QUESTAO),
FOREIGN KEY(CS_BI_QUESTOES_CD_QUESTAO)
REFERENCES CS_BI_QUESTOES(CD_QUESTAO)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_RESPOSTAS_PAC (
CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER UNSIGNED NOT NULL,
DS_OBSERVACOES TEXT NULL,
CD_ALTERNATIVA INTEGER UNSIGNED NULL,
CD_QUESTAO INTEGER UNSIGNED NULL,
PRIMARY KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE),
INDEX CS_BI_RESPOSTAS_PAC_FKIndex1(CS_BI_IDENTIFICACAO_CD_PACIENTE),
FOREIGN KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE)
REFERENCES CS_BI_IDENTIFICACAO(CD_PACIENTE)

```

```

        ON DELETE NO ACTION
        ON UPDATE NO ACTION
    );

CREATE TABLE CS_BI_PARAMETROS_ALTERNATIVAS (
    CD_ALTERNATIVA INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    CS_BI_PARAMETROS_CD_PARAMETRO INTEGER UNSIGNED NOT NULL,
    DS_ALTERNATIVA VARCHAR(60) NULL,
    PRIMARY KEY(CD_ALTERNATIVA, CS_BI_PARAMETROS_CD_PARAMETRO),
    INDEX
CS_PARAMETROS_ALTERNATIVAS_FKIndex1(CS_BI_PARAMETROS_CD_PARAMETRO),
    FOREIGN KEY(CS_BI_PARAMETROS_CD_PARAMETRO)
    REFERENCES CS_BI_PARAMETROS(CD_PARAMETRO)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_DIAGNOSTICOS (
    CD_DIAGNOSTICO INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER UNSIGNED NOT NULL,
    NM_DIAGNOSTICO VARCHAR(40) NULL,
    DS_DIAGNOSTICO TEXT NULL,
    PRIMARY KEY(CD_DIAGNOSTICO, CS_BI_IDENTIFICACAO_CD_PACIENTE),
    INDEX CS_DIAGNOSTICOS_FKIndex1(CS_BI_IDENTIFICACAO_CD_PACIENTE),
    FOREIGN KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE)
    REFERENCES CS_BI_IDENTIFICACAO(CD_PACIENTE)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_FINANCEIRO (
    CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER UNSIGNED NOT NULL,
    CS_BI_CONVENIOS_CD_CONVENIO INTEGER UNSIGNED NOT NULL,
    DT_FATURAMENTO DATE NOT NULL,
    VL_COBRANCA DOUBLE NULL,
    PRIMARY
    CS_BI_CONVENIOS_CD_CONVENIO),
    KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE,
    INDEX CS_FINANCEIRO_FKIndex1(CS_BI_IDENTIFICACAO_CD_PACIENTE),
    INDEX CS_FINANCEIRO_FKIndex2(CS_BI_CONVENIOS_CD_CONVENIO),
    FOREIGN KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE)

```

```

REFERENCES CS_BI_IDENTIFICACAO(CD_PACIENTE)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
FOREIGN KEY(CS_BI_CONVENIOS_CD_CONVENIO)
  REFERENCES CS_BI_CONVENIOS(CD_CONVENIO)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_EXAME_LABO_PAC (
  CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER UNSIGNED NOT NULL,
  CS_BI_EXAME_LABO_CD_EXAME INTEGER UNSIGNED NOT NULL,
  DS_OBSERVACAO TEXT NOT NULL,
  PRIMARY KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE,
  CS_BI_EXAME_LABO_CD_EXAME),
  INDEX CS_EXAME_LABO_PAC_FKIndex1(CS_BI_IDENTIFICACAO_CD_PACIENTE),
  INDEX CS_EXAME_LABO_PAC_FKIndex2(CS_BI_EXAME_LABO_CD_EXAME),
  FOREIGN KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE)
  REFERENCES CS_BI_IDENTIFICACAO(CD_PACIENTE)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  FOREIGN KEY(CS_BI_EXAME_LABO_CD_EXAME)
  REFERENCES CS_BI_EXAME_LABO(CD_EXAME)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_DADOS_CLIN_PAC (
  CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER UNSIGNED NOT NULL,
  CS_BI_DADOS_CLIN_CD_DADO_CLINICO INTEGER UNSIGNED NOT NULL,
  DS_VALOR VARCHAR(200) NOT NULL,
  PRIMARY KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE,
  CS_BI_DADOS_CLIN_CD_DADO_CLINICO),
  INDEX
  CS_DADOS_CLINICOS_PACIENTE_FKIndex1(CS_BI_DADOS_CLIN_CD_DADO_CLINICO),
  INDEX
  CS_DADOS_CLINICOS_PACIENTE_FKIndex2(CS_BI_IDENTIFICACAO_CD_PACIENTE),
  FOREIGN KEY(CS_BI_DADOS_CLIN_CD_DADO_CLINICO)
  REFERENCES CS_BI_DADOS_CLIN(CD_DADO_CLINICO)
  ON DELETE NO ACTION

```

