

GLAUCO BONAN NEGRI

RECUPERAÇÃO DE DADOS UTILIZANDO ORACLE FLASHBACK

ASSIS

2009

# RECUPERAÇÃO DE DADOS UTILIZANDO ORACLE FLASHBACK

GLAUCO BONAN NEGRI

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: \_\_\_\_\_

Analisador (1): \_\_\_\_\_

ASSIS  
2009

GLAUCO BONAN NEGRI

RECUPERAÇÃO DE DADOS UTILIZANDO ORACLE FLASHBACK

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: \_\_\_\_\_

Área de Concentração: \_\_\_\_\_

\_\_\_\_\_

ASSIS

2009

## RESUMO

Este trabalho de conclusão de curso tem por objetivos abordar e mostrar as diversas funcionalidades da Ferramenta Oracle Flashback e as facilidades que essa ferramenta proporciona aos usuários, desde a recuperação de chaves, atributos, tabelas e até mesmo de todo os banco de dados.

**Palavras-chave:** Banco de Dados, Timestamp, Date, Aspecto de Tempo, Flashback.

## ABSTRACT

This work of completion aims to address and display the various features of the tool Oracle Flashback and the facilities that this tool provides users, since the recovery of keys, attributes, tables and even the entire database.

**Keywords:** Database, Timestamp, Date, Aspect of Time, Flashback.

## LISTA DE FIGURAS

Figura 1 - Sistema de Banco de Dados.....	13
Figura 2 - RECYCLEBIN.....	20
Figura 3 - Proposta Geral do Trabalho.....;	25
Figura 4 - Ferramentas Flashback.....;	28
Figura 5 - Tabelas Gerais.....	29
Figura 6 - Tabela instituicoes.....	30
Figura 7 - Tabela participantes.....	30
Figura 8 - Tabela tipoeventos.....;	30
Figura 9 - Tabela evento.....	31
Figura 10 - Tabela doador.....	31
Figura 11 - Tabela doacoes.....	31
Figura 12 - Tabela noticia.....	32
Figura 13 - Tabela voluntariados.....	32
Figura 14 - Relacionamento instituicoes para doador.....;	34
Figura 15 - Relacionamento instituicoes, doador, doacoes.....;	35
Figura 16 - Relacionamento instituicoes, participantes, evento, tipoevento...36	
Figura 17 - Criação e Inserção da Tabela Instituições.....	41
Figura 18 - Criação e Inserção da Tabela Notícia.....	41
Figura 19 - Criação da Tabela Tipoeventos.....	49
Figura 20 - Criação da Tabela Evento.....	49
Figura 21 - Criação da Tabela Doador.....;	52
Figura 22 - Criação da Tabela Doações.....	52
Figura 23 - Criação e Inserção da Tabela Participantes.....	56
Figura 24 - Criação e Inserção da Tabela Voluntariados.....	57

## LISTA DE TABELAS

Tabela 1 – Tabela de Notícia.....	42
Tabela 2 – Tabela de Notícia alterada.....	43
Tabela 3 – Tabela de Notícia recuperada.....	43
Tabela 4 – Tabela de Instituições.....	44
Tabela 5 – Tabela de Instituições alterada.....	45
Tabela 6 – Tabela de Instituições recuperada.....	46
Tabela 7 – Banco de Dados recuperado.....	48
Tabela 8 – Tabela doador utilizando Flashback Version Query.....	53
Tabela 9 – Tabela doações utilizando Flashback Version Query.....	54
Tabela 10 – Verificação do Flashback Data Archive.....	58
Tabela 11 – Tabela participantes.....	59
Tabela 12 – Tabela participantes recuperada.....	60

## LISTA DE ABREVIATURAS E SIGLAS

SGBD	Sistemas Gerenciadores de Banco de Dados
DBA	Administradores de Banco de Dados
DATATYPE	Tipos de Dados
UNDO	Automatic Undo Managment
SCN	System Change Number (Sequenciador Interno)
SYSDBA	Privilégio de Sistema
RECYCLEBIN	Lixeira do Sistema de Banco de Dados
BDR	Banco de Dados Relacional
DBMS	Pacotes padrão da Oracle
VERSIONS_STARTSCN	SCN inicial da versão de transação
VERSIONS_STARTTIME	TIMESTAMP inicial da versão de transação
VERSIONS_ENDSCN	SCN final da versão de transação
VERSIONS_ENDTIME	TIMESTAMP inicial da versão de transação
VERSIONS_XID	Id da transação que criou a versão da linha
VERSIONS_OPERATION	Operação executada na linha
XID	Id da transação
START_SCN	Numero do SCN que inicia a transação
START_TIMESTAMP	Horário de inicio da transação
COMMIT_SCN	SCN gerado no COMMIT
COMMIT_TIMESTAMP	Horário do COMMIT
UNDO_SQL	SQL gerado na transação

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>10</b>
<b>2</b>	<b>SISTEMA DE BANCOS DA DADOS.....</b>	<b>12</b>
2.1	SISTEMA GERENCIADOR DE BANCOS DE DADOS.....	12
<b>3</b>	<b>ASPECTO DE TEMPO.....</b>	<b>14</b>
3.1	O TIPO DATE.....	15
3.2	TIMESTAMP.....	16
<b>4</b>	<b>FERRAMENTAS FLASHBACK.....</b>	<b>18</b>
4.1	FLASHBACK QUERY.....	18
4.2	FLASHBACK DATABASE.....	19
4.3	FLASHBACK DROP.....	20
4.4	FLASHBACK TABLE.....	21
4.5	FLASHBACK TRANSACTION QUERY.....	21
4.6	FLASHBACK VERSION QUERY.....	22
4.7	FLASHBACK DATA ARCHIVE.....	23
<b>5</b>	<b>PROPOSTA DO TRABALHO.....</b>	<b>24</b>
5.1	PASSO 1. TABELAS.....	25
5.2	PASSO 2.RESTRIÇÕES.....	26
5.3	PASSO 3. RELACIONAMENTOS .....	26
5.4	PASSO 4. DADOS.....	27
5.5	APLICAÇÃO DAS FERRAMENTAS.....	28
<b>6</b>	<b>ESTUDO DE CASO.....</b>	<b>29</b>
6.1	APLICAÇÃO DO PASSO 1: TABELAS.....	30
6.2	APLICAÇÃO DO PASSO 2: RESTRIÇÕES.....	32
6.3	APLICAÇÃO DO PASSO 3: RELACIONAMENTOS.....	34
6.4	APLICAÇÃO DO PASSO 4: MOSTRANDO OS DADOS.....	36
6.5	APLICAÇÃO DO PASSO 5: USANDO AS FERRAMENTAS.....	41
6.5.1	<b>FLASHBACK QUERY.....</b>	<b>41</b>
6.5.2	<b>FLASHBACK DATABASE.....</b>	<b>46</b>



