UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

Trabalho de Licenciatura

USO DE VISÕES NAS BASES DE DADOS RELACIONAIS - PROBLEMAS E BENEFÍCIOS

João Sebastião Ambrósio Metambo

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

Trabalho de Licenciatura

USO DE VISÕES NAS BASES DE DADOS RELACIONAIS - PROBLEMAS E BENEFÍCIOS

Estudante: João Sebastião Ambrósio Metambo

Supervisor: dr. Virgílio Culpa

Maputo, Julho de 1996

DECLARAÇÃO

"Declaro que este trabalho é resultado da minha própria investigação, que não foi submetido para outro grau se não o indicado - **Licenciatura em Informática**, da Universidade Eduardo Mondlane".

Maputo, aos 3 de Julho de 1996

O estudante

João Sebastião Ambrósio Metambo

(João Sebastião Ambrósio Metambo)

AGRADECIMENTOS

Endereço os meus agradecimentos a todos que directa ou indirectamente contribuíram na realização deste trabalho, em especial:

O meu supervisor dr. Virgílio Culpa que incansavelmente me deu todos o seu apoio e disponibilidade desde o início, na interupção até ao final.

A dr. Maarit, que me ajudou na escolha do tema e que me deu os primeiros passos.

A dr. Esselina Macome que lutou para que eu prosseguisse com o trabalho nos momentos mais difíceis em que me encontrava, procurando sabendo a todo o custo o ponto de situação do trabalho.

A Luísa Gonçalves, que sempre esteve ao lado dando o apoio moral necessário.

Os meus pais Ambrósio e Rosa, que incansavelmente lutaram para ver o filho formado e com o diploma nas mãos.

Os meus irmãos e amigos também deram o apoio moral necessário.

O autor

ÍNDICE

Prefácio

Visões nas Bases de Dados Relacionais tem sido um tema muito falado e discutido por especialistas de informática pois pretendem proporcionar ao utilizador do computador e de vários sistemas de informação, uma fácil visão e percepção dos dados existentes nas Bases de Dados Relacionais.

Estes trabalho está estruturado com 4 (quatro) capítulos, alguns com subcapítulos que são os seguintes:

O capítulo 1, é a introdução, e nela apresentamos os objectivos deste trabalho, indicamos o local onde o estudo foi efectuado e por ser de grande importância para a empresa, mencionamos duas vantagens deste estudo, para em seguida, fazer a apresentação da sintaxe usada neste trabalho.

O capítulo 2, tem como título Base de Dados e nele falaremos um pouco sobre o modelo relacional, falaremos sobre Base de Dados, e por último uma pequena abordagem do SQL que servirá de ferramenta para a criação, manipulação e eliminação de Visões.

O capítulo 3, será o capítulo chave, Visões, pois está relacionado com o tema do trabalho "Uso de Visões nas Bases de Dados Relacionais - Problemas e Benefícios", falaremos de como podemos criar uma visão, quais as suas aplicações, quais as suas vantagens e desvantagens assim como tentaremos mostrar alguns problemas das visões.

O capítulo 4, constará o resumo do trabalho e as suas respectivas conclusões.

Por último teremos a bibliografia para depois apresentar os anexos.

1. Introdução

Este trabalho tem por objectivo o estudo de visões nas bases de dados relacionais, sua aplicabilidade, modo de utilização, regras de funcionamento na manipulação de dados, apresentação das suas vantagens e desvantagens.

Como sendo de grande importância, houve uma necessidade de se estudar e apresentar aos vários utilizadores de sistemas de informação, a vantagem que este tema traz no que diz respeito a facilidade de utilização, privacidade e independência em relação as aplicações.

O presente texto, tem como conteúdo, o estudo de visões nas bases de dados relacionais usando para tal o SQL. O estudo é justificado pela importância que ele tem num sistema de gestão de bases de dados, e pela necessidade de melhor percepção da informação existente nas tabelas da base de dados, de modo a receber dela a informação necessária.

Este estudo foi realizado na empresa Água de Maputo, mais concretamente na agência piloto da Matola, situada na cidade da Matola. No que diz respeito a importância citada acima deste trabalho, vamos apresentar duas vantagens, uma por parte do consumidor e outra que se reflete à direcção da empresa:

- Ao consumidor de água - poderá saber a sua conta cliente a qualquer altura para efectuar os pagamentos atempados dos seus débitos, assim como se não concordar pode consultar os valores das leituras, o número do contador, o endereço, pois por engano podem-lhe ser debitados consumos de outrem. Parte desta informação constante na BD (Base de Dados), poderá ser vista numa visão.

- A direcção da empresa - pode procurar saber a qualquer altura os valores facturados, as taxas de cobranças, variadas estatísticas, endereços de certos consumidores, zonas com maiores necessidades de água, etc..., isto tudo para saber se a empresa satisfaz cabalmente as necessidades dos seus clientes e que ajudará na tomada de decisões.

Uma **visão** é como uma janela que através dela, os dados da tabela podem ser vistos ou alterados. Uma **visão** é derivada de uma outra tabela ou visão que é referenciada como a tabela de base da visão - uma tabela "real" com dados que se encontram armazenados fisicamente (Bloomfield, 1990).

Em princípio, qualquer tabela derivada pode ser definida como uma visão. O processo de derivação pode envolver a projecção de certos campos de uma tabela básica, ou a união de duas ou mais tabelas básicas, ou ainda a execussão de qualquer sequência de projecções, uniões e operações similares sobre qualquer colecção de tabelas básicas (Date, 1989).

Uma visão é armazenada como uma frase SELECT somente. É uma tabela virtual, isto quer dizer, uma tabela que não existe fisicamente como tal mas que aparece aos usuários como se existisse (Bloomfield, 1990).

Quanto a sintaxe, usaremos letras maiúsculas para apresentar os comandos SQL, por exemplo,

CREATE VIEW,

as palavras que devem ser escritas pelos utilizadores nas suas aplicações ou nos exemplos apresentados serão escritas em minúsculas. Por exemplo,

CREATE VIEW nome_da_visão;

significa que o comando SQL "CREATE VIEW" deve ser seguido pelo nome da visão escolhido pelo utilizador.

Os termos <pesquisa>, <interrogação> e <inquérito> serão tratados como sinónimos sempre que pretendam traduzir a palavra inglesa <query>; igualmente <tabela> e <relação> serão tratados como sinónimos (embora existam diferenças entre os dois conceitos); o termo <atributo> e <coluna> como sinónmos e por último os termos <tupla> e <linha>.

A descrição das tabelas, as tabelas usadas vem apresentadas nos anexos.

Para a elaboração deste trabalho assim como para as experiências feitas usamos os softwares seguintes: SQLTalk - sistema de gestão de BD para Dos; Wintalk - sistema de gestão de BD para windows; Quest - sistema de gestão de BD para windows; Word 2.0 - processador de texto; Excel - 4.0 folha de cálculo e Paint Shop - captador de ecrans.

2. Base de Dados

Tabela (ou **Relação**) - Conjunto de linhas do mesmo tipo, onde não existem linhas repetidas, não existe qualquer significado na ordem das linhas da tabela. É um conceito idêntico ao de ficheiro em bases de dados não relacionais.

A popularidade da tecnologia Relacional vem aumentando com muita rapidez. Hoje em dia, várias organizações estão investigando ou implementando algumas formas da tecnologia relacional.

2.1. O Modelo Relacional

O Modelo Relacional, proposto pelo matemático E.F.Codd em 1969, então ao serviço da IBM (International Bisness Machine) no San Jose Research Laboratory, foi concebido sobre as teorias matemáticas dos conjuntos, e baseado na lógica dos predicados de primeira ordem, cobrindo os três aspectos que qualquer SGBD (Sistema de Gestão de Bases de Dados) deveria cobrir relativamente aos dados: estrutura, integridade e manipulação (Silva, 1990).

O modelo relacional foi o precursor. Pela primeira vez a visão que ele dava de uma colecção de dados não estava ligada à maneira de aceder aos dados. O conceito base é o de **Tabela** ou **Relação** que se pode sem dificuldade interpretar em termos de acontecimentos reais. Uma colecção de dados complexa não é mais que uma colecção de tabelas. Cada tabela descreve um conjunto de acontecimentos da mesma natureza.

É importante notar que o modelo relacional é um conceito relacional que define a percepção do usuário sobre os dados. Isto não implica uma especificação física ou aproximação técnica sobre o armazenamento dos dados e as estratégias de acesso.

O modelo relacional foi formalmente introduzido por Dr. E. F. Codd em 1970 e tem desenvolvido até então uma série de artigos (Fleming e Von Halle 1989).

É verdade porém, que o modelo relacional tem as suas raízes na teoria matemática de conjuntos.

O modelo relacional consiste de três conceitos:

1. *Estrutura de dados* - organização de dados, como os usuários os podem perceber.
2. *Manipulação de dados* - tipos de operações que os usuários podem usar sobre a estrutura de dados relacional.
3. *Integridade de dados* - conjunto de regras que governam como os valores dos dados relacionais se comportam quando os usuários realizam as operações relacionais.

1. Estrutura de Dados

Os dados relacionais são organizados em relações. Uma relação é um conceito intelectual baseado na teoria matemática de conjuntos, ou por outras palavras podemos dizer que uma relação é uma tabela bi-dimensional com propriedades especiais. De facto, para reforçar esta definição de relação como um tipo de tabela especial, podemos dizer que relações são tabelas relacionais.

Uma Tabela relacional consiste dum conjunto de colunas com os seus respectivos nomes e um número de linhas sem designação. Cada coluna designada está associada a um domínio, onde por sua vez o domínio é uma especificação dos valores que podem aparecer numa ou mais colunas.

As tabelas relacionais tem seis propriedades especiais que as distingue das tabelas não relacionais ou parcialmente relacionais.

Propriedade 1. *A Introdução nas colunas é simples*

Esta propriedade implica que colunas não podem conter grupos repetidos. Contudo, as mesmas tabelas são referidas como "normalizadas" ou como estando na "primeira forma normal (1NF)." É importante que se entenda a significância e o efeito desta propriedade porque é uma propriedade fundamental da estrutura relacional de dados.

Propriedade 2. *As introduções nas colunas são do mesmo tipo.*

Nos termos relacionais, esta propriedade diz que todos os valores introduzidos são do mesmo domínio.

Isto significa que cada valor numa coluna representa um valor específico para o mesmo tipo de acontecimento na tabela.

Esta propriedade, também, é significante e usual. Primeiro, implica o acesso de dados aos usuários que estão certos ou que conhecem o tipo de dados contido nas diversas colunas. Por outro lado esta propriedade implica a validação de dados. Ao se introduzirem valores o sistema vai verificando se pertencem ao mesmo domínio.

Esta propriedade, em conjunção com a primeira, dá a tabela relacional uma estrutura muito estável.

Isto implica que todas as linhas da tabela relacional tenham um "formato" que não só faz com que todas as linhas tenham o mesmo número de colunas mas que, cada coluna contenha valores definidos de um mesmo domínio.

Assim na manipulação de dados, estas propriedades podem relativamente expressar as operações através das tabelas relacionais com muita facilidade.

na perspectiva do desenho da base de dados, podemos aderir a esta propriedade. Cada coluna na tabela será definida sobre um domínio e conterá somente um tipo de dados.

Propriedade 3. *Cada linha é única.*

A propriedade 3 insurge-se de que duas linhas na tabela relacional não podem ser idênticas; é que há somente uma coluna (ou, um conjunto de colunas) de valores que identificam unicamente as linhas específicas na tabela de modo a distinguir uma linha da outra. A estas colunas chamamos de chave primária.

Esta propriedade garante que todas as linhas da tabela relacional sejam significantes. Os usuários podem referir-se a uma linha particular através da especificação do valor da chave primária.

Esta propriedade também afectará o desenho da base de dados. Quando se desenha uma base de dados relacional, requere-se que cada tabela tenha uma chave primária.

Propriedade 4. *Sequência das colunas (da esquerda para a direita) é insignificante.*

A ordem em que as colunas ocorrem é insignificante. Cada usuário pode ir buscar colunas de qualquer ordem.

O benefício óbvio é que algumas tabelas podem ser partilhadas por vários usuários e podem servir uma multidão de requerimentos de acesso. Também, os desenhadores estão livres de mudar a sequência em que as colunas se encontram fisicamente armazenadas (a não ser que por razões de realização) sem afectar o significado ou a formulação dos pedidos dos usuários.

A habilidade para partilhar dados é importante.

Propriedade 5. *Sequência das linhas (de baixo para cima) é insignificante.*

Esta propriedade é análoga a propriedade 4, mas é aplicada para as linhas em vez de colunas. O benefício óbvio é a habilidade de buscar linhas da tabela relacional em qualquer sequência.

Assim, a mesma tabela pode ser partilhada por muitos usuários, uma vez que os usuários podem ver a mesma informação em diferentes sequências. Do mesmo modo os desenhadores estão livres de modificar a ordem em que as linhas estão fisicamente armazenadas (a menos que hajam razões de realização) sem afectar o significado ou a percepção dos dados pelo usuário.

Propriedade 6. *Cada coluna tem um único nome.*

Pensando nesta propriedade como uma extensão da propriedade 4; que diz que a sequência das colunas é insignificante, as mesmas colunas são referenciadas pelos nomes e não pelas posições.

Significância da Estrutura de Dados Relacional

Como podemos ver, há importantes razões para cada uma das seis propriedades. Contudo, as propriedades debitam uma estrutura de dados que é intuitiva para os usuários, fácil de validar, flexível com respeito ao acesso dos requerimentos.