```

        ON UPDATE NO ACTION,
FOREIGN KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE)
REFERENCES CS_BI_IDENTIFICACAO(CD_PACIENTE)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_DADOS_CLINICOS_DIAG (
    CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER UNSIGNED NOT NULL,
    CS_BI_MEDICOS_INDICACOES_CD_MEDICO INTEGER UNSIGNED NOT NULL,
    CD_DIAG1 INTEGER UNSIGNED NULL,
    CD_DIAG2 INTEGER UNSIGNED NULL,
    CD_DIAG3 INTEGER UNSIGNED NULL,
    CD_DIAG_DEF1 INTEGER UNSIGNED NULL,
    CD_DIAG_DEF2 INTEGER UNSIGNED NULL,
    CD_DIAG_DEF3 INTEGER UNSIGNED NULL,
    HDA TEXT NULL,
    DT_EXAME_CLINICO DATE NULL,
    PRIMARY KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE,
CS_BI_MEDICOS_INDICACOES_CD_MEDICO),
    INDEX
CS_DADOS_CLINICOS_DIAG_FKIndex1(CS_BI_IDENTIFICACAO_CD_PACIENTE),
    INDEX
CS_DADOS_CLINICOS_DIAG_FKIndex2(CS_BI_MEDICOS_INDICACOES_CD_MEDICO),
    FOREIGN KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE)
REFERENCES CS_BI_IDENTIFICACAO(CD_PACIENTE)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    FOREIGN KEY(CS_BI_MEDICOS_INDICACOES_CD_MEDICO)
REFERENCES CS_BI_MEDICOS_INDICACOES(CD_MEDICO)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
);

CREATE TABLE CS_BI_MEDICAMENTOS_PAC (
    CS_BI_IDENTIFICACAO_CD_PACIENTE INTEGER UNSIGNED NOT NULL,
    CS_BI_MEDICAMENTOS_CD_MEDICAMENTO INTEGER UNSIGNED NOT NULL,
    DS_OBSERVACAO TEXT NULL,
    QT_DOSAGEM INTEGER UNSIGNED NULL,

```

```

PRIMARY KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE,
CS_BI_MEDICAMENTOS_CD_MEDICAMENTO),
INDEX
CS_BI_MEDICAMENTOS_PAC_FKIndex1(CS_BI_IDENTIFICACAO_CD_PACIENTE),
INDEX
CS_BI_MEDICAMENTOS_PAC_FKIndex2(CS_BI_MEDICAMENTOS_CD_MEDICAMENTO),
FOREIGN KEY(CS_BI_IDENTIFICACAO_CD_PACIENTE)
REFERENCES CS_BI_IDENTIFICACAO(CD_PACIENTE)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY(CS_BI_MEDICAMENTOS_CD_MEDICAMENTO)
REFERENCES CS_BI_MEDICAMENTOS(CD_MEDICAMENTO)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

```

```

CREATE TABLE CS_BI_QUESTIONARIO_QUESTOES (
CS_BI_QUESTIONARIOS_CD_QUESTIONARIO INTEGER UNSIGNED NOT NULL,
CS_BI_QUESTOES_CD_QUESTAO INTEGER NOT NULL,
PRIMARY KEY(CS_BI_QUESTIONARIOS_CD_QUESTIONARIO,
CS_BI_QUESTOES_CD_QUESTAO),
INDEX
CS_BI_QUESTIONARIO_QUESTOES_FKIndex1(CS_BI_QUESTIONARIOS_CD_QUESTIONARIO
),
INDEX
CS_BI_QUESTIONARIO_QUESTOES_FKIndex2(CS_BI_QUESTOES_CD_QUESTAO),
FOREIGN KEY(CS_BI_QUESTIONARIOS_CD_QUESTIONARIO)
REFERENCES CS_BI_QUESTIONARIOS(CD_QUESTIONARIO)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
FOREIGN KEY(CS_BI_QUESTOES_CD_QUESTAO)
REFERENCES CS_BI_QUESTOES(CD_QUESTAO)
ON DELETE NO ACTION
ON UPDATE NO ACTION
);

```

Estes scripts são gerados a partir da ferramenta DBDesigner. Isto é um facilitador para a criação da base, visto que o desenvolvedor não necessita digitar todos os scripts, evitando possíveis erros e equívocos durante a digitação.