6.5.3	FLASHBACK TABLE E FLASHBACK DROP.....	49
6.5.4	FLASHBACK VERSION QUERY.....	51
6.5.5	FLASHBACK TRANSACTION QUERY.....	55
6.5.6	FLASHBACK DATA ARCHIVE.....	56
7	CONCLUSÃO.....	61
8	REFERÊNCIAS.....	62

## 1. INTRODUÇÃO

Com a rápida evolução dos Sistemas Gerenciadores de Bancos de Dados (SGBD) existentes e os treinamentos de Administradores de Bancos de Dados (DBA) para melhoria de desempenho e agilidade no processo de desenvolvimento, muitos desses administradores acabam por sua vez sobrecarregando a base de dados e cometendo erros prejudiciais aos projetos ou aos dados empresariais importantes, realizando rotinas para eliminar tabelas, relacionamentos, atributos, destruindo assim, dados vitais para a sobrevivência da empresa.

Por razão desses constantes erros imprevisíveis, bem como a demora e a complexidade dos *backups*, a Oracle disponibilizou diversos recursos que permitem a desenvolvedores ou administradores de bancos de dados recuperarem os dados e objetos, sem a necessidade de recorrer aos *backups extensos*. Esses recursos estão disponibilizados nas ferramentas Oracle Flashback.

Várias ferramentas de recuperação vêm sendo desenvolvidas pela Oracle, a versão Oracle 9i é a pioneira em relação a Flashback distribuindo várias ferramentas para seus usuários. Desde então, foram feitas oito tipos de ferramentas como: Flashback Query, Flashback Database, Flashback Drop, Flashback Table, Flashback Transaction Query, Flashback Version Query, Flashback Data Archive e Flashback Transaction Backout.

Estas ferramentas têm suas funcionalidades de grande utilidade para administradores, já que permitem recuperar desde uma tabela até a própria base de dados por completo, bem como manter e monitorar os comandos e alterações efetuadas nos bancos de dados.

Para tal, este trabalho está dividido em seis capítulos, sendo este primeiro a Introdução.

No Capítulo 2 serão apresentados conceitos sobre Sistemas de Bancos de Dados.

No Capítulo 3 serão apresentados conceitos e noções sobre Aspecto “Tempo”, essencial para o entendimento e para o uso das ferramentas de Flashback.

No Capítulo 4 serão apresentadas as ferramentas de Flashback, bem como suas funcionalidades específicas.

No Capítulo 5, serão mostradas as propostas que este trabalho apresenta assim como tabelas, relacionamentos, restrições, dados, e as aplicações das ferramentas.

No Capítulo 6, serão mostradas as funcionalidades das ferramentas de Flashback.

## **2. SISTEMA DE BANCOS DE DADOS**

O gerenciamento de banco de dados está evoluindo, deixando de ser uma aplicação especializada para se tornar o componente central de um ambiente moderno da computação. Estes são conjuntos de registros dispostos em estrutura regular que possibilita a reorganização dos mesmos e a produção de informação. Um banco de dados normalmente agrupa registros utilizáveis para um mesmo fim.

De uma forma bem simplificada e geral, pode-se entender por sistema de banco de dados qualquer sistema que reúna e mantenha organizada uma série de informações relacionadas a um determinado assunto em uma determinada ordem. A lista telefônica é um exemplo, nela percebe-se que todos os dados referentes a uma pessoa estão na mesma linha, chamado de registro.

O tipo ou categoria da informação (nome, telefone, etc.) sobre uma pessoa, está separado em colunas, as quais são conhecidas como atributos.

Ainda um sistema de banco de dados deve garantir a segurança das informações armazenadas contra eventuais problemas com o sistema, além de impedir tentativas de acesso não autorizadas.

Como exemplo, considere a área de um banco responsável por todas as informações de seus clientes, bem como de suas contas-poupança. Um modo de guardar as informações no computador é armazená-las em sistemas de arquivos permanentes. Para permitir aos usuários a utilização dessas informações, o sistema deve apresentar um conjunto de programas de aplicações que tratam esses arquivos (SILBERSCHATZ, KORTH, SUDARSHAN, 2006).

Um banco de dados é usualmente mantido e acessado por meio de um Sistema Gerenciador de Banco de Dados.