Há actualmente dois tipos de relações. A apresentada anteriormente com seis propriedades, e que é mais conhecida por **Uma Relação de Base**. Geralmente refere-se a relações de base como **Tabelas Relacionais** ou somente **Tabelas**. Há também um segundo tipo de relação chamada **Visão**.

Uma visão, do mesmo modo que uma relação de base ou tabela, é uma estrutura bi-dimensional de linhas e colunas (Fleming e Von Halle 1989).
Uma visão aparece ao usuário (muita das vezes) como sendo equivalente a uma relação de base mas, de facto reflecte o resultado de uma ou mais operações relacionais aplicadas a uma ou mais relações de base. os dados na visão não são armazenados, mas sim dinamicamente obtidos ou calculados através de tabelas, todas as vezes que a visão é referenciada pelo usuário. A visão, assim, reflecte automaticamente os valores de dados sem as tabelas associadas. Por essência, uma visão serve como uma "janela" lógica, através do qual, cada usuário pode observar sempre que o desejar, a informação das tabelas relacionais, opcionalmente apresentadas e diferentemente das próprias tabelas.

2. Manipulação de dados

O segundo componente do modelo relacional é a manipulação de dados, ou tipo de operações que o usuário pode realizar sobre as tabelas relacionais. Há dois tipos básicos de operações: Assignamento de relações para outras relações(chamadas assignamento relacional) e manipulação de relações usando oito operadores relacionais.
O conceito de assignamento relacional é como o conceito de frase de assignamento num programa em que o resultado da expressão é assignada para a variável. A variável num assignamento relacional é uma tabela relacional, e a expressão envolve tabelas relacionais e operações relacionais. Assim podem se realizar operações sobre as tabelas relacionais e assignar os seus resultados noutras tabelas relacionais.

Os oito operadores relacionais são: **select, project, product, join, union, intersection, difference e division.**

Significância da Manipulação de Dados

Os operadores relacionais definem as funções permissíveis de manipulação de dados. Eles não são uma especificação para uma linguagem de acesso de dados. Assim, estes operadores não requerem que quaisquer verbos particulares ou outras sintaxes de linguagens sejam implementadas sem que haja uma linguagem de acesso aos dados relacionais. De facto uma linguagem necessária de acesso

relacional SQL, não tem verbos explícitos para os operadores select, project, join, e outros. A mesma linguagem poderá somente suportar a funcionalidade implementada por esses operadores quando examinamos o utilizador sobre a estrutura de armazenamento e as estratégias de acesso.

Como podemos ver, as definições e propriedades governam os operadores relacionais com um poder e flexibilidade de manipulação de dados.

3. Integridade de Dados

A terceira componente do modelo relacional é a integridade de dados.
A significância dos dados dentro das tabelas relacionais podem complicar com certas regras de integridade. Essas regras de integridade constrangem valores permissíveis dentro das colunas das tabelas. Realmente, sem esses constrangimentos, os valores podem facilmente assumir valores incorrectos, incompletos, ou mesmo valores misturados.

Há duas regras gerais conhecidas para a integridade no modelo relacional, chamadas : **Regra de Integridade de Entidades** e **Regra de Integridade Referencial**.

Regra de Integridade de Entidades, diz que a chave primária não pode aceitar valores nulos. Um valor nulo, implica que o valor para uma coluna não foi aplicado, o que quer dizer desconhecido ou inapropriado. Como uma chave primária identifica uma única linha na tabela relacional, os seus valores são sempre apropriados e não devem ser desconhecidos.
Por outras palavras, a integridade de entidades obriga a que cada linha na tabela seja única. Para implementar a integridade de entidades em SQL, deve-se definir uma coluna ou um grupo de colunas como <not null> e construir um <unique index> nessas colunas.

A integridade de entidades tem a ver com a identificação de objectos, i.e., se um objecto não for identificado torna-se impossível falar sobre ele, executar qualquer operação sobre ele ou usa-lo para qualquer propósito. Portanto os objectos devem possuir uma identidade. Esta identidade em termos informáticos significa que os objectos devem ser endereçáveis, e o mecanismo de endereçamento nos sistemas relacionais é conseguido através das chaves primárias.
Como a teoria dos conjuntos pode ser directamente aplicada aos problemas de bases de dados, então:

1. *Relações (tabelas) são conjuntos;*
2. *Conjuntos não contém elementos duplicados;*
3. *Então as relações não contém valores duplicados.*

Na perspectiva do desenho da base de dados, a integridade de entidade é importante. Requer que as operações de inserção, modificação e eliminação mantenham única e existente a chave primária. Os passos do desenho incluem a identificação e forçosa a integridade de entidade.

Regra de Integridade Referencial - a integridade referencial garante a integridade entre tabelas relacionadas. Por exemplo, numa tabela de inscrições para exames não podem existir linhas sem correspondência na tabela de alunos.
O problema de integridade referencial está directamente relacionado com a definição de chaves estrangeiras. Não é mais do que assegurar que a BD não contenha uma chave estrangeira não válida, isto é sem correspondência na tabela de referência.

Significância da Integridade de Dados Relacional

A Integridade de Dados Relacional é logicamente uma parte integral do modelo relacional. Somente como modelo relacional não proporciona a implementação das especificações para a estrutura de dados e operadores de manipulação, também não dita como a integridade dos dados será implementada por um determinado produto. Ela faz, contudo, implicar que a integridade de dados seja definida e forçada sem envolver o usuário nos detalhes técnicos de implementação tais como ligação de registos ou ponteiros. Afinal de contas, a regra de integridade de dados representa um negócio de regras e não considerações técnicas.

A definição de modelo relacional também implica que a integridade de dados seja considerada como parte da implementação da base de dados e não como parte da implementação das aplicações.

Em relação as tabelas, em seguida estarão apresentadas as tabelas usadas neste trabalho:
Neste trabalho foi usada parte da base de dados dos consumidores da Empresa Água de Maputo localizados na cidade da Matola.

TESTE0 - É a base de dados onde podemos encontrar o cadastro dos consumidores e os seus respectivos contadores.

Eis os nomes das tabelas:

ABON	**AG**	**BRT**
CLIENT	**COMPTINS**	**LCLIENT**

Os comandos de criação destas tabelas podemos ver no Anexo1:

A descrição das mesmas tabelas podem ser vistas no Anexo2.

TESTE3 - Base de dados dos parâmetros

Eis os nomes das tabelas:

COPER	**MATRICULE**	**NATURE**
QUARTIER	**RUES**	**TARIF**

Os comandos de criação destas tabelas também podem ser vistas no Anexo1.
A descrição destas tabelas está no Anexo2.

2.2. Base de Dados

Uma base de dados computarizada é essencialmente um grande volume de dados armazenados com a sua descrição. O seu conceito não foi uma ruptura na tecnologia de computador, ele foi trazido gradualmente através de experiências e a pressão dos requisitos. Várias organizações contribuíram no desenvolvimento das bases de dados; incluindo fabricantes de computadores, casas de software, usuários de organizações e individualidades profissionais.

2.2.1 História da Evolução das Bases de Dados

Uma base de dados pode ser vista como a técnica mais moderna de armazenamento de dados que inicia com a invenção de cartões perfurados pelo Dr. Herman Hollerith (Hollingdale e Tootil, 1970) do Bureau de Consenso dos Estados Unidos em 1880. Dr. Hollerith confrontou-se com o problema de senso de 1880 dos 13 milhões de americanos antes de 1890 quando o senso seguinte foi oportuno. Em 1886 ficou claro que o trabalho não completou no tempo usando o significado manual de contagem. A necessidade foi a mãe da invenção: do conhecimento do uso de cartões perfurados em Jacquard Looms, Dr. Hollerith inventou o método de armazenamento de informação, introduzindo assim a era de ficheiros de cartões mecanizados que ficaram no armazenamento médio da informação principal para os seguintes 60 anos.

O primeiro computador ENIAC reaparece operacional em 1946.

Nesses primeiros dias, os computadores eram largamente usados para cálculos científicos onde a facilidade para o armazenamento de dados não era a característica importante. - a velocidade de cálculo aritmético é que era necessária. Mas quando o seu uso foi subsequentemente estendido para o processamento de dados, as limitações dos ficheiros de cartões começou a ser noticiado.

Uma das organizações que reapareceu e que consistia na imitação, foi o bureau de senso dos Estados Unidos. Encarado com a aproximação do senso de 1950, o bureau aparece particularmente ansioso em ter um rápido armazenamento à médio prazo, é essa vantagem passa a invenção de dispositivos de fita magnética.

Em 1951 um novo computador, foi chamado UNIVAC-1, desenhado pelo par ECHERT-MAUCHLEY. Esse computador tinha um único dispositivo chamado *Sistema de fita magnética*, que poderia ler uma fita magnética em ambos sentidos com uma alta velocidade (Rosen, 1969). A necessidade dum senso do bureau assim moderou duas maiores invenções nos anos 70 que foram: Os cartões perfurados e os ficheiros de fita magnética.

O impacto da fita magnética sobre o armazenamento de dados foi esmagador.

O medo da destruição acidental duma caixa de cartões e de obter cartões fora da sequência, foi também considerável. Com a chegada da fita magnética, todas essas éguas nocturnas acabaram. Elas eram claras, digno de confiança e bem arrumadas.

A sua capacidade de armazenamento e velocidade foram fenomenais comparando com um ficheiro de cartões. Contudo, a fita magnética não alterou substancialmente o modo de processamento. O ficheiro da organização era ainda sequencial, embora tenha permitido a representação do comprimento dos registos variáveis numa forma conveniente.

Todas as tecnologias do sistema de cartões como ficheiros, registos e campos foi carregada para o sistema de fita magnética. O sistema de processamento de dados usado nos anos 1950, foi um subsistema de folha de pagamento muito simples, desenhado por isolação e independência de outros subsistemas relatados. Um subsistema de folha de pagamento típico consistia de um grande número de pequenos programas com muitos ficheiros, por sua vez, cada um contendo informação fragmentada.

A introdução de discos magnéticos nos meados dos anos 60, deu um estímulo adicional em frente desta integração.

Para acessar um registo sobre uma fita magnética, é necessário procurar todos os registos intervenientes sequencialmente, mas sobre o disco, um registo pode ser acessado directamente, evitando passar por outros registos, e ganha uma total velocidade de chamada na ordem de magnitude de 2 à 4 sobre a fita magnética.

O armazenamento em disco, proporciona uma maior necessidade de suporte de hardware para grandes ficheiros integrados.

Nos meados dos anos 60, o conceito de sistema de gestão de informação (MIS - Managment Information System) ganhou popularidade.

A aproximação básica foi de executar os programas do pacote MIS sobre os ficheiros de saída sobre todos os subsistemas relevantes, mas logo, achou-se que para grande organização, o número de ficheiros de entrada para o pacote MIS era excessivamente alto com o servidor de problemas de ordenamento extensivo. Adicionalmente a falha de um sistema poderia facilmente destruir toda a operação. A duplicação de dados nos ficheiros, resulta numa actualização inconsistente trazida de outro problema.
Muitas organizações investiram grandes somas de dinheiro somente para descobrir que o pacote MIS não fora efectivo como eles gostariam que fosse.

O problema foi a falha de coordenação entre os ficheiros dos maiores sistemas.

Ele foi logo realizado para que fosse necessário uma base de dados contendo uma colecção de dados integrada e generalizada, teoricamente para todos os sistemas, uma organização servindo todos os programas de aplicação.

Foi reconhecido que uma mesma base de dados podera constituir programa e linguagem independente se for para servir a todas as aplicações; e em particular uma mudança nos dados não requererá uma mudança no programa de aplicação. Se uma base de dados serve para responder eficientemente aos conflitos necessários de todos os programas de aplicação, então ele póderá proporcionar uma adequada facilidade de representação de dados - suportada por uma variedade de técnicas de acesso de dados.

Este conceito duma base de dados nos meados dos anos 60, era usado para referir alguns ficheiros grandes.

Em paralelo com o modelo Codasyl, outras ideias baseadas em conceitos matemáticos foram também persuadidas pelos cientistas de computadores. O maior resultado desta pesquisa foi o esperado **Modelo Relacional**, por Dr. Edgar F. Codd que foi o primeiro a propor em 1970. O modelo é simples elegante e já muito poderoso. Ele apanhou a imaginação de pronta pesquisa, e essa tem sido a principal inspiração ao lado da nova área de pesquisa da base de dados. Se Bachman é o pai das bases de dados, então Codd é o pai da pesquisa de base de dados.

2.2.2. Bases de Dados Relacionais

Embora o modelo Relacional seja um conceito intelectual, uma base de dados relacional é uma materialização do uso do conceito da tecnologia de SGBD (Fleming e Halle 1989). Como a palavra indica, uma "base de dados relacional" herda dois conjuntos de características, uma define o seu aspecto relacional e outra o aspecto base de dados.

Agora estamos em altura de compreender o aspecto relacional.

Uma base de dados relacional compreende o negócio de dados que aparece ao usuário para comportar as regras do modelo relacional.
Os usuários entendem os dados como um conjunto de tabelas que obedecem as seis propriedades das relações, ou os seus equivalentes, e são protegidos pelas regras de integridade de relação.

Em relação ao aspecto "base de dados" da base de dados relacional, é muito complicado, não há uma definição universal de uma base de dados relacional, aspecto este, que é considerado como uma estrutura computarizada de armazenamento de dados que o usuário vê como tabelas relacionais. Além do conceito, todos os produtos do SGBD relacional alivia a sua própria interpretação de que é uma base de dados.

Quanto as tabelas apresentadas anteriormente, englobam duas bases de dados designadamente:

TESTE0 - base de dados dos consumidores
TESTE3 - base de dados dos parâmetros.

2.3. SQL como ferramenta

- Codd (1990), afirmou que SQL foi inventado nos finais de 1972. Embora tenha reclamado que a linguagem tinha sido baseada em vários antigos artigos seus que falavam sobre o modelo relacional, esses são bastante fracos na sua fidelidade para o modelo.
A Linguagem de interrogações estruturada (ou Linguagem de Inquéritos Estruturada, conforme os autores), SQL - Strutured Query Language, é uma linguagem para interacção com bases de dados (BD) Relacionais, e não uma linguagem completa para desenvolvimento de aplicações. A sua função é a de suportar a definição, manipulação e controlo dos dados numa Base de Dados Relacional.

A linguagem SQL surgiu como uma maneira de expressar o uso das operações anteriores sobre as bases dados relacionais, numa forma de alto nível. Estritamente como linguagem de base de dados, a SQL não possui estruturas de controlo como IF-THEN, ou WHILE, nem operadores normalmente encontrados nas linguagens de 3ª geração.

Esta é a razão por que se combina com outras linguagens de alto nível para o desenvolvimento de aplicações de manipulação de dados.

Esta ligação é estabelecida através de uma API - «Application Programming Interface», que invoca a SQL no interior de um programa em C, COBOL, PL/1, Pascal, numa linguagem de quarta geração, ou outra.

Esta «deficiência» da SQL não é sem dúvida a sua maior força: dado que a SQL não conhece nada sobre o controlo do programa, elimina-se a necessidade da linguagem de desenvolvimento de aplicações conhecer algo da estrutura ou natureza da base de dados que manipula.

A Linguagem SQL foi, e tem sido adoptada pelos principais produtores de software (perto de 100 no início do ano 1990). O SQL faz ainda parte da estratégia SAA: (Systems Aplication Architecture) da IBM. Quatro dialetos foram incorporados em sistemas de gestão de base de dados da IBM: DB2 em MVS, SQL/DS em VM, SQL/400 e Database Manager em OS/2 Extended Edition. Subsequentemente, o SQL foi adoptado como standard nos EUA pelo ANSI (American National Standard Institute), pela ISO (International Standards Organization), a nível internacional pela OSF (Open Software Foundation), pelo X/OPEN e pela FIPS (Federal Information Porcessing Standards).

2.3.1. Os Componentes do SQL

A SQL é muito mais que uma linguagem de interrogações estruturadas. Inclui características para alterar os dados de uma base de dados, e para especificar esquemas de segurança. Estas características agrupam-se em conjuntos de comandos, que são defenidos por LDD - Linguagem de Definição de Dados (<<DDL - Data Definition Language>>), LMD - Linguagem de Manipulação de Dados (<<DML - Data Manipulation Language>>) e LCD - Linguagem de Controlo de Dados (<<DCL - Data Control Language>>).

A SQL contém ainda um conjunto de palavras e símbolos usados para a construção de comandos.

- O Catálogo

Um SGBD relacional que possua a linguagem SQL, contém também um catálogo.
Um catálogo é um conjunto de tabelas especiais, que contém informação sobre as tabelas criadas pelos utilizadores - é pois uma base de dados relacional dinâmica e activa que representa a meta-data. Meta-data não é nada mais do que dados que descrevem todos os dados da base de dados, bem como o conteúdo do próprio catálogo.

Os catálogos são dependentes da versão comercial do SQL, e tem por objectivo o incremento da eficiência da linguagem. É de salientar que os catálogos não fazem parte do SQL standard.

- Linguagem de Definição de Dados (LDD)

O esquema da BD deve ser definido através dos comandos de SQL da linguagem de definição de dados, que se denomina por CREATE TABLE, CREATE INDEX, ALTER TABLE, DROP TABLE, DROP VIEW e DROP INDEX. Quando se termina a fase de definição de dados através das cláusulas e dos predicados apropriados, o resultado é um conjunto de tabelas e índices. Os nomes respectivos são armazenados em tabelas no catálogo.

- Linguagem de Manipulação de Dados (LMD)

A fase posterior à definição de dados é a manipulação dos mesmos. Os comandos da linguagem de manipulação de dados permitem manipular os dados através de inserções, alterações, eliminações e interrogações. A criação de visões faz parte da LMD. Os comandos básicos da LMD são os seguintes: INSERT, UPDATE, DELETE e CREATE VIEW.

- Linguagem de Controlo de Dados (LCD)

A linguagem de controlo de dados consiste num conjunto de comandos SQL usados para atribuir autorizações de acesso aos dados, para reservar e definir espaços, e para a administração da BD.
Alguns destes comandos representam funções que devem ser executadas pelo administrador da BD.
Os comandos da LCD são os seguintes:
Commit, Rollback, Grant e Revoke.

3. Visões

Na visão externa da arquitectura ANSI/SPARC, a base de dados é conhecida como visões externas (Date 1989).

Segundo (Date, 1989), o termo "VISÃO" é reservado em SQL (e geralmente nos sistemas relacionais) para significar, especificamente, uma tabela derivada (e não uma tabela básica).

Por sua vez, (Codd, 1990), diz que: Visões são relações virtuais representadas somente pelos seus nomes e as suas definições, e que os SGBD armazenam as definições de visões no catálogo, e suporta definições de visões expressas em termos de três alternativas seguintes: (1) só tabelas relacionais de base, (2) outras visões apenas, ou (3) mistura de tabelas relacionais de base e visões.

O "*Esquema Externo*" consiste de definições dessas visões e das tabelas de base.

A estrutura ANSI/SPARC é muito geral, permitindo variedades arbitrárias entre o nível externo e conceptual.
Em princípio, normalmente os tipos de dados estruturados suportados para os dois níveis podem ser diferentes; por exemplo, o nível conceptual pode ser baseado em relações, enquanto um dado usuário poderá ter uma visão externa da base de dados como hierárquica. Na prática, entretanto, muitos sistemas usam o mesmo tipo de estrutura como base para ambos os níveis, e os sistemas SQL não são uma excepção para esta regra geral, uma visão é ainda uma tabela, desde que o mesmo tipo de objecto é suportado por ambos os níveis.

Portanto uma visão é uma tabela virtual, isto é, uma tabela que não existe fisicamente mas é vista pelo usuário como se existisse. (Em contrapartida, uma tabela de base é uma tabela real, no sentido que, para cada linha de cada tabela, há realmente uma linha correspondente no seu armazenamento físico.

As visões não são suportadas por si só fisicamente, elas separam os dados armazenados dos distinguíveis. Além disso as suas definições são feitas com base noutras tabelas, são armazenadas no catálogo.

3.1. Criação de uma Visão

Uma visão é uma tabela virtual resultante de uma interrogação à BD, de tal modo que apenas a sua definição fica gravada e não os seus dados. Uma visão representa um subconjunto de uma ou mais tabelas, e pode ser definida como sendo uma tabela virtual (Silva 1990).

Para criar uma Visão (<View>) é usado um processo idêntico àquele usado para criar uma tabela, com ligeiras diferenças, por dois motivos principais:

- As colunas estão já definidas na tabela que vai dar origem à visão; os tipos de dados e tamanhos das colunas vão manter a estrutura que possue na tabela base, e não necessitam de ser mencionados no comando CREATE VIEW.

- Para criar uma visão, devem ser seleccionadas da(s) tabela(s), a(s) coluna(s) que se pretende que venham a integrar a lista de colunas da visão.

A sintaxe para criar uma visão é a seguinte:

```
CREATE VIEW nome_visão [(nomes_colunas_visão)]
        AS    SELECT colunas
                FROM  tabela(s)
                        WHERE  condição(ões);
```

Depois de apresentada a sintaxe para a criação de uma visão, introduzimos agora uma nova opção, WITH CHECK OPTION, que deve aparecer sempre que a visão possa ser alterada. Esta nova opção controlará se os valores inseridos satisfazem a condição WHERE. Se a visão não se destina a ser alterada, então será considerada como uma visão de leitura.

Veja-se a nova sintaxe incluindo a nova opção:

```
CREATE  VIEW  nome_visão [(nomes_colunas_visão)]
        AS   SELECT colunas
                FROM  tabela(s)
                        [WHERE  condição(ões)];
                                [WITH CHECK OPTION;]
```

A lista de nomes de colunas da visão não precisa de aparecer. Se for preenchida, então deverá conter o mesmo número de colunas e na mesma ordem que a lista de colunas escritas após o comando SELECT. Os nomes das colunas da visão não necessitam de ser idênticos aos da tabela original. Caso não sejam escritos assumem-se como sendo iguais aos nomes das colunas da tabela original.

Se a lista de nomes de colunas da visão não for preenchida, então não poderão aparecer duas colunas com nomes iguais após o comando SELECT.

A cláusula ORDER BY não pode ser usada na definição de uma visão.

Portanto com base nas tabelas apresentadas no capítulo anterior podemos criar algumas visões:

```
CREATE VIEW nomes
        AS  SELECT centre, client, ordre, denabon, nomabon, catcli, dmaj, nature
                FROM client
                        WHERE client  > "220002500"
                                WITH CHECK OPTION;
```

Este comando permite criar uma visão apartir da tabela de base *client*, contendo as colunas especificadas no comando SELECT, e cujo código seja superior a 220002500.

CENTRE	CLIENT	ORDRE	DENABON	NOMABON	CATCLI	DMAJ	NATURE
1	220002600	1	2	CAROLINA BOMBE	1	20-05-1993	1
1	220002700	1	1	M. ODETE ANASTACIO	1	20-05-1993	1
1	220002800	1	1	FELICIANO GEITA	1	20-05-1993	1
1	220002900	1	1	RACHID USSEMANE RAJU	1	20-05-1993	1
1	220002915	1	1	TEXEIRA JOSE B. CHANGALE	1	20-05-1993	1
1	220003000	1	1	JOAO TITOSSE	1	20-05-1993	1
1	220003200	1	1	ALBERTO J. TSAMBE	1	20-05-1993	1
1	220003100	1	1	HENRIQUE C. BRITO HANE	1	20-05-1993	1
1	220003300	1	1	FELIX RAFAEL JAVANE	1	20-05-1993	1
1	220003350	1	1	CALISTO DAVID MUHATE	1	20-05-1993	1
1	220003400	1	1	SEBASTIAO MENINO HELE	1	20-05-1993	1
1	220003500	1	1	ISIDRO CALADO MANJATE	1	20-05-1993	1
1	220003600	1	1	JOAO ANTONIO CHICO	1	20-05-1993	1
1	220003700	1	1	CARLOS TUAZE MASSANGO	1	20-05-1993	1
1	220003800	1	1	JACINTO M. MACUACUA	1	20-05-1993	1
1	220003900	1	1	DOMINGOS HERTIGEL	1	20-05-1993	1

Como podemos ver, o resultado da operação, é uma selecção dentre todos os clientes pertencentes a tabela client, somente apresenta os clientes com o código de identificação superior à 220002500.

Em caso de actualizações (inserções, alterações ou eliminações) da informação da tabela subjacente à visão, a informação da visão será igualmente alterada, permanecendo constantemente actualizada.

Suponhamos que a empresa pretenda obter uma listagem dos maiores consumidores de água para que tenham um tratamento especializado, pois são o suporte financeiro da empresa.

Então, para criar uma visão dos maiores consumidores de água tendo como valor mínimo 500000MT, escrevemos o seguinte comando:

```
CREATE VIEW maior_consumidor
        AS  SELECT *
                FROM lcliente
                        WHERE montant > '500000'
```

```
WITH CHECK OPTION;
```

Portanto, como podemos ver esta visão contém somente clientes com um consumo superior a 500000 MT.
É de salientar que durante o estudo na empresa Água de Maputo, a direcção solicitou uma informação pormenorizada dos maiores consumidores e que depois passou-se a dar um tratamento especial a esses consumidores, é o caso de pronta resposta em caso de avarias no contador, ropturas na conducta, ou qualquer uma anomalia verificada no sistema de abastecimento de água a esses consumidores.
O utilizador ao fazer este trabalho torna-se indiferente ao crescimento da base de dados.

Uma outra preocupação, foi a de junto à base de dados, obter as percentagens dos consumidores por escalões de consumo: de 0 à 10 m^3; de 10 à 20 m^3; de 20 à 30 m^3 e os que tem um consumo superior a 30 m^3 de água por mês, dados estes que foram importantes aquando do estudo tarifário, mais concretamente para não prejudicar a camada mais desfavorável.

Do mesmo modo podemos criar uma visão dos clientes que efectuaram o pagamento da factura do mês de fevereiro de 1996. para um fim específico.

```
CREATE VIEW pag_fev
        AS  SELECT  *
                FROM  lclient
                        WHERE   refem = '199602' AND coper = '010'
                                WITH CHECK OPTION;
```

Para esta visão *pag_fev* será o nome da visão, o (*) asterístico depois do comando SELECT dá a entender que a nova visão terá o mesmo número de colunas com a da tabela de base "*Lclient*" e que os nomes manter-se-ão inalterados. Na cláusula WHERE da condição temos as expressões *Refem* e *Coper*, que são as colunas que iremos trabalhar como base da nossa condição, portanto o *Refem* que quer dizer referência de emissão, dá-nos o mês de facturação e esse mês é identificado por quatro dígitos para o ano e dois para identificar o mês; o *Coper* (vide a tabela coper no Anexo3), que quer dizer código de operação, ele utiliza três dígitos para identificar a operação e o "010" é o código reservado à operação cobrança, portanto só aparece na tabela da conta cliente a linha dum consumidor com o código de operação "010" só e somente só depois do pagamento da factura referente ao mês em questão.