### **2.1 SISTEMA GERENCIADOR DE BANCO DE DADOS (SGBD)**

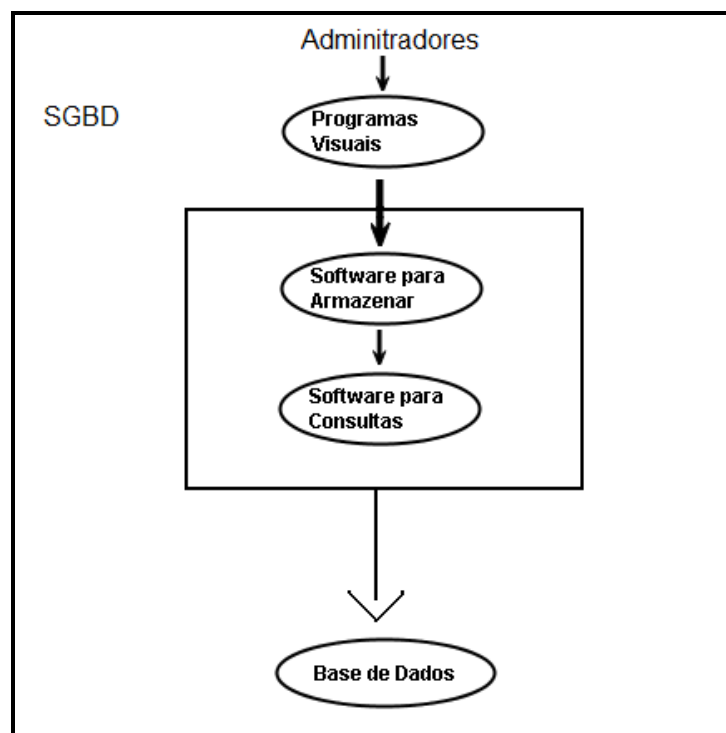
O primeiro Sistema Gerenciador de Banco de Dados comercial surgiu no final de 1960 com base nos primitivos sistemas de arquivos disponíveis na época, os quais não controlavam o acesso concorrente por vários usuários. Os SGBDs

evoluíram desses sistemas de arquivos de armazenamento em disco, criando novas estruturas de dados com o objetivo de armazenar informações. Com o tempo, os SGBD's passaram a utilizar diferentes formas de representação, ou modelos de dados, para descrever a estrutura das informações contidas em seus bancos de dados (TAKAI, ITALIANO, FERREIRA, 2005).

Um Sistema Gerenciador de Banco de Dados ou Sistema Gestor de Base de Dados é uma coleção de programas que permite a criação e gerência de base de dados, antes dos SGBDs as aplicações utilizavam sistemas de arquivos do Sistema Operacional.

Ele é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados, este conjunto de dados, é comumente chamado de Banco de Dados. Logo este é uma coleção de dados inter-relacionados e uma coleção de programas para acesso a dados. O principal objetivo do SGBD é proporcionar um ambiente tanto conveniente, quanto eficiente e flexível para a recuperação e armazenamento das informações do banco de dados onde esses são projetados para gerar um grande volume de informações reduzindo o tempo de desenvolvimento da aplicação (SILBERSCHATZ, KORTH, SUDARSHAN, 2006).

A Figura 1 ilustra uma visão geral de um Sistema de Banco de Dados.



**Figura 1. Sistema de Banco de Dados**

### 3. ASPECTO TEMPO

Um banco de dados pode ser classificado sobre várias características. O sentido deste tópico é explicar a utilização do aspecto tempo relacionados ao Flashback.

Para a representação de dados temporais utilizam-se alguns tipos de dados tais como: DATE, TIMESTAMP, etc.

A opção presente em Aspecto de tempo é armazenar as informações sempre que ocorre um dado evento ou quando um evento é considerado verdadeiro. Um evento pontual, ou fato pontual, normalmente são associados aos bancos de dados por um único ponto de tempo representado por alguma granularidade. Esta informação é freqüentemente representada por uma série de dados temporais. Séries de dados temporais envolvem valores, que são registrados de acordo com uma sucessão específica, e predefinida de tempo (ELMASRI, NAVATHE, 2005).

Segundo Angonese (1999) os bancos de dados podem ser divididos em quatro tipos diferentes em relação a armazenagem por aspecto de tempo: bancos de dados instantâneos, de tempo de transação, de tempo de validade e bitemporais.

- Instantâneos: são bancos de dados convencionais, onde são armazenados somente valores de tempo presentes.
- De Tempo de Transação: uma alternativa de armazenamento de informações temporais é associar a cada valor definido o tempo de transação, sob forma de um rotulo temporal (Timestamp).
- De Tempo de Validade: associa a cada informação somente o tempo de sua validade no mundo real.

- Bitemporais: é a forma mais completa de armazenar informações temporais, nos quais os tempos de transação e de validade são associados a cada informação.

Sistemas Gerenciadores de Bancos de Dados também tem por sua função gerenciar a questão de “volta ao tempo” por intermédio de vários formatos de atributos. Os dois tipos de dados para o armazenamento e para retornar *backup* de dados, com a diferença de armazenar frações de segundos com precisão de até 9 dígitos, são:

- DATE: Protocolo para representação referente a data e hora completos ou abreviados
- TIMESTAMP: Praticamente um dos protocolos mais usados no Oracle incluindo frações de segundos.

### 3.1 O TIPO DATE

Este é o DATATYPE (tipo de data), um tipo de atributo bem familiar utilizado na representação dos valores de data e tempo onde dados podem ou poderão ser mudados no futuro.

O tipo de dado DATE do Oracle é um tipo especial capaz de armazenar datas entre 4712 a.C. a 9999 d.C, mas além de armazenar informações de século, ano, mês e dia, o mesmo também é capaz de armazenar informações de hora, minuto e segundo (LEGATTI, 2009).

O atributo do tipo DATE costuma ser usado em tabelas para a representação de informação referente à data e hora de algum evento. Estes ocupam apenas 7 bytes e são capazes de armazenar datas com a precisão, na casa de segundos. Melhor detalhando, um atributo DATE é capaz de manter as seguintes informações temporais associadas a um determinado evento: século, ano, mês, dia, hora, minuto e segundo de ocorrência do evento.

Um dos problemas com o tipo de dados DATE, é sua granularidade, onde tenta determinar um intervalo de tempo entre dois eventos quando os eventos acontecem sem 1 segundo de diferença entre eles.

Para aplicações onde necessitam manter informações de tempo com maior precisão, o Oracle disponibiliza outro tipo de dado, denominado TIMESTAMP.

### **3.2. TIMESTAMP**

Um dos maiores problemas com o tipo de dados DATE, era sua inabilidade para granular o suficiente para determinar qual evento deve acontecer primeiro em relação a outro evento.