Tendo esta visão, o utilizador do sistema pode efectuar apartir desta visão, um tratamento pormenorizado das cobranças diárias feitas por diferentes caixas para apurar falhas que tem sido

muito comuns ao cobrar facturas. Geralmente os montantes totais apresentados pelos caixas diferem com os totais reais cobrados, e quando isso acontece, apesar de se dar uma margem máxima de erro ao caixa, quando se excede podem ser feitos trabalhos com visões do género para identificar a falha e atribuir responsabilidades.

3.2. Aplicabilidade

No que diz respeito as visões podemos ter várias aplicações.
Podemos criar uma visão com base numa única tabela como pudemos ver no ponto anterior; podemos criar visões através de duas tabelas ou mais tabelas reais; podemos criar e trabalhar com visões contendo expressões e/ou funções; podemos aplicar funções de grupo em visões; podemos inserir linhas através de visões; alterar dados através de visões; apagar linhas através de visões; eliminar visões ora criadas e até restringir o acesso de dados às tabelas através de visões. Todas estas considerações podemos encontrar neste subcapítulo Aplicabilidade.

3.2.1. Visões de duas ou mais tabelas

Uma visão pode possuir colunas de tabelas distintas.
Para o nosso caso concreto, temos uma tabela de nome *lclient* onde temos toda a informação da conta cliente de todos os consumidores (vide tabela *lclient* no Anexo3), mas não consta o nome do consumidor, e existe uma tabela de nome client (vide tabela client no Anexo3) onde podemos encontrar o nome do consumidor. Se um dado consumidor aparece nos balcões e pretenda saber a sua conta cliente para efectuar o pagamento das dívidas que ele tem com a empresa mas não se recordando do número da sua instalação e nem sequer ter em seu poder uma factura anterior ou recibo, sabendo somente o nome, pode-se criar uma visão entre as duas tabelas para ligar o nome do consumidor com os dados da conta cliente e satisfazer preocupações como estas de consumidores que não tenham em seu poder qualquer identificação da instalação senão o nome do próprio consumidor.

Para este caso criamos uma visão que identifique o consumidor pelo nome, refem ou seja o mês da factura e o seu valor:

```
CREATE VIEW  listagem
        AS   SELECT  client.ag, client.nomabon, lclient.refem, lclient.montant
                FROM  client, lclient
                        WHERE  client.client = lclient.client;
```

Tendo esta visão a pesquisa pode ser feita pelo nome do consumidor. Um outro utilizador pode criar uma outra visão com mais dados para resolver a questão ou criar uma visão com os mesmos dados mas dispostos numa ordem diferente desta.
Portanto os mesmos dados da base de dados podem ser vistos por diferentes utilizadores de maneiras diferentes.

Podemos ainda criar uma visão que nos dá quase toda a informação que nos facilite desbloquear qualquer preocupação, portanto código e nome do consumidor de água, localização geográfica da instalação, mês e montante da factura, número, diâmetro e ano de fabrico do contador:

```
CREATE VIEW  lista_completa
        AS  SELECT   client.ag, client.nomabon, ag.quartier, ag.nomrue, ag.cadr, ag.etage,
              ag.porte, lclient.refem, lclient.montant, comptins.compteur, comptins.diametre,
              comptins.anneefab
                    FROM   client, ag, lclient, comptins
                          WHERE  client.ag = ag.ag and ag.ag = lclient.client and
                                lclient.client = comptins.ag;
```

Assim para esta visão poderemos encontrar informação suficiente para saber os dados completos dos consumidores de água.
Na realidade podemos ver que são quatro tabelas e por sinal grandes, mas o utilizador ao trabalhar com esta visão assume como se fosse um todo, isto é como se tratasse somente duma tabela e que a mesma tabela constasse fisicamente na base de dados facilitando assim a sua percepção dos dados que na realidade se encontram dispersos na base de dados.

3.2.2. Visões com expressões e funções

Podem-se utilizar expressões e funções na criação de uma visão, desde que sejam indicados nomes para todas as colunas da linha. Por exemplo, se o consumo médio mensal para os clientes gerais públicos e industriais é de 500.000 MT, criar uma visão que apresente o código do consumidor, o mês da factura e a percentagem consumida em relação a média:

```
CREATE VIEW  percent_consumo (código, mês, percent_consumida)
        AS  SELECT  client, refem, montant/500000
                FROM  lclient
                      WHERE  coper = '001' and (catcli = "02" or catcli = "03");
```

O pedido faz mensão aos consumidores gerais públicos e gerais industriais. Na tabela Iclient existe uma coluna de nome catcli onde podemos encontrar a categoria de cada cliente que podem ser: "01" -. domésticos; "02" - geral público; "03" - geral industrial e "04" - ADM-EDM. Para mais informações sobre catcli, veja Anexo1 e 2.

3.2.3. Funções de grupo em visões

Todas as funções de grupo COUNT, COUNT(*), SUM, AVG, MAX, MIN podem ser utilizadas em visões. Por exemplo, para calcular a média das percentagens de consumo de água, escreve-se o seguinte comando:

```
SELECT  AVG(percent_consumida)
        FROM  percent_consumo;
```

O resultado será uma linha e uma coluna com o valor da média das percentagens do consumo de água por factura criada no ponto anterior.

No ponto3.1. deste trabalho, falamos de irregularidades dos caixas nas cobranças de facturas. Supomos que a direcção da empresa tenha em seu poder as estatísticas da facturação de todos os períodos e tenha se apercebido dos falhas por exemplo graves na cobrança de facturas do período de fevereiro. Querendo apurar os prejuísos, vamos supor que peça o total da cobrança das facturas do referido período.

Para responder a esta solicitação, o utilizador do sistema, tendo criado antes uma visão com todas as cobranças do período em questão, dará o seguinte comando para a obtenção do total cobrado:

```
SELECT  SUM(montant)
        FROM  pag_fev;
```

Como resultado teremos uma linha e uma coluna contendo o total cobrado para as facturas do período de fevereiro de 1996.

3.2.4. Actualização de visões

Se uma visão é derivada de uma única tabela, então no lugar de alterar a tabela original, pode-se alterar a visão da mesma maneira que se pretendia alterar a tabela original. Por exemplo, para alterar o montante da factura dos consumidores da Matola cujo período é fevereiro, escreve-se o seguinte comando:

```
UPDATE pag_fev
    SET  montant = 1.2*montant;
```

Este comando irá actualizar na tabela original todos os montantes dos consumidores referidos acrescidos de 20% do montante facturado.
No entanto, se uma visão é o resultado de uma junção de duas ou mais tabelas, então as operações de inserção, eliminação e alteração requerem uma técnica especial. Até ao presente, a alteração de visões tem sido um tema bastante discutido nas publicações da especialidade, existindo por isso opiniões divergentes quanto a sua realização. Na prática são os sistemas comerciais que implementam a SQL que impõem certas regras, devendo para tal, o leitor consultar o manual do SGBD com que trabalha para conhecer qual o processo escolhido pelo fabricante.

Como precaução, indicam-se algumas restrições para a actualização de visões criadas pela junção de duas ou mais tabelas. De uma maneira geral, uma visão pode ser actualizada da mesma maneira que uma tabela se:

- É derivada de uma única tabela,
- Não contém uma cláusula GROUP BY, ou DISTINCT,
- Não contém qualquer das funções de grupo COUNT, COUNT(*), AVG, SUM, MAX ou MIN,
- Nenhuma das colunas da visão for derivada de uma expressão aritmética ou uma constante.

Portanto consideramos como actualização de visões:

- a alteração do valor duma coluna da visão;
- a inserção de linhas na visão;
- a eliminação de linhas na visão.

3.2.4.1. Alteração do valor duma coluna da visão

Como exemplo temos o apresentado atrás:

```
UPDATE pag_fev
    SET  montant = 1.2*montant;
```

3.2.4.2. Inserção de linhas em visões

A inserção de linhas em visões segue exactamente o processo pelo qual as linhas são inseridas em tabelas. Tal como foi indicado anteriormente, a inserção de linhas derivadas da junção de duas ou mais tabelas, obedecem as regras apresentadas.
Um exemplo ilucidativo:

```
INSERT INTO  pag_fev
        VALUES ("002", "500000000", "01", "199602", "9999", "01", "010", 1996-06-20, ,
                50.4, "1", "C", "2", "2", 0);
```

3.2.4.3. Eliminação de linhas em visões

As precauções que se aplicam à alteração e inserção de valores em visões aplicam-se também à eliminação de valores em visões.

Exemplo:

```
DELETE  FROM nomes
        WHERE  centre = "002" and client = "220005000";
```

Este comando elimina a linha do consumidor identificado pelo código de instalação 220005000 e constante no centro 002. Esta linha não só desaparece da visão *nomes* mas também da tabela *client* que se encontra armazenada fisicamente na base de dados.

O usuário do sistema ao fazer as actualizações das visões não se preocupa com a reorganização das bases de dados, deixando essa tarefa para o administrador de base de dados, e se for o mesmo não o faz nas visões.

3.2.5. Eliminação de visões

Se podemos criar visões para trabalharmos com ela também podemos destruí-la,
o comando para eliminar visões em SQL é:

```
DROP  VIEW  nome_visão;
```

A eliminação de uma visão não causa efeito algum na base de dados associada, no entanto a visão deixa de estar relacionada com os índices existentes.
Ao eliminar uma visão, eliminam-se automaticamente todas as visões nela baseadas.

Por exemplo:

```
DROP VIEW maior_consumidor;
```

Neste caso a definição de visão que se encontrava no catálogo, deixa de existir após o comando DROP VIEW.

3.2.6. Restrição do acesso as tabelas atravês de visões

Já foi visto que as visões podem ser usadas para dividir a BD em segmentos de colunas e/ou linhas das suas tabelas. Seguidamente e através dos comandos GRANT e REVOKE da LCD, a informação que seja confidencial pode ser escondida de todos, excepto dos utilizadores autorizados.

A tabela de previlégios contém comandos que atribuem e retiram os previlégios sobre tabelas e visões. Um utilizador com autoridade de Aministrador de BD pode atribuir e retirar previlégios sobre qualquer tabela ou visão na BD (Gietz, 1989).

Por exemplo, se um determindo utilizador estiver permitido a trabalhar com a tabela da conta cliente (*lclient*), excepto com a coluna de montante, cria-se a seguinte visão:

```
CREATE VIEW teste_acesso
        AS    SELECT client, refem, ndoc, nature
              FROM   lclient;
```

Seguidamente seria atribuído ao utilizador uma permissão de acesso à visão no lugar do acesso à tabela:

```
GRANT SELECT, UPDATE (refem, ndoc)
        ON  teste_acesso
              TO  António, aadm;
```

Se o utilizador tiver a confiança do seu director, podem-lhe ser atribuídas todas as permissões na visão parcial da conta cliente:

```
GRANT ALL PRIVILEGES
        ON    teste_acesso
              TO  John, jadm;
```

Desta maneira a informação confidencial pode ficar escondida na tabela, enquanto que a informação não confidencial pode ser divulgada aos utilizadores através do recurso às visões.

Se existe uma informação confidencial na base de dados que não conste na visão e nem se quer teria sido dado permissão alguma de modificação aos utilizadores do sistema pelo administrador de base de dados, então podemos dizer que com toda a certeza que a segurança está garantida e esses dados.

3.3. Vantagens

As visões têm importância na maximização da performance e da segurança de um sistema numa base de dados.

3.3.1. Segurança

Uma das razões mais fortes para o uso de visões no lugar de tabelas, é a possibilidade de restringir o acesso a certas colunas da tabela. Sendo assim, as visões podem ser encaradas como um todo para oferecer segurança ao sistema.

As visões por oposição as tabelas (reais) denominam-se também de tabelas virtuais - isto é, tabelas que não existem senão em termos lógicos, mas apresentam-se ao utilizador como se existissem também em termos físicos.

3.3.2. Facilidade de utilização

A medida que a BD vai crescendo, torna-se mais conveniente para determinados utilizadores/aplicações trabalharem com visão constituída apenas pelas colunas necessárias em vez de trabalharem com toda a tabela. Assim, uma pesquisa efectuada numa visão constituída por um número de colunas restritas, revela-se mais simples que a realização da mesma pesquisa em toda a tabela.

3.3.3. Privacidade

Quem trabalha com o sistema operativo UNIX conhece certamente os acessos que se podem atribuir a um utilizador, a um grupo de utilizadores ou a todos os utilizadores do sistema (Silva 1990).
Dentro de cada aplicação, é habitual existir um sistema complementar de acessos dividido, por exemplo, em níveis. As visões servem igualmente para controlar o acesso da informação mais

delicada. Uma tabela pode conter determinadas colunas confidenciais para um determinado departamento (movimentos de contabilidade confidenciais, salários dos funcionários, classificação de desempenho, etc.); nesse caso, os restantes departamentos trabalharão com uma visão dessa tabela sem as colunas confidenciais, no lugar de trabalhar com todas as colunas da tabela.

3.3.4. Independência em relação as aplicações

Em sistemas multi-utilizador, as visões possibilitam que, em tempo real, diferentes utilizadores vejam os mesmos dados de diferentes maneiras, de acordo com as suas necessidades específicas (Silva 1990).

3.4. Desvantagens

As visões possuem igualmente algumas desvantagens

3.4.1. As visões não são armazenadas na BD, apenas a definição da visão é que fica armazenada no catálogo. A visão é recalculada de cada vez que é acedida. Sendo assim, as visões podem requerer mais tempo de processamento que as tabelas.

3.4.2. Para actualizar visões devem ser utilizadas técnicas especiais

Consideramos técnicas especiais as seguintes:

- Primeiro, a visão na sua criação deve admitir a hipótese de poder ser actualizável
- Segundo, temos de ter em conta os aspectos apresentamos no ponto 3.2.4., portanto a actualização não é para qualquer visão que tenha sido criada.

3.5. Problemas das visões

Muito embora visões sejam ferramentas úteis para consultas, apresentam problemas significativos se, actualizações, inserções ou eliminações são expressas usando visões (Korth e Silberschatz 1989).
A dificuldade é que uma modificação na base de dados expressa em termos de visão deve ser traduzida para uma modificação em tabelas reais no modelo conceitual da base de dados.

Consideremos que um caixa necessita de ver quase todos os dados referentes as cobranças, mais concretamente: centro, código do cliente, período de cobrança, natureza, etc, com excepção dos montantes cobrados por consumidor. Definimos essa visão como:

```
CREATE  VIEW  info_cobrança
        AS    SELECT centre, client, refem, ndoc, nature, dc, caisse
              FROM   lclient
                     WHERE  coper = "010"
                            WITH CHEK OPTIONS;
```

Uma vez que o SQL permite que o nome de uma visão apareça onde quer que um nome de tabela apareça, o caixa poderá escrever

```
INSERT  INTO info_cobrança
        VALUES  ("002","501000000","199504","777","01","C","02");
```

Esta inserção deve ser representada por uma inserção na tabela *lclient*, uma vez que, *lclient* é a tabela real da qual a visão *info_cobrança* é construída. Portanto para inserir uma tupla em *lclient* devemos ter algum valor para montant. Há duas considerações razoáveis para lidar com essa inserção:

- Rejeitar a inserção e retornar uma mensagem de erro ao usuário.
- Inserir uma tupla ("002", "501000000", null, "199504", "777", "01", null, null, null, null, null, "C", null, "02", null) na tabela *lclient*.

O símbolo null representa um valor nulo. Isso signnifica que o valor é desconhecido ou inexistente. A maioria dos sistemas toma a última consideração e cria valores nulos.
Entretanto a presença de valores nulos aumenta a complexidade das consultas às bases de dados. Assuma que inserimos a tupla acima, produzindo uma relação acrescida da tupla.
Consideremos a seguinte consulta:

```
SELECT  SUM (montant)
        FROM  lclient
              WHERE  coper = "010";
```

Não é possível desenvolver a adição usando null. Problemas similares aparecem quando usamos outros operadores agregados. Como resultado, todas as operações agregadas excepto COUNT ignoram valores nulos nos atributos de argumentos.

Todas as comparações envolvendo null são falsas por definição. Entretanto uma palavra chave especial null pode ser usada em um predicado para teste de um valor nulo. Para encontrar todos os clientes que aparecem na tabela lclient com valores nulos para montant, escrevemos:

```
SELECT centre, client
    FROM lclient
        WHERE montant is null;
```

O predicado is not null testa a ausência de valores nulos.

Podemos ilustrar um outro problema resultante da modificação da BD através de visões, para tal usaremos a seguinte visão:

```
CREATE VIEW cliente_morada
    AS  SELECT nomabon, nomrue, cadr
        FROM client, ag
            WHERE client.ag = ag.ag
                WITH CHEK OPTION;
```

Esta visão lista os nomes dos consumidores e o seu respectivo endereço (nome da rua e número da entrada). Consideremos a seguinte inserção nesta visão:

```
INSERT INTO client_morada
    VALUES ("JOÃO METAMBO", "Rua da Aviação", "757");
```

O único método possível de inserção de tuplas nas tabelas *client* e *ag* é inserir (null, null, null, null, null, "JOÃO METAMBO", null, null, ...) para a tabela *client*, e (null, null, null, null, null, "Rua da Aviação", null, null, null, "757", null, null, ...) para a tabela *ag*.

Suponhamos que o sistema faça isso. Então obteremos as tabelas com estas tuplas inseridas. Isto torna-se insatisfatório pois

```
SELECT *
    FROM client_morada;
```

não incluem a tupla ("JOÃO METAMBO", "Rua da Aviação", "757"). Para ver por que isso acontece, lembre-se que todas as comparações envolvendo null são definidas como falsas. Assim a cláusula where na definição da visão (client.ag = ag.ag) nunca é satisfeita para as tuplas adicionadas às tabelas *client* e *ag*.

Como resultado da anomalia apresentada muitos sistemas de BD impõem a seguinte regra nas modificações permitidas através das visões:

- Uma modificação é permitida por uma visão somente se a visão em questão é definida em termos de uma tabela da BD relacional real.

Segundo Korth e Silberschatz (1989), o problema das modificações gerais de BD através de visões é matéria de pesquisas que estão em andamento.

4. Resumo e conclusões

Concluiremos este trabalho sumarizando em termos gerais as vantagens de se separar a visão de dados de um usuário (definida por um esquema externo) da visão da comunidade (definida pelo esquema conceitual).

- Os usuários ficam imunes ao crescimento da base de dados.
- Os usuários podem ficar imunes a reestruturação da base de dados.
- A percepção do usuário fica simplificada.

É óbvio que o esquema externo permite que o usuário localize somente os dados de seu interesse. O que talvez não seja óbvio, é que, pelo menos para recuperação , o esquema externo pode também simplificar consideravelmente as operações LMD do usuário. Particularmente a necessidade de operações LMD para saltar de tabela em tabela fica bastante reduzida, porque o usuário pode ter uma visão na qual todas as tabelas envolvidas estejam unidas (Date, 1984).

- O mesmo dado pode ser visto por usuários diferentes de maneiras diferentes.
- Segurança automática fornecida aos dados secretos.
- Os usuários podem ficar indiferentes em relação as aplicações ligadas a base de dados

todas estas situações contra o facto de:

- O usuário levar um pouco mais de tempo na pesquisa em comparação com a tabela
- Se pretender actualizações das mesmas:
    a) não se esquecer de introduzir a opção WITH CHECK OPTIONS
    b) verificar se a visão é derivada de uma única tabela, caso não seja
        . não introduzir a cláusula GROUP BY ou DISTINCT
        . não introduzir as funções de grupo COUNT, COUNT(*), AVG, SUM, MAX ou MIN
        . não criar colunas derivadas de uma expressão aritmética ou constante;

para evitar os problemas de constrangimentos dos dados na base de dados ou a inserção de valores inadequados que possam criar tuplas não identificáveis ao actualizar as visões.
Podemos ver que temos mais benefícios no seu uso, pois as precauções a tomar são menores em comparação com as inúmeras vantagens encontradas;

portanto é benéfico o uso de visões nas bases de dados relacionais.

Desta forma podemos concluir que a melhor maneira de podermos visualizar e trabalhar com os dados das tabelas numa base de dados relacional é através de visões.

BIBLIOGRAFIA

Korth, H. F. e A. Silberschatz (1989). Sistema de Banco de Dados. 582 pp. São Paulo, McGraw-Hill.

Fleming, C. C. e B. V. Halle (1989). Handbook of Relational Database Design. 605 pp. U.S.A., Addison-Wesley Publishing Company.

Codd, E. F. (1990). The Relational Model for Database Management, Version 2. 538 pp. U.S.A., Addison-Wesley Publishing Company.

Silva, L. (1990). SQL Avançado, 1ª Edição. 163 pp. Lisboa, Editorial Presença.

Bloomfield, R. (1990) Introdution To Oracle RDBMS, Edition 3.0. R.S.A., Oracle Corporation UK Limited.

Date, C.J.(1984). Introdução a Sistemas de Bancos de Dados, 7ª Reimpressão. 513 pp. Rio de Janeiro, Campus.

Gietz, W. (1989). SQLTalk/Windows, version 2.0. 220 pp. Gupta Tecnologies, Inc.

ANEXO1

CRIAÇÃO DAS TABELAS

Criação da tabela ABON:

```
Create Table ABON       (CENTRE CHAR(3) NOT NULL,
                        CLIENT CHAR(12) NOT NULL,
                        ORDRE CHAR(2) NOT NULL,
                        TARIF CHAR(2),
                       FORFPERSO CHAR(6),
                        PERFAC CHAR(2),
                       MOISFAC CHAR(2),
                        DABON DATE NOT NULL,
                        DRES DATE,
                       NATURE CHAR(2) NOT NULL,
                        RECU CHAR(6));
```

Criação da tabela AG:

```
Create Table AG         (CENTRE CHAR(3) NOT NULL,
                          AG CHAR(12) NOT NULL,
                       COMMUNE CHAR(5),
                      QUARTIER CHAR(5) NOT NULL,
                          RUE CHAR(5),
                       NOMRUE VARCHAR(30) NOT NULL,
                        ETAGE CHAR(2),
                        PORTE CHAR(2),
                        CADR VARCHAR(30) NOT NULL,
                        DMAJ DATE,
                       TOURNEE CHAR(15),
                       ORDTOUR CHAR(15));
```

Criação da tabela BRT:

```
Create Table BRT        (CENTRE CHAR(3) NOT NULL,
                          AG CHAR(12) NOT NULL,
                       TOURNEE CHAR(15),
                       ORDTOUR CHAR(15),
                        DRES DATE,
                       SERVICE CHAR(1) NOT NULL,
                        CATBRT CHAR(1) NOT NULL,
                       DIAMBRT CHAR(4) NOT NULL,
                        NATBRT CHAR(1),
                        DMAJ DATE NOT NULL);
```

Criação da tabela CLIENT:

```
Create Table CLIENT        (CENTRE CHAR(3) NOT NULL,
                           CLIENT CHAR(12) NOT NULL,
                            ORDRE CHAR(2),
                              AG CHAR(12) NOT NULL,
                           DENABON CHAR(2),
                           NOMABON VARCHAR(30) NOT NULL,
                            CONSO CHAR(4),
                            CATCLI CHAR(2) NOT NULL,
                             NATIO CHAR(2),
                          TELEPHONE CHAR(9),
                          MATRICULE CHAR(5),
                           FACTURE CHAR(6),
                             DMAJ DATE,
                           NATURE CHAR(2) NOT NULL);
```

Criação da tabela COMPTINS:

```
Create Table COMPTINS      (CENTRE CHAR(3) NOT NULL,
                              AG CHAR(12) NOT NULL,
                          COMPTEUR CHAR(9) NOT NULL,
                          DIAMETRE CHAR(3) NOT NULL,
                          ANNEEFAB CHAR(4) NOT NULL,
                            POSE DATE NOT NULL,
                           DEPOSE DATE,
                           CUMCONS INTEGER,
                          DERPERFN CHAR(6),
                         DERINDEXF INTEGER,
                           DERCASF CHAR(2),
                          DERCONSF INTEGER,
                            REGFAC INTEGER,
                         ETATCONT CHAR(1));
```

Criação da tabela LCLIENT:

```
Create Table LCLIENT       (CENTRE CHAR(3) NOT NULL,
                           CLIENT CHAR(12) NOT NULL,
                            ORDRE CHAR(2),
                            REFEM CHAR(6) NOT NULL,
```

```
NDOC CHAR(6) NOT NULL,
NATURE CHAR(2),
COPER CHAR(3),
DENR DATE,
EXIG INTEGER,
MONTANT FLOAT(53),
MODEREG CHAR(1),
DC CHAR(1),
ORIGINE CHAR(3),
CAISSE CHAR(2),
ECART FLOAT(53),
```

Em seguida a lista dos índices:

```
Create Unique Index IAG
    On   AG          (AG ASC);

Create Unique Index ICLIENT
    On   CLIENT      (CLIENT ASC,
                     ORDRE ASC);

Create Unique Index IABON
    On   ABON        (CLIENT ASC,
                     ORDRE ASC);

Create Unique Index IBRT
    On   BRT         ( AG ASC);

Create Index     ICOMPTINS
    On   COMPTINS    (AG ASC);

Create Index     ILCLIENT
    On   LCLIENT     ( CLIENT ASC,
                     ORDRE ASC,
                     REFEM ASC,
                     NDOC ASC);
```

TESTE3 - Base de dados dos parâmetros

Eis as tabelas

COPER	**MATRICULE**	**NATURE**
QUARTIER	**RUES**	**TARIF**

Criação da tabela COPER:

```
Create Table COPER      (COPER CHAR(3) NOT NULL,
                        LIBCOURT CHAR(3),
                        LIBELLE VARCHAR(30),
                        COMPTGENE CHAR(10),
                        DC CHAR(1) NOT NULL,
                        DMAJ DATE,
```

Criação da tabela MATRICULE:

```
Create Table MATRICULE      (CENTRE CHAR(3) NOT NULL,
                            MATRICULE CHAR(5) NOT NULL,
                            LIBELLE VARCHAR(30),
                            PASSE CHAR(8) NOT NULL,
                            FONCTION CHAR(3),
                            CFONCT CHAR(1),
                            CODEHIER CHAR(1),
                            DMAJ DATE);
```

Criação da tabela NATURE:

```
Create Table NATURE      (NATURE CHAR(2) NOT NULL,
                         LIBCOURT CHAR(3),
                         LIBELLE VARCHAR(30),
                         COMPTGENE CHAR(10),
                         DC CHAR(1) NOT NULL,
                         DMAJ DATE);
```