Oracle expandiu-se no tipo de dados DATE e disponibilizou o tipo de dados TIMESTAMP, que guarda todas as informações que o tipo de dados DATE armazena, mas também inclui frações de segundos (milissegundos, micro segundos, etc.).

Este tipo é explicitamente utilizado para armazenar datas e horários em que a diferença de tempo é de grande importância.. Calculando a diferença de tempo entre dois tipos de dados TIMESTAMP, é muito mais fácil do que o velho tipo de dados DATE.

Portanto é um protocolo de ordenação que assegura quaisquer operações de leitura e escrita onde essas sejam executadas por ordem de timestamp. Dando dois exemplos, quando se usa uma determinada hora do relógio do sistema (clock) como timestamp, isto é, o timestamp de uma transação é igual a hora em que a transação entra no sistema, ou, usar como um contador lógico que é incrementado sempre que se usa um novo timestamp, isto é, o timestamp da transação é igual ao valor do contador no momento em que a transação entra no sistema.

O protocolo de ordenação por timestamp garante a serialização de conflito. Essa asserção provém do fato de que operações conflitantes são processadas pela ordem do timestamp. O protocolo garante também resistência a deadlocks (conjunto de transações tal que toda transação está esperando outra transação também nela contida), já que uma transação nunca espera. O protocolo consegue gerar escalas que não podem ser recuperadas (desfeitas), entretanto



ele pode receber uma extensão para fazer escalas cascadeadas. (SILBERSCHATZ, KORTH, SUDARSHAN, 2006)

O `TIMESTAMP` possui também mais duas variações de tipo de dados onde esses são `TIMESTAMP WITH TIME ZONE` onde inclui um deslocamento de fuso horário (`TIME ZONE`) em seu valor, e o `TIMESTAMP WITH LOCAL TIME ZONE` que inclui um deslocamento de fuso horário em seu valor diferenciando em que este deslocamento não é armazenado como parte dos dados da coluna, mas normalizando para o fuso horário do banco de dados (`DBTIMEZONE`) (LEGATTI, 2009).

## 4. FERRAMENTAS FLASHBACK

Oracle Flashback fornece um conjunto de novos recursos para visualizar e retroceder para frente ou para trás no tempo. Essas ferramentas oferecem a capacidade de consultar dados históricos, realizar análises, alterar e realizar a auto-reparação do serviço para recuperar corrupções lógicas, para monitoramento de alterações no banco e como recuperação e retorno em períodos de tempo das exclusões de dados sendo de muita utilidade e de alto desempenho de seus usuários

Assim a Oracle introduz o conceito de UNDO (Automatic Undo Management) e com ela uma funcionalidade que permite ao usuário errar e ao DBA recuperar um registro muito mais facilmente do que nas versões anteriores.

Com o flashback, o conceito de UNDO, funciona como o tempo de retenção definido pelo parâmetro UNDO\_RETENTION (em segundos), significa que durante UNDO\_RETENTION, o banco tenta manter as transações mesmo “comitadas” no segmento de UNDO possa utilizá-la para recuperar os registros alterados ou excluídos.

O Flashback esta selecionado em 7 tipos, para ter uma alta gama de funcionalidades, essas ferramentas foram introduzidas para usuários de banco de dados, as ferramentas se destacam como, Flashback Query, Flashback Database, Flashback Drop, Flashback Table, Flashback Transaction Query, Flashback Version Query e Flashback Data Archive. A seguir, são mostradas as 7 ferramentas, bem como suas descrições.

### 4.1. Flashback Query

Essa ferramenta fornece recursos para visualizar dados em um determinado ponto no tempo no passado, especificando uma data, horário ou um SCN (System Change Number – sequenciador interno, que governa o procedimento de recovery) para efetuar uma pesquisa.

Por padrão, operações no banco de dados usam os dados disponíveis mais recentemente “comitados”. Se for necessário pesquisar determinados dados

em algum ponto no passado, será preciso utilizar o recurso de Flashback Query na qual será necessário especificar um “horário” para que este possa efetuar a pesquisa desses dados. (LEGATTI, 2007)

Este recurso é também muito útil caso seja preciso restaurar dados que foram erroneamente deletados ou alterados. Os bancos de dados mais antigos não proporcionavam para seus DBAs, essas funcionalidades de Flashback, dificultando suas restaurações. Dependendo do tamanho do banco de dados, este processo poderia ser pesado e cansativo.

Uma alternativa para a utilização do Flashback Query é o pacote DBMS\_FLASHBACK onde essa opera no nível de sessão, enquanto o Flashback Query opera no nível do objeto.

## 4.2. Flashback Database

Essa ferramenta provê uma alternativa rápida e eficiente para recuperar o banco de dados. Flashback Database retorna rapidamente um banco de dados para um determinado ponto específico do tempo ou passando um SCN (System Change Number), corrigindo assim eventuais problemas causados por dados corrompidos ou erros humanos. É como um botão de “rebobinar” para seu banco de dados. Proporciona o ponto de banco de dados em tempo de recuperação sem a necessidade de um *backup* do banco de dados ser restaurado, tornando assim o tempo de recuperação rápida.

Após uma operação Flashback Database, para ter acesso de gravação ao banco de dados que sofreu flashback, é preciso reabri-lo com um comando “alter database open resetlogs”. Você deve ter o privilégio de sistema SYSDBA para usar o comando flashback database.

Este recurso pode ser utilizado para situações como corrupção de dados lógicos ou perda de dados feita por algum usuário de uma maneira não intencional.

### 4.3. Flashback Drop

Esta ferramenta será usada quando for preciso recuperar uma tabela que foi erroneamente eliminada, sendo assim, seria necessário importar esta tabela de um arquivo de exportação (dump) recente ou realizar uma recuperação incompleta baseada em horário, onde a recuperação é encerrada após todas as alterações feitas até determinado momento terem sido comitadas.