Criação da tabela QUARTIER:

```
Create Table QUARTIER      (CENTRE CHAR(3) NOT NULL,
                           COMMUNE CHAR(5) NOT NULL,
                           QUARTIER CHAR(5) NOT NULL,
                           LIBELLE VARCHAR(30),
                           DMAJ DATE);
```

Criação da tabela RUES:

```
Create Table RUES       (CENTRE CHAR(3) NOT NULL,
                        COMMUNE CHAR(5) NOT NULL,
                         RUE CHAR(5) NOT NULL,
                         NRUE VARCHAR(30) NOT NULL,
                         DMAJ DATE);
```

Criação da tabela TARIF:

```
Create Table TARIF      (CENTRE CHAR(3) NOT NULL,
                        PRODUIT CHAR(2),
                         TARIF CHAR(2) NOT NULL,
                        LIBELLE CHAR(30),
                         DMAJ DATE);
```

Em seguida a lista dos índices:

```
Create Unique Index ITARIF
    On TARIF        (       CENTRE ASC,
                    PRODUIT ASC,
                     TARIF ASC);

Create Unique Index IMATRICULE
    On MATRICULE    (       CENTRE ASC,
                    MATRICULE ASC);

Create Unique Index ICOPER
    On COPER        (       COPER ASC);

Create Unique Index INATURE
    On NATURE       (       NATURE ASC);

Create Index    IRUES
    On RUES         (       CENTRE ASC,
                    COMMUNE ASC,
                      RUE ASC);

Create Unique Index IQUARTIER
```

```
On QUARTIER     (       CENTRE  ASC,
                        COMMUNE  ASC,
                        QUARTIER  ASC);
```

ANEXO2

DESCRIÇÃO DAS TABELAS USADAS

DESCRIÇÃO DAS TABELAS

ABON - nome da tabela que trata dos dados referentes à assinatura do contrato

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
CLIENT - (mecanográfico) código de identificação da instalação de água.
ORDRE - ordem de contratos efectuados na mesma instalação.
TARIF - código usado para identificar o tipo de consumidor de água, baseda na seguinte classificação: 01-Doméstico, 02-Geral Público, 03-Geral Industrial.
FORFPERSO - Empreitada individual, refere-se à média dos consumos efectuados durante um ano.
PERFAC - período de facturação.
MOISFAC - mês de facturação.
DABON - data de assinatura do contracto de água.
DRES - data de rescisão de contracto de água.
DMAJ - data de actualização do ficheiro referente aos dados da assinatura do contracto.
NATURE - código usado para identificar a natureza do consumidor com base na finalidade a que se destina a água.

AG - nome da tabela que trata dos dados sobre a localização geográfica da instalação.

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
AG - (mecanográfico) código de identificação da instalação de água.
COMMUNE - distrito urbano onde se encontra a instalação.
QUARTIER - bairro onde se encontra a instalação.
NOMRUE - nome da rua onde se encontra a instalação.
RUE - número da rua, código atribuído a rua.
ETAGE - andar, campo reservado para identificar o andar onde se encontra a instalação caso se trate de um prédio.
PORTE - número da porta.
CADR - número de entrada para a instalação.
DMAJ - data de actualização do ficheiro cliente.
TOURNEE - inspecção, refere-se à zona ou área de actuação de um leitor.
ORDTOUR - ordem das instalações numa determinada zona.

BRT - nome da tabela que contém toda informação relaccionada com o contador colocado na instalação.

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
AG - (mecanográfico) código de identificação da instalação de água.
TOURNEE - inspecção, refere-se à zona ou área de actuação de um leitor.
ORDTOUR - ordem das instalações numa determinada zona.
DRAC - data de colocação do contador.
DRES - data de retirada.
SERVICE - flag que identifica se o contador está ou não em serviço.
CATBRT - calibre do contador.
DIAMBRT - diâmetro de contador.
DMAJ - data de actualização.

CLIENT - nome da tabela que trata dos dados pessoais do cliente.

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
CLIENT - (mecanográfico) código de identificação da instalação de água.
ORDRE - ordem de contratos efectuados na mesma instalação.
AG - (mecanográfico) código de identificação da instalação de água.
DENABON - denominação a ser usada .
NOMABON - nome do abonado.
CONSO - consumo facturado.
CATCLI categoria do cliente.
NATIO - nacionalidade.
TELEPHONE - telefone do consumidor.
MATRICULE - código do utilizador que efectuou o contracto.
FACTURE - número da factura.
DMAJ - data da assinatura.
NATURE - natureza do consumidor.

COMPTINS - nome da tabela que contém a informação dos contadores e os dados para a facturação.

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
AG - (mecanográfico) código de identificação da instalação de água.
COMPTEUR - número do contador.
DIAMETRE - diâmetro de contador.
ANNEEFAB - ano de fabrico.
POSE - data de colocação.
DEPOSE - data de retirada.
CUMCONS - consumo acomulado.
DERPERFN - último período normal facturado.
DERINDEXF - último índice facturado.
DERCASF - último caso facturado.
DERCONSF - último consumo facturado.
REGFAC - regularização da factura.
ETATCONT - estado do contador (activo ou não activo).

LCLIENT - tabela onde consta toda a informação da conta cliente do consumidor

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
CLIENT - (mecanográfico) código de identificação da instalação de água.
ORDRE - ordem de contratos efectuados na mesma instalação.
REFEM - código para identificar o mês de factura com o formato: mm/aaaa, ex: 05/1994 (Maio de 1994)
NDOC - número do documento emitido (número da factura/recibo emitido)
NATURE - código da natureza do consumidor
COPER - código de operação efectuada
DENR - data de registo
EXIG - código do prazo de pagamento da factura
MONTANT - valor da factura emitida

MODEREG - código atribuído ao modo de pagamento: 1 - Dinheiro, 2 - Cheque
DC - operação efectuada: débito ou crédito
CAISSE - número do caixa onde se efectuou o pagamento
ECART - montante referente a múlta por não pagamento dentro do prazo

COPER - designação da tabela onde se encontram todos os códigos de operação

COPER - código da operação
LIBCOURT - abreviatura ou curta descrição
LIBELLE - escrição do código
COMPTGENE - número da conta a ser atribuída
DC - operação efectuada: débito ou crédito
DMAJ - data de actualização

MATRICULE - tabela de registo dos utilizadores do sistema

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
MATRICULE - código do utilizador do sistema.
LIBELLE - nome do utilizador
PASSE - password (senha de acesso ao sistema)
FONCTION - função do utilizador
CFONCT - código funcional
CODEHIER - código hierárquico
DMAJ - data de actualização

NATURE

NATURE - código da natureza do consumidor
LIBCOURT - abreviatura
LIBELLE - descrição da natureza
COMPTGENE - código da conta geral a ser atribuída por essa natureza
DC - tipo de operação: débito ou crédito
DMAJ - data de actualização

QUARTIER - tabela dos bairros que o sistema usa

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
COMMUNE - código do distrito urbano
QUARTIER - código do bairro
LIBELLE - nome do bairro
DMAJ - data de actualização

RUES - tabela das rua que o sistema usa

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
COMMUNE - código da cidade
RUE - nome da rua ou avenida
NRUE - código da rua

DMAJ - data de actualização

TARIF - nome da tabela contendo todas as tarifas em uso no sistema

CENTRE - agência ou dependência onde se podem encaixar os consumidores.
PRODUIT - código do producto a comercializar
TARIF - código da tarifa a ser aplicada
LIBELLE - descrição da tarifa
DMAJ - data de actualização

ANEXO3

TABELAS

Fonte das tabelas:

Base de Dados da Água de Maputo - Matola, criado pela Saur Internacional.

Tabela 1 abon

CENTRE	CLIENT	ORDRE	TARIF	FORFPERSO	PERFAC	MOISFAC	DABON	DRES	DMAJ	NATURE	RECU
1	220000100	1	1	120	2	2	1/1/93		31/7/93		
1	220000200	1	1		2	2	1/1/93		20/5/93		
1	220000300	1	1		2	2	1/1/93		20/5/93		
1	220000400	1	1	12	2	2	1/1/93		10/8/93		
1	220000500	1	1	120	2	2	1/1/93		31/7/93		
1	220000600	1	1		2	2	1/1/93		20/5/93		
1	220000700	1	1	180	2	2	1/1/93		31/7/93		
1	220000800	1	1	384	2	2	1/1/93		20/5/93		
1	220000900	1	1		2	2	1/1/93		20/5/93		
1	220001000	1	1	168	2	2	1/1/93		4/8/93		
1	220001100	1	1		2	2	1/1/93		20/5/93		
1	220001200	1	1	24	2	2	1/1/93		20/5/93		
1	220001295	1	1		2	2	1/1/93		10/9/93		
1	220001300	1	1	156	2	2	1/1/93		16/9/93		
1	220001400	1	1		2	2	1/1/93		20/5/93		
1	220001500	1	1		2	2	1/1/93		20/5/93		
1	220001600	1	1	120	2	2	1/1/93		23/7/93		
1	220001700	1	1		2	2	1/1/93		20/5/93		
1	220001800	1	1	264	2	2	1/1/93		23/7/93		
1	220001900	1	1	372	2	2	1/1/93		16/9/93		
1	220002000	1	1	144	2	2	1/1/93		2/10/93		
1	220002100	1	1	228	2	2	1/1/93		31/7/93		
1	220002200	1	1	96	2	2	1/1/93		31/7/93		
1	220002300	1	1	120	2	2	1/1/93		31/7/93		
1	220002350	1	1	240	2	2	1/1/93		4/8/93		
1	220002400	1	1	144	2	2	1/1/93		2/10/93		
1	220002500	1	1	228	2	2	1/1/93		22/10/93		
1	220002600	1	1	120	2	2	1/1/93		20/5/93		
1	220002700	1	1	180	2	2	1/1/93		20/5/93		
1	220002800	1	1		2	2	1/1/93		20/5/93		
1	220002900	1	1		2	2	1/1/93		20/5/93		
1	220002915	1	1	120	2	2	1/1/93		20/5/93		
1	220003000	1	1		2	2	1/1/93		20/5/93		
1	220003100	1	1	240	2	2	1/1/93		31/7/93		
1	220003200	1	1	0	2	2	1/1/93		8/2/94		
1	220003300	1	1		2	2	1/1/93		20/5/93		

Tabela 2 ag

CENTRE	AG	COMMUNE	QUARTIER	RUE	NOMRUE	NUMRUE	ETAGE	PORTE	CADR	CPARC	DMAJ	TOURNEE	ORDTOU
1	220000100	1	809		Rua Custódio Alvim Pereira				310		30/3/94	80902	2500
1	220000200	1	809		Rua Custódio Alvim Pereira				305		30/3/94	80902	2520
1	220000300	1	809		Rua Custódio Alvim Pereira				306		12/10/93	22000	1
1	220000400	1	809		Rua Custódio Alvim Pereira				311		30/3/94	80902	2460
1	220000500	1	809		Rua Alvim Pereira				312		30/3/94	80902	2420
1	220000600	1	809		Rua Dom Custodio A. Pereir				307		30/3/94	80902	2440
1	220000700	1	809		Rua Alvim Pereira				308		30/3/94	80902	2400
1	220000800	1	809		Rua Alvim Pereira				313		30/3/94	80902	2380
1	220000900	1	809		Rua Alvim Pereira				309		30/3/94	80902	2360
1	220001000	1	809		Rua Alvim Pereira				314		30/3/94	80902	2340
1	220001100	1	809		Rua de Diu				320		30/3/94	80902	2120
1	220001200	1	809		Rua de Diu				321		30/3/94	80902	2140
1	220001295	1	701		Rua da Aviação				725		30/9/93	22000	1
1	220001300	1	809		Rua de Diu				315		30/3/94	80902	2100
1	220001400	1	809		Rua do Dio				316		30/3/94	80902	2160
1	220001500	1	809		Rua de Diu				322		30/3/94	80902	2180
1	220001600	1	809		Rua de Diu				323		30/3/94	80902	2200
1	220001700	1	809		Rua de Diu				317		2/6/94	80902	2220
1	220001800	1	809		Rua de Diu				324		2/6/94	80902	2240
1	220001900	1	809						318		2/6/94	80902	2260
1	220002000	1	809		Rua de Diu				325		2/6/94	80902	2280
1	220002100	1	809		Rua de Diu				319		2/6/94	80902	2300
1	220002200	1	809						326		2/6/94	80902	2320
1	220002300	1	809		Rua do Principe				270		2/6/94	80902	1840
1	220002350	1	809		Rua António Champollimoud				10871/7		1/10/93	80904	60
1	220002400	1	809						276		2/6/94	80902	1820
1	220002500	1	809		Rua do Principe				272		2/6/94	80902	1940
1	220002600	1	809						277		2/6/94	80902	1860
1	220002700	1	809		Rua do Príncipe				272		2/6/94	80902	1920
1	220002800	1	809		Rua do Prîncipe				278		6/9/93	22000	1
1	220002900	1	809		Rua do Principe				279		2/6/94	80902	1900
1	220002915	1	809								20/5/93	80904	2260
1	220003000	1	809						280		2/6/94	80902	1960
1	220003100	1	809						281		2/6/94	80902	2000
1	220003200	1	809						282		2/6/94	80902	2060
1	220003300	1	809	0	Rua do Príncipe				273		25/8/93	22000	1

Tabela 3 brt

CENTRE	AG	TOURNEE	ORDTOUR	DRES	SERVICE	CATBRT	DIAMBRT	NATBRT	DMAJ
1	220000100	80902	2500	1/1/93		1		20	23/7/93
1	220000200	80902	2520	1/1/93		1		20	20/5/93
1	220000300	22000	1	1/1/93		1		20	20/5/93
1	220000400	80902	2460	1/1/93		1		20	20/5/93
1	220000500	80902	2420	1/1/93		1		20	23/7/93
1	220000600	80902	2440	1/1/93		1		20	20/5/93
1	220000700	80902	2400	1/1/93		1		20	20/5/93
1	220000800	80902	2380	1/1/93		1		20	20/5/93
1	220000900	80902	2360	1/1/93		1		20	20/5/93
1	220001000	80902	2340	1/1/93		1		20	20/5/93
1	220001100	80902	2120	1/1/93		1		20	20/5/93
1	220001200	80902	2140	1/1/93		1		20	20/5/93
1	220001295	22000	1	1/1/93		1		20	10/9/93
1	220001300	80902	2100	1/1/93		1		20	20/5/93
1	220001400	80902	2160	1/1/93		1		20	20/5/93
1	220001500	80902	2180	1/1/93		1		20	20/5/93
1	220001600	80902	2200	1/1/93		1		20	23/7/93
1	220001700	80902	2220	1/1/93		1		20	20/5/93
1	220001800	80902	2240	1/1/93		1		20	23/7/93
1	220001900	80902	2260	1/1/93		1		20	20/5/93
1	220002000	80902	2280	1/1/93		1		20	20/5/93
1	220002100	80902	2300	1/1/93		1		20	20/5/93
1	220002200	80902	2320	1/1/93		1		20	20/5/93
1	220002300	80902	1840	1/1/93		1		20	20/5/93
1	220002350	80904	60	1/1/93		1		20	20/5/93
1	220002400	80902	1820	1/1/93		1		20	20/5/93
1	220002500	80902	1940	1/1/93		1		20	20/5/93
1	220002600	80902	1860	1/1/93		1		20	20/5/93
1	220002700	80902	1920	1/1/93		1		20	20/5/93
1	220002800	22000	1	1/1/93		1		20	20/5/93
1	220002900	80902	1900	1/1/93		1		20	20/5/93
1	220002915	80904	2260	1/1/93		1		20	20/5/93
1	220003000	80902	1960	1/1/93		1		20	20/5/93
1	220003100	80902	2000	1/1/93		1		20	20/5/93
1	220003200	80902	2060	1/1/93		1		20	20/5/93

Tabela 4 client

CENTRE	CLIENT	ORDRE	AG	DENABON	NOMABON	MODEP	BANQUE	COMPTE	CONSO	CATCLI	NATIO	TELEPHONE	MATRICULE	FACTURE	DMAJ	NATURE
1	220000200	1	220000200	1	SALVADOR B. LANGA	0			1	1	0				20/5/93	1
1	220000400	1	220000400	1	CANDIDO MUNGUAMBE	0			1	1	0				20/5/93	1
1	220000700	1	220000700	2	ESPERANCA PEREIRA	0			1	1	0				20/5/93	1
1	220000800	1	220000800	1	CAPIAO SAMUEL FADUCO	0			1	1	0				20/5/93	1
1	220000900	1	220000900	1	JOSE CORREIA FERNANDES	0			1	1	0				20/5/93	1
1	220001000	1	220001000	1	ORANCE L. THANDIZWA	0			1	1	0				20/5/93	1
1	220001100	1	220001100	1	JOAO J. JOHANE	0			1	1	0				20/5/93	1
1	220001200	1	220001200	1	G. GANY FATEMAMAD	0			1	1	0				20/5/93	1
1	220001300	1	220001300	1	FANUEL PEDRO HUBO	0			1	1	0				20/5/93	1
1	220001400	1	220001400	1	MANUEL SIQUISSE CHISSICO	0			1	1	0				20/5/93	1
1	220001500	1	220001500	1	WILSON TEMBE	0			1	1	0				20/5/93	1
1	220001700	1	220001700	1	GLEDISSE ADAO NHABUNGO	0			1	1	0				20/5/93	1
1	220001900	1	220001900	1	ALFREDO B. SITOE	0			1	1	0				20/5/93	1
1	220002000	1	220002000	1	JOAO M. MAROSSA	0			1	1	0				20/5/93	1
1	220002100	1	220002100	1	RICARDO M. TIMBA	0			1	1	0				20/5/93	1
1	220002200	1	220002200	1	ISAC LANGA	0			1	1	0				20/5/93	1
1	220002300	1	220002300	1	FRANCISCO MATIAS PESSANA	0			1	1	0				20/5/93	1
1	220002350	1	220002350	1	MARIO JACINTO MACHAVA	0			1	1	0				20/5/93	1
1	220002400	1	220002400	1	JOSE B. M. AGUIAR	0			1	1	0				20/5/93	1
1	220002500	1	220002500	1	ENOQUE JOSHUA LIBOMBO	0			1	1	0				20/5/93	1
1	220002600	1	220002600	2	CAROLINA BOMBE	0			1	1	0				20/5/93	1
1	220002700	1	220002700	1	M. ODETE ANASTACIO	0			1	1	0				20/5/93	1
1	220002800	1	220002800	1	FELICIANO GEITA	0			1	1	0				20/5/93	1
1	220002900	1	220002900	1	RACHID USSEMANE RAJU	0			1	1	0				20/5/93	1
1	220002915	1	220002915	1	TEXEIRA JOSE B. CHANGALE	0			1	1	0				20/5/93	1
1	220003000	1	220003000	1	JOAO TITOSSE	0			1	1	0				20/5/93	1
1	220003200	1	220003200	1	ALBERTO J. TSAMBE	0			1	1	0				20/5/93	1
1	220003100	1	220003100	1	HENRIQUE C. BRITO HANE	0			1	1	0				20/5/93	1
1	220003300	1	220003300	1	FELIX RAFAEL JAVANE	0			1	1	0				20/5/93	1
1	220003350	1	220003350	1	CALISTO DAVID MUHATE	0			1	1	0				20/5/93	1
1	220003400	1	220003400	1	SEBASTIAO MENINO HELE	0			1	1	0				20/5/93	1
1	220003500	1	220003500	1	ISIDRO CALADO MANJATE	0			1	1	0				20/5/93	1
1	220003600	1	220003600	1	JOAO ANTONIO CHICO	0			1	1	0				20/5/93	1
1	220003700	1	220003700	1	CARLOS TUAZE MASSANGO	0			1	1	0				20/5/93	1
1	220003800	1	220003800	1	JACINTO M. MACUACUA	0			1	1	0				20/5/93	1
1	220003900	1	220003900	1	DOMINGOS HERTIGEL	0			1	1	0				20/5/93	1

Tabela 5 comptins

CENTRE	AG	COMPTEUR	DIAMETRE	ANNEEFAB	POSE	DEPOSE	CUMPER	CUMCONS	DERPERFN	DERINDEXF	DERCASF	DERCONSF	REGFAC	ETATCONT
1	220000100	240843	15	1968			309	34	199406	647	0	20	0	1
1	220000200	13155	15	1982	1/1/93		351	371	199406	3302	0	62	0	1
1	220000300	7160	15	1980	1/1/93		299	363	199402	2793	4	20	20	1
1	220000400	20633	15	1982	1/1/93		26	0	199406	1136	84	20	0	0
1	220000500	472608	15	1973			26	0	199406	953	84	20	0	0
1	220000600	6925	15	1983	1/1/93		351	334	199406	1405	0	49	0	1
1	220000700	9849	15	1982			26	0	199406	1526	4	30	0	0
1	220000800	5343	15	1990			26	0	199406	101	84	64	0	0
1	220000900	7364	15	1982	1/1/93		351	291	199406	3698	0	35	0	1
1	220001000	5351	15	1990	1/1/93		26	0	199406	437	84	28	0	0
1	220001100	17240	15	1982	1/1/93		351	286	199406	3504	0	46	0	1
1	220001200	2294	15	1981			309	14	199406	96	0	20	0	1
1	220001295	14540	15	1983			56	0	199404	0	84	20	0	0
1	220001300	15411	15	1982			309	88	199406	977	0	21	0	1
1	220001400	11962	15	1982	1/1/93		351	95	199406	2101	0	20	0	1
1	220001500	7441	15	1982	1/1/93		351	117	199406	2229	0	21	0	1
1	220001600	3965	15	1974			26	0	199406	0	1	20	0	0
1	220006900	4998	15				26	0	199404	1166	84	46	0	0
1	220001700	6694	20	1990	1/1/93		179	13	199406	400	93	20	0	0
1	220001800	32364	15	1969			26	0	199406	1439	84	44	0	0
1	220001900	13298	15	1980			309	238	199406	1437	0	78	0	1
1	220002000	10580	15	1979			309	194	199406	588	0	36	0	1
1	220002100	14960	15	1982			309	1468	199402	1464	4	38	76	1
1	220002200	10612	15	1982			26	0	199406	1283	84	20	0	0
1	220002300	20965	15	1982			26	0	199406	762	84	20	0	0
1	220002350	12419	15	1982			62	51	199406	51	4	40	0	0
1	220002400	379	15	1991	1/1/93		351	309	199406	561	0	138	0	1
1	220002500	4889	15	1979	1/1/93		104	19	199406	2362	4	38	0	0
1	220002600	9801	15	1982			26	0	199406	1032	84	20	0	0
1	220002700	20284	15	1983			26	0	199406	319	84	30	0	0
1	220002800	2003	20	1983	1/1/93		104	20	199404	536	4	20	0	0
1	220002900	14320	15	1982	1/1/93		68	14	199406	2356	84	20	0	0
1	220002915	999	15				312	525	199406	525	0	21	0	1
1	220003000	13143	15	1982	1/1/93		351	345	199406	2680	0	49	0	1
1	220003100	7840	15	1983	1/1/93		26	0	199406	1772	84	40	0	0
1	220003200	FA 564	15	1993	1/1/93		197	1089	199406	103	0	42	0	1
1	220003300	4140	20	1979	1/1/93		299	261	199404	845	0	75	0	1

Tabela 6 Iclient

CENTRE	CLIENT	ORDRE	REFEM	NDOC	NATURE	COPER	DENR	EXIG	MONTANT	MODEREG	DC	ORIGINE	CAISSE	ECART
2	32	1	199502	30	11	70	1994-07-25	0	2.5		D	2		0
2	1	1	199503	31	12	70	1995-03-22	0	2.5		D	2		0
2	35	1	199504	32	11	70	1995-03-29	0	2.5		D	2		0
2	505003950	1	199504	34	12	33	1995-04-20	0	50.4		D	2		0
2	37	1	199504	35	11	70	1995-04-20	0	2.5		D	2		0
2	34	1	199504	36	12	70	1995-04-11	0	72		D	2		0
2	501001300	1	199602	129	1	1	1995-05-14	1	11.4		D	2		0
2	501001600	1	199504	128	1	1	1995-05-14	1	13.4		D	2		0
2	501001700	1	199504	127	2	1	1995-05-14	1	91.2		D	2		0
2	510002800	1	199504	545	1	1	1995-05-14	1	11.4		D	2		0
2	510002810	1	199504	546	1	1	1995-05-14	1	11.4		D	2		0
2	510005063	1	199602	541	2	1	1995-05-14	1	2163.6		D	2		0
2	510005125	1	199504	548	2	1	1995-05-14	1	1172		D	2		0
2	502002401	1	199504	87	1	10	1995-05-18		27	1	C	2	2	0
2	504000680	1	199602	186	1	10	1995-05-18		13.4	1	C	2	2	0
2	502002700	1	199602	82	1	10	1995-05-18		124.4	1	C	2	2	0
2	502002500	1	199602	84	1	10	1995-05-18		109.4	1	C	2	2	0
2	505002800	1	199602	149	2	10	1995-05-18		46.4	1	C	2	2	0
2	504000200	1	199504	175	1	10	1995-05-18		209.9	1	C	2	2	0
2	504003800	1	199504	145	1	10	1995-05-19		20.9	2	C	2	2	0
2	503004000	1	199504	51	1	10	1995-05-19		45.8	2	C	2	2	0
2	503003000	1	199504	59	1	10	1995-05-19		32.6	2	C	2	2	0
2	503003000	1	199504	59	1	10	1995-05-19		0	1	C	2	2	0
2	507002940	1	199504	208	1	10	1995-05-19		11.4	2	C	2	2	0
2	510002850	1	199602	43	2	10	1995-05-19		46.4	2	C	2	2	0
2	507000300	1	199602	382	1	10	1995-05-22		11.4	1	C	2	2	0
2	35	1	199504	489	1	10	1995-05-22		11.4	1	C	2	2	0
2	501003150	1	199504	249	1	10	1995-05-22		127.4	2	C	2	2	0
2	510000060	1	199602	533	2	10	1995-05-22		46.4	2	C	2	2	0
2	505002600	1	199602	148	1	10	1995-05-22		88.4	2	C	2	2	0
2	505000800	1	199504	169	1	10	1995-05-22		109.4	2	C	2	2	0
2	501001000	1	199504	237	1	10	1995-05-23		134.9	2	C	2	2	0
2	507001100	1	199602	374	1	10	1995-05-23		11.4	2	C	2	2	0
2	510000945	1	199504	410	1	10	1995-05-23		16.9	1	C	2	2	0
2	503002200	1	199504	68	1	10	1995-05-23		160.4	2	C	2	2	0
3	401000500	1	199602	555	3	1	1995-05-22	1	128100		D	3		1000
3	401001391	1	199504	565	3	1	1995-05-22	1	92400		D	3		1000
3	401001415	1	199504	568	3	1	1995-05-22	1	708225		D	3		1000
3	501000100	1	199504	569	2	1	1995-05-22	1	47775		D	3		1000
3	501000110	1	199504	570	3	1	1995-05-22	1	226275		D	3		1000
3	501000200	1	199504	592	2	1	1995-05-22	1	776055		D	3		1000
3	501000705	1	199504	578	2	1	1995-05-22	1	1677480		D	3		1000
3	501000750	1	199504	586	3	1	1995-05-22	1	1056300		D	3		1000
3	601000200	1	199504	629	2	1	1995-05-22	1	203070		D	3		1000
3	601000260	1	199504	616	3	1	1995-05-22	1	92400		D	3		1000
3	701000350	1	199504	647	3	1	1995-05-22	1	25814250		D	3		1000
3	701000400	1	199504	657	3	1	1995-05-22	1	2207625		D	3		1000
3	701000450	1	199504	652	3	1	1995-05-22	1	15405915		D	3		1000
3	801000200	1	199504	660	2	1	1995-05-22	1	47775		D	3		1000