A partir da versão 10g, quando uma tabela é eliminada, o SGBD não remove automaticamente o espaço desta das tablespaces, a não ser que o parâmetro PURGE tenha sido especificado no comando DROP. Na realidade os metadados da tabela, dos índices e das restrições (*constraints*) associadas a esta são renomeados e colocados em uma tabela conhecida como recyclebin, ou simplesmente, lixeira. Desta forma, caso a tabela tenha sido excluída erroneamente é possível recuperá-la posteriormente. Esta funcionalidade é chamada de Flashback Drop e através do comando FLASHBACK TABLE é realizada a recuperação. (SANTOS, 9999).

Esta funcionalidade permite recuperar uma tabela apagada, ou logo se desfazer um comando Drop Table sem precisar ativar o recurso Flashback Database. Para suportar esse recurso o Oracle introduziu o conceito de lixeira que funciona semelhante ao do Windows, sendo nomeado de RECYCLEBIN.

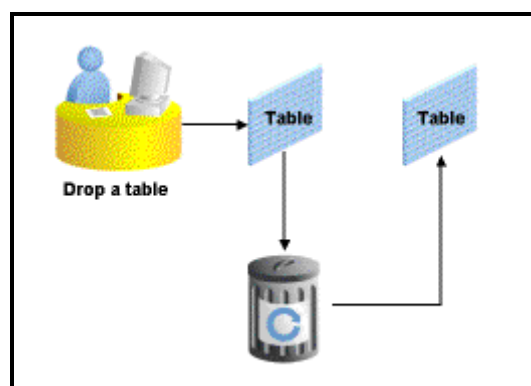


Figura 2. RECYCLEBIN

#### 4.4. Flashback Table

Esta ferramenta retorna uma tabela para um determinado ponto no tempo, especificando uma data ou um SCN (SCN – sequenciador interno, que governa o procedimento de *recovery*), podendo restaurar os dados de uma tabela juntamente com seus índices, gatilhos (triggers) e restrições (constraints), mesmo com banco de dados *on-line*, desfazendo as alterações feitas em apenas uma tabela, por exemplo. Vale lembrar que o Oracle Flashback Table não é capaz de recuperar dados quando há uma corrupção física (por exemplo, um problema de disco ou segmentos de dados ou índices inconsistentes).

Através do Flashback Table, será possível recuperar uma tabela e seus dados de acordo com uma determinada janela de tempo, por exemplo, quero recuperar a tabela CLIENTES e seus dados conforme ela estava na data de 06/02/2006 às 13h20, ou ainda, a um determinado número de alterações do sistema (SCN – System Change Number). Este é um número gerado pelo próprio Oracle para determinar a sincronização de cada “datafile” juntamente com o “controlfile”, ou seja, a cada momento que ocorre um evento Check Point, o Oracle sincroniza todos os “datafiles” e “controlfile” gerando um número. Este identificará que todos os dados estão íntegros e devidamente gravados nos “datafiles” até aquele determinado instante. É possível utilizar este número como argumento para definir o momento do tempo que se quer a tabela recuperada. (SQL Magazine)

#### 4.5. Flashback Transaction Query

Permite o usuário pesquisar, visualizar e retornar os comandos realizados em uma única transação, ou em todas as transações durante um período de tempo consultados dados de linha em pontos específicos de tempo.

Possibilita a reconstrução de instruções SQL previamente executadas no banco.

O Flashback Transaction Query usa uma view de dicionário de dados `FLASHBACK_TRANSACTION_QUERY` para recuperar informações de transações de banco de dados para todas as tabelas envolvidas em uma transação. Detectando uma determinada operação errada, pode-se utilizar o recurso Flashback Transaction Query para identificar alterações que foram feitas pela transação, e solicitar uma modificação para inverter essas mudanças, facilitando assim a análise e a auditoria das transações.

Esta view possui uma coluna `UNDO_SQL` que fornece a sentença SQL que poderá ser utilizada para desfazer uma mudança feita anteriormente (LEGATTI, 2008)

#### 4.6. Flashback Version Query

Permite o usuário visualizar todas as versões de todas as linhas que já existiu em uma ou mais tabelas em um determinado intervalo de tempo específico. Logo caso queira ver dados de um usuário em um determinado tempo em sua base de dados, por exemplo, as mudanças desses dados como código, nome, endereço, telefone, etc.

Só serão mostradas as alterações submetidas a `COMMIT` ou `ROLLBACK`.

O Flashback Versions Query fornece uma maneira simples e fácil de mostrar todas as versões de todas as linhas em uma tabela entre dois SCN's ou tempos de intervalo (`TIMESTAMPS`), informando se as linhas foram inseridas, eliminadas ou atualizadas.

Esta é uma extensão à linguagem SQL que permite ao DBA recuperar diferentes versões de linhas de tabela em qualquer intervalo de tempo. Uma nova versão de registro será criada toda vez que o comando **COMMIT** for emitido, e mesmo que uma linha seja eliminada e reinserida várias vezes, todas estas alterações estarão disponíveis para acesso. (LEGATTI, 2008)

#### 4.7. Flashback Data Archive

Para manter um registro do histórico de todas as alterações feitas nas linhas em tabelas críticas é um procedimento propenso a erros que requer aplicações personalizadas ou gatilhos (triggers) de banco de dados para manter os repositórios de históricos de mudanças. (BRYLA, LONEY, 2008)

Com o Oracle 11g pode-se definir para uma tabela um arquivo que não é mais que um repositório constituído por um ou mais *tablespaces* e que vão guardar toda a informação de UNDO dessa tabela registrando somente instruções de “update” e “delete”, mais não instruções de “insert”.

Basicamente cria-se uma ou mais áreas de repositório, atribuindo um período de retenção padrão para os objetos no repositório e, em seguida, marcando as tabelas apropriadas para rastreamento.

Este é utilizado também para monitorar e manter um histórico das alterações ocorridas nos dados do banco, de forma transparente, segura e eficiente.

## 5. PROPOSTA DO TRABALHO

Em função da complexidade dos *backups*, a melhoria de desempenho e agilidade no processo de desenvolvimento e outros fatores que influenciam em uma boa recuperação de dados, a Oracle passou a oferecer recursos que permitem aos desenvolvedores ou administradores de bancos de dados recuperarem dados, objetos, chaves, atributos, tabelas, sem a necessidade de recorrer aos *backups*.

Devido a erros de usuários operadores, muitos desses erros são extremamente difíceis de evitar e podem ser particularmente difíceis de recuperar de planejamento prévio e sem a tecnologia adequada. Esses erros podem provocar a corrupção de dados e causar danos a um ou mais componentes de uma infra-estrutura de TI, excluindo acidentalmente dados importantes, apagando restrições ou atributos e eliminando tabelas e relacionamentos.

Por razão de todo esse panorama a Oracle Technologies lançou novos recursos para esses reparos, os recursos são disponibilizados pioneiramente no banco de dados Oracle 9i, assim trazendo novas opções com sua evolução.

Este trabalho tem como proposta demonstrar o uso das ferramentas de flashback com base no modelo de dados apresentado.

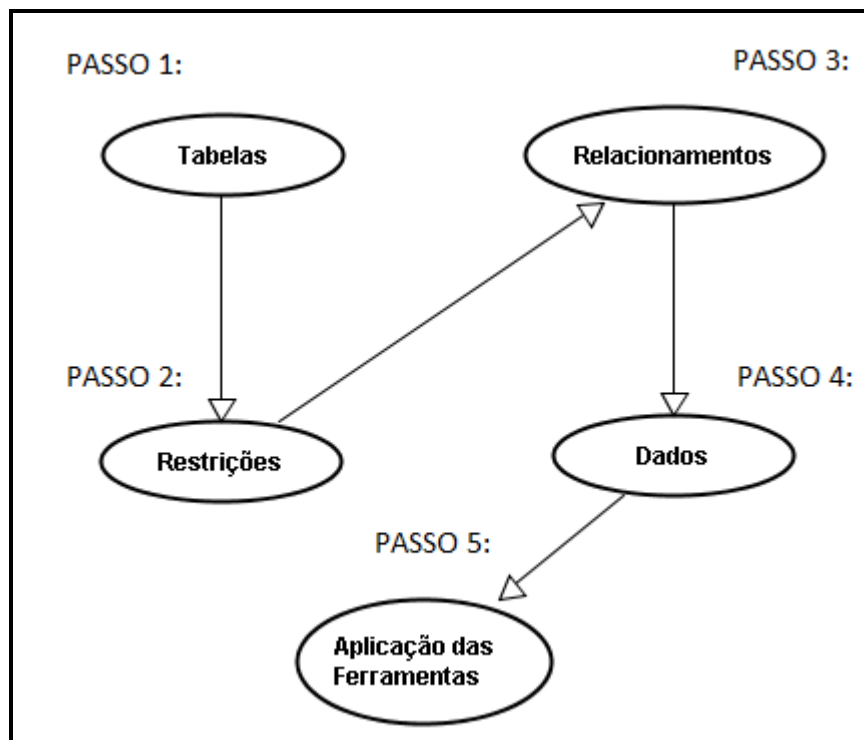
Na Figura 3, é ilustrada esta proposta. Nela são apresentados os passos necessários para o desenvolvimento do trabalho. Esses passos serão responsáveis por apresentar o modelo de dados a ser utilizado no processo de flashback.

Verificando o Passo 1: Tabelas, será mostrado todas as tabelas que serão utilizadas no tópico estudo de caso, no Passo 2: Restrições, será mostrado as restrições que as tabelas definidas possuirão, o Passo 3: mostrará todos os relacionamentos das 8 tabelas constituídas na proposta e por fim o Passo 4: onde definirá os dados que serão usados.

A Modelagem de Dados irá ser usada como representação de alto nível e considerando exclusivamente o ponto de vista do usuário criador do dado.



A Apresentação do Modelo de Dados será uma forma de apresentar todas as estruturas que serão utilizadas para o estudo de caso visando às funcionalidades das ferramentas Flashback.



**Figura 3. Proposta Geral do Trabalho**

### 5.1 Passo 1: Tabelas

Tabela é um conjunto de dados dispostos em número finito de colunas e número ilimitado de linhas. Todos os dados de um Banco de Dados Relacional (BDR) são armazenados em tabelas, cada linha contém um mesmo conjunto de colunas. Em um banco de dados podem existir centenas de tabelas, sendo que o limite pode ser imposto tanto pela ferramenta utilizada, quanto pelos recursos de hardware disponíveis no equipamento.

Colunas são consideradas os *atributos* da tabela, e caracterizam os tipos de dados que deverão constar (numéricos, datas, coordenadas, etc.). O número de linhas pode ser interpretado como o número de combinações de valores dos atributos da tabela, e pode conter linhas idênticas, dependendo do seu objetivo.

A forma de referenciar inequivocamente uma única linha é através da utilização de uma chave primária.

Este passo tem por função mostrar todas as tabelas que serão utilizadas no estudo de caso. Estas tabelas mostrarão as ferramentas e suas funcionalidades uma por uma de maneira que facilite o trabalho dos administradores e a recuperação dos dados, assim como tabelas, atributos, etc.

## **5.2 Passo 2. Restrições**

Restrições é um tipo de limitar os dados que podem ser armazenados na tabela. Para muitas aplicações a restrição obtida não possui o refinamento necessário. Uma restrição é necessária para restringir os dados de uma coluna com relação a outras colunas ou linhas, por exemplo, em uma tabela contendo informações sobre produtos deve haver apenas uma linha para cada código de produto.

Duas das formas mais comuns de restrições são a chave primária e a chave estrangeira, onde a primeira é a propriedade de identificar de forma única uma linha da tabela, logo uma exclusividade da tabela tendo no máximo uma só chave primária em cada tabela. A segunda se refere ao tipo de relacionamento entre tabelas de um banco de dados, onde uma tabela receberá como atributo a chave primária de outra tabela.

Este passo tem por função mostrar todas as restrições de todas as tabelas do projeto, sendo elas chaves primárias ou chaves estrangeiras, facilitando na utilização das ferramentas de recuperações de dados.