Tabela 7 coper

COPER	LIBCOURT	LIBELLE	COMPTGENE	DC	DMAJ
1	Feg	Factura emissão geral	99901	D	09-12-1993
2	Fei	Factura emissão isolada	99902	D	09-12-1993
4	Fod	Factura obras diversas	99903	D	09-12-1993
7	Tcr	Taxa de corte e religação	721085	D	09-12-1993
70	Deg	Deposito de garantia	132015	D	09-12-1993
75	Rdg	Restituiçao deposito garantia	132015	C	09-12-1993
102	Afc	Ajustamento fundo caixa	99909	D	09-12-1993
103	Fuc	Fundo da caixa	99910	D	09-12-1993
200	Ado	Anulação de operação	99911	D	09-12-1993
71	Dal	Despesa abertura nova lig.	721075	D	09-12-1993
73	Onl	Obras novas ligações	721075	D	09-12-1993
76	Afc	Aferiçao contador	99906	D	09-12-1993
78	Pfp	Penalidade por falta de pag.	721095	D	09-12-1993
50	Ham	Haver Manual	721015	C	23-03-1994
80	Cpf	Cobrança penalidade falta pag.	721095	C	28-01-1994
30	Cco	Colocação contador	721075	D	28-01-1994
20	Tav	Taxa de vistoria	721040	D	09-12-1993
8	Mnl	Material novas ligações	99905	D	09-12-1993
84	Hav	Haver	721015	C	09-12-1993
22	Fel	Fecho da ligação	99901	D	09-12-1993
86	Pre	Prestação	99908	D	09-12-1993
10	Cob	Cobrança	111110	C	09-12-1993
23	Ras	Rescisão assinatura	99901	D	09-12-1993
31	Rco	Retirada contador	99901	D	09-12-1993
5	Rea	Reabertura ligação	721085	D	28-01-1994
82	Cna	Cobranca não atribuida	1311115NN	D	28-01-1994
51	Dan	Dívida Anterior	131001010	D	28-01-1994
33	Sco	Substituição contador	99901	D	02-06-1994
52	Ada	Anulação Dívida Anterior	721015	C	28-01-1994
85	Afp	Anulação factura prest.	x	C	09-12-1993
32	Mco	Multa contador	99901	D	09-12-1993
300	Dfu	Despesas de funcionamento	99901	D	01-06-1994

Tabela 8 matricule

CENTRE	MATRICULE	LIBELLE	PASSE	FONTION	CFONCT	CDEHIER	DMAJ
0	513	Carlos Filipe Nhagatoa	ouof;(;`	24	9	9	3/8/93
0	514	Mário Fenias Tembe	ouof;(;`	24	9	9	3/8/93
0	515	Dinis Niquice Uane	ouof;(;`	24	9	9	3/8/93
0	510	Manecas Ricardo	ouof;(;`	24	9	9	3/8/93
0	516	Raimundo Nataniel Manjate	ouof;(;`	24	9	9	3/8/93
0	511	Abílio Alberto Guambe	ouof;(;`	24	9	9	3/8/93
0	512	Martinho Alberto Mahumane	ouof;(;`	24	9	9	3/8/93
0	517	Hilário Nhantumbo	ouof;(;`	24	9	9	3/8/93
0	518	Victorino Nhangumbe	ouof;(;`	24	9	9	3/8/93
0	101	Sidonio Jose Chiziane	Mm^oo(M<	24	8	8	15/4/94
0	102	Fernando Zacarias Zucule	GN]57nTF	24	8	8	14/9/93
0	103	Maria Eugenia Sibia	oooZP=P~	24	8	8	14/4/94
0	519	Alfredo Simbine	ouof;(;`	24	9	9	2/9/93
0	104	Maria Joana Pais	eOY0~24d	24	8	8	15/4/94
0	10	Hortensio	io}edmgO	24	7	8	6/7/94
0	11	Escrivão	i▒u}e{gO	24	8	8	6/7/94
0	520	Palau Samo	ouof;(;`	24	9	9	2/9/93
0	12	Agostão José Zitha	oM_P,Qxw	24	8	8	14/4/94
0	100	Ana Paula I. Ibrahim	RYkNmvsC	24	8	8	14/6/94
0	0	PEPE Thierry	s}onubsC	100	9	9	30/9/93
0	500	Hilario Simeão Manjate	mfoo_o_)	24	8	8	14/4/94
0	11111	GUILLET Christophe	35<>\)]H	100	8	8	15/11/93
0	105	Adelaide da Conçeicão	os▒e}*Z	24	8	8	15/4/94
0	600	Jose Luis Afonso Mate	Og_yYlkG	24	8	8	14/4/94
0	107	Paula Maria Angelina	~woi~^V#	24	8	8	15/4/94
0	521	Nunes	ouof;(;`	24	9	9	27/11/93
0	522	Alexandre	ouof;(;`	100	9	9	13/12/93
0	523	Jaime Matsinhe	ouof;(;`	100	9	9	16/12/93
0	106	Virginia Antonio Uamusse	▒GW[4lYl	24	8	8	15/4/94
0	502	António da Conceição C.Manjate	ouof;(;`	24	8	8	29/9/93
0	503	Flávio L.Pondza	ouof;(;`	24	8	8	29/9/93
0	504	Joaquim Saveca	ouof;(;`	24	9	9	2/9/93
0	506	Manuel B.Tembe	ouof;(;`	24	9	9	3/8/93
0	507	Afonso J.Mbembele	ouof;(;`	24	9	9	3/8/93
0	508	Moisés C.Cumaio	ouof;(;`	24	9	9	3/8/93
0	501	Anacleto Francisco	ouof;(;`	24	8	8	29/9/93
0	505	Joaquim T. Migaços	ouof;(;`	24	8	8	29/9/93
0	509	Esmeraldino Fonseca	ouof;(;`	24	9	9	3/8/93
0	1	João Metambo	koq~i~o9	1	9	9	25/7/94
0	99999	PEPE Thierry	__YWaw^z	100	9	9	13/1/93
0	13	Pedro Mário Paulino	ouof;(;`	24	8	8	19/1/94
0	14	Maria Isabel Cutana	oowHTgf)	24	8	8	14/4/94
0	524	José Bucuana	ouof;(;`	100	9	9	21/2/94
0	108	Maria de Lurdes D. de Almeida	}▒▒}0B@j	24	8	8	15/4/94
0	15	António Chiúle	[WWTMS=`	24	8	8	15/4/94
0	109	Marta Vasco Mambo	cwOy7\%W	24	8	8	22/4/94
0	110	Hernani Mussa	\M]M"]O"	24	8	8	21/6/94
0	111	Ilca Ernesto Neves	{|ncGl1]	24	8	8	21/6/94

Tabela 9 quartier

CENTRE	COMMUNE	QUARTIER	LIBELLE	DMAJ
1	1	709	T-3	06-09-1993
1	1	711	Ndalhavela	06-09-1993
1	1	712	Unidade D	06-09-1993
1	1	701	Machava sede	07-07-1993
1	1	702	Trevo	07-07-1993
1	1	708	Tsalala - Matola Gare	07-07-1993
1	1	710	Acordos de Lusaka	06-09-1993
1	1	706	Cingatela	07-07-1993
1	1	713	Vale do Infulene	06-09-1993
1	1	814	Matola Rio	20-09-1993
1	1	801	Matola A	07-07-1993
1	1	813	Malhompsene	07-07-1993
1	1	802	Matola B	07-07-1993
1	1	803	Matola C	07-07-1993
1	1	804	Matola D	07-07-1993
1	1	805	Matola F	07-07-1993
1	1	811	Mussumbuluco	07-07-1993
1	1	807	Matola H	07-07-1993
1	1	809	Fomento	07-07-1993
1	1	707	Bunhissa	07-07-1993
1	1	810	Liberdade	07-07-1993
1	1	812	Cicuana	07-07-1993
1	1	704	P. Lumumba	07-07-1993
1	1	705	São Damaso	07-07-1993
1	1	703	Infulene Unidade A	20-01-1994
1	1	808	Matola J	07-07-1993
1	1	806	Matola G	07-07-1993

Tabela 10 rues

CENTRE	COMMUNE	RUE	NRUE	DMAJ
1	1	1	Av. Abel Batista	10-05-1993
1	1	2	Rua Coqueiros	10-05-1993
1	1	3	Rua do mercado	10-05-1993
1	1	4	Av. Fernando Lira	10-05-1993
1	1	5	Av. de Angola	10-05-1993
1	1	6	Rua das acacias	10-05-1993
1	1	7	Rua das tilias	10-05-1993
1	1	8	Rua das flores	10-05-1993
1	1	9	Rua das mangueiras	10-05-1993
1	1	10	Rua dos abacateiros	10-05-1993
1	1	11	Av. Eng. Amancio Cruz	10-05-1993
1	1	12	Rua Virgilio Rodrigez	10-05-1993
1	1	13	Av. de Moçambique	10-05-1993
1	1	14	Av. Eduardo Capucho Paulo	10-05-1993
1	1	15	Av. da Liberdade	10-05-1993
1	8	16	Av. Manuel Boullosa	10-05-1993
1	1	1236	Rua Rui Pina	18-08-1993
1	1	1232	Av. Honório Barreto	18-08-1993
1	1	1235	Rua Felisberto Machatine	18-08-1993
1	1	1234	Rua Damião de Góis	18-08-1993
1	1	1233	Av. Fernão Lopes	18-08-1993
1	1	1231	Av. Aniceto do Rosário	18-08-1993
1	1	1230	Av. Régulo Mucapera	18-08-1993
1	1	1229	Rua Major Curado	18-08-1993
1	1	1228	Rua Oliveira Martins	18-08-1993
1	1	1223	Rua Lino dos Santos	18-08-1993
1	1	1222	Rua Direita	18-08-1993
1	1	1211	Rua da Escola	18-08-1993
1	1	1210	Rua da Alegria	18-08-1993
1	1	1208	Rua dos Coqueiros	18-08-1993
1	1	1207	Rua dos Limoeiros	18-08-1993
1	1	1206	Rua das Laranjeiras	18-08-1993
1	1	1204	Rua do Fomento	18-08-1993
1	1	1203	Av. do Douro	18-08-1993
1	1	1202	Av. do Minho	18-08-1993
1	1	1201	Rua Paula Isabel	18-08-1993
1	1	1200	Rua de Moçambique	18-08-1993
1	1	908	Rua Nossa Srª da Saúde	18-08-1993
1	1	907	Rua da Esperança	18-08-1993
1	1	906	Rua Principal	18-08-1993
1	1	905	Rua de Camões	18-08-1993
1	1	900	Rua Padre Américo	18-08-1993
1	1	590	Rua dos Cajueiros	18-08-1993
1	1	437	Av. da Indústria	18-08-1993
1	1	254	Rua da Guarda	18-08-1993
1	1	282	Rua de Castelo Branco	18-08-1993
1	1	278	Rua de Aveiro	18-08-1993
1	1	279	Rua de Viseu	18-08-1993
1	1	277	Rua de Braga	18-08-1993
1	1	283	Rua de Portalegre	18-08-1993
1	1	285	Rua de Setúbal	18-08-1993
1	1	286	Rua Projectada á de Portalegre	18-08-1993
1	1	300	Av. Afonso Costa	06-09-1993
1	1	301	Av. Almirante Alves Leite	06-09-1993
1	1	302	Av. Álvaro de Castro	06-09-1993
1	1	303	Av. António Enes	06-09-1993
1	1	304	Av. Caldas Xavier	06-09-1993
1	1	305	Av. Castro da Silva	06-09-1993
1	1	306	Av. Comandante Vasco Rodrigues	06-09-1993

Tabela 11 Nature

NATURE	LIBCOURT	LIBELLE	COMPTGENE	DC	DMAJ
4	VAE	Venda agua domesticos ADM.EDM	131111504	D	27-01-1994
12	PDA	Prod. div. administração	131111502	D	27-01-1994
11	PDD	Prod. div. domesticos	131111501	D	27-01-1994
1	VAD	Venda agua domesticos	131111501	D	27-01-1994
13	PDG	Prod. div. geral	131111503	D	27-01-1994
0	VEA	Venda agua	131001115	D	23-06-1994
2	VAA	Venda agua administração	131111502	D	27-09-1993
3	VAG	Venda agua geral	131111503	D	27-01-1994
14	PDE	Prod. div. domesticos ADM.EDM	131111504	D	27-01-1994