## **5.3 Passo 3. Relacionamentos**

Relacionamento do Modelo de Dados é uma associação entre entidades distintas, é por fim a criação de atributos com chaves externas do Modelo Relacional. Isto é feito ligando-se um atributo de uma Tabela A com um atributo

de uma Tabela B por meio de inclusão do atributo chave da Tabela B (conhecido como chave estrangeira ou *foreign key*) da Tabela A.

Estes relacionamentos podem ser divididos em 3 tipos:

- Um para um (**1 para 1**) - indica que as tabelas têm relação única entre si. Pode-se escolher qual tabela vai receber a chave estrangeira;
- Um para muitos (**1 para N**) - a chave primária da tabela que tem o Lado 1 aparece na tabela do Lado N com um atributo comum. No Lado N ela é chamada de chave estrangeira;
- Muitos para muitos (**N para M**) - quando tabelas têm entre si relação n..m, é necessário criar uma nova tabela com as chaves primárias das tabelas envolvidas, ficando assim uma chave composta, ou seja, formada por diversos atributos-chave de outras tabelas.

Este passo tem por função mostrar todos os relacionamentos, simples ou compostos, utilizados em todas as tabelas visualizadas neste trabalho, facilitando o manuseamento das ferramentas Flashbacks.

#### 5.4 Passo 4. Dados

Dados nada mais são que “fatos conhecidos” que podem ser armazenados e que possuem um significado implícito. Os dados e as descrições correspondentes são armazenadas na base e gerenciadas pelo SGBD, onde esses dados nada mais são do que matérias primas para obtenção de informações. (TAKAI, ITALIANO, FERREIRA, 9999)

Por exemplo dados cadastrais de uma pessoa e um determinado banco de dados, onde ela possuirá seu código, nome, endereço, telefone, conta, etc.

Este passo tem por função mostrar os dados propostos para as funcionalidades dos Flashbacks, eles são dados inseridos aleatoriamente sendo eles modificados no futuro se preciso.

## 5.5 Passo 5: Aplicação das Ferramentas

Este passo consiste em uma detalhada descrição das aplicações das ferramentas flashback, mostrando o que cada uma proporciona e, sendo assim, ao final, será mostrada uma breve apresentação das funções e das capacidades que cada uma dessas ferramentas oferece, conforme ilustrado na Figura 4.

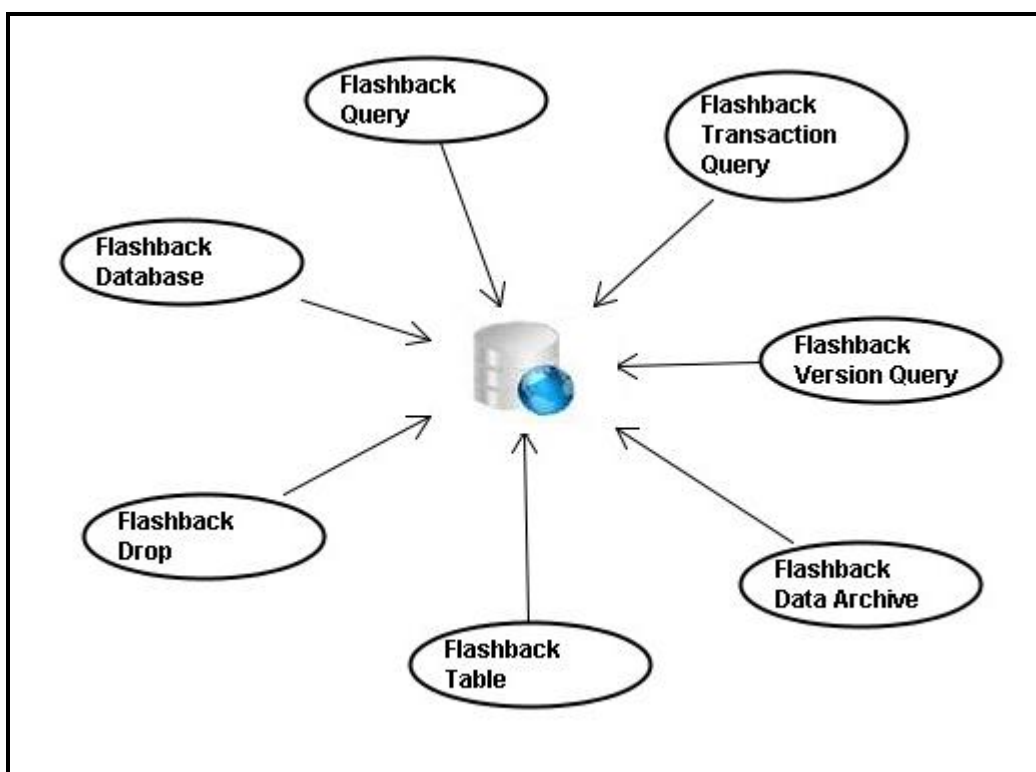


Figura 4. Ferramentas Flashback

## 6. ESTUDO DE CASO

O Modelo de Dados utilizado para o Estudo de Caso é baseado em um portal para as instituições do Município de Assis.

O portal Integraassis foi desenvolvido por alunos da FEMA (Fundação Educacional do Município de Assis), onde este portal integra todas as instituições presentes na cidade de Assis, melhorando a comunicação entre as mesmas e proporcionando um meio comunicação eficaz para as crianças e adolescentes.

O projeto tem como parceria a Instituição Nosso Lar de Assis e apoio da Fundação Telefônica.

### 6.1 Aplicação do Passo 1: Apresentação das Tabelas

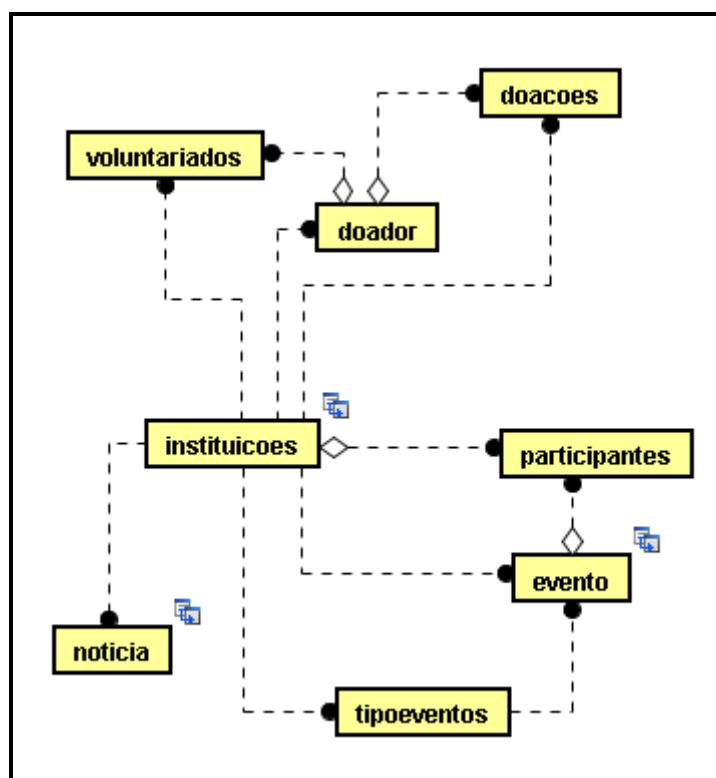


Figura 5. Tabelas Gerais

Conforme mostrado na Figura 5, o estudo de caso deste trabalho é composto de 8 tabelas todas utilizando relacionamentos e restrições. Nas figuras a seguir são apresentados os atributos e os relacionamentos de cada uma das tabelas.

- Instituição;

<b>instituicoes</b>
- <<Primary Key>> codigo : int
- nome : varchar2
- email : varchar2
- endereco : varchar2
- telefone : int
- nomepresidente : varchar2
- objetivos : varchar2
- tipo : char

Figura 6: Tabela instituicoes

- Participante;

<b>participantes</b>
- <<Primary Key>> codigo : int
- <<Foreign Key>> cod_evento : int
- nome : varchar2
- rua : varchar2
- bairro : varchar2
- numero : int
- telefone : int
- <<Foreign Key>> codinstituicao : int

Figura 7: Tabela participantes

- Tipo de Evento

<b>tipoeventos</b>
- <<Primary Key>> codigo : int
- nome : varchar2
- descricao : varchar2

Figura 8: Tabela tipoeventos

- Evento;

<b>evento</b>
- <<Primary Key>> codigo : int
- <<Foreign Key>> cod_tipoevento : int
- titulo : varchar2
- descricao : varchar2
- local : varchar2
- data : varchar2
- horario : varchar2
- <<Foreign Key>> codinstituicao : int

Figura 9: Tabela evento

- Doador;

<b>doador</b>
- <<Primary Key>> codigo : int
- nome : varchar2
- rua : varchar2
- bairro : varchar2
- numero : int
- telefone : int
- email : varchar2
- <<Foreign Key>> codinstituicao : int

Figura 10: Tabela doador

- Doação;

<b>doacoes</b>
- <<Primary Key>> codigo : int
- <<Foreign Key>> cod_doador : int
- descricao : String
- data : String
- <<Foreign Key>> codinstituicao : int

Figura 11: Tabela doacoes

- Noticia;

<b>noticia</b>
- <<PrimaryKey>> codigoNoticia : int
- titulo : String
- descricao : String
- <<ForeignKey>> codinstituicao : int

Figura 12: Tabela noticia

- Voluntariados;

<b>voluntariados</b>
- <<PrimaryKey>> codigo : int
- <<ForeignKey>> cod_doador : int
- disponibilidade : String
- atividade : String
- <<ForeignKey>> codinstituicao : int

Figura 13: Tabela voluntariados

## 6.2 Aplicação do Passo 2: Restrições

As tabelas descritas possuem diferentes restrições limitadas em chaves primárias e chaves estrangeiras. Todas as tabelas, possuem uma restrição em comum com a tabela “instituicoes” (Figura 6), essas tabelas utilizam-se de um atributo chamado “codinstituicao” para este relacionamento facilitando a manutenção e manuseio das tabelas descritas.

Essas restrições serão usadas para mostrar algumas funcionalidades das ferramentas Oracle Flashback, utilizadas para recuperação e monitoramento de determinados tipos de históricos. Abaixo será mostrado as restrições de todas as tabelas presentes no projeto:



**Tabela instituicoes:**

Chave Primaria – codigo:int

Tabela participantes:

Chave Primaria – codigo:int

Chave Estrangeira – cod\_evento:int

Chave Estrangeira – codinstituicao:int

**Tabela evento:**

Chave Primaria – codigo:int

Chave Estrangeira – cod\_tipoevento:int

Chave Estrangeira – codinstituicao:int

**Tabela tipoevento:**

Chave Primaria - codigo:int

**Tabela doador:**

Chave primaria – codigo:int

Chave Estrangeira – codinstituicao:int

Tabela doacoes:

Chave primaria – codigo:int

Chave Estrangeira – cod\_doador:int

Chave primaria – codinstituicao:int

**Tabela noticia:**

Chave primaria – codigo:int

Chave Estrangeira – codinstituicao:int

**Tabela voluntariados:**

Chave primaria – codigo:int

Chave Estrangeira – cod\_doador:int

Chave Estrangeira – codisntituicao:int

### 6.3 Aplicação do Passo 3: Relacionamentos

Neste trabalho todas as tabelas estão relacionadas. O relacionamento focaliza a tabela “instituiçoes” (Figura 6), onde todas as outras terão uma chave estrangeira “codinstituicao” para se comunicar com ela. Logo esse “codinstituicao” passará pelas tabelas “notícia” (Figura 12), “voluntariados” (Figura 13), “participantes” (Figura 7), “evento” (Figura 9), “tipoeventos” (Figura 8), “doador” (Figura 10), “doações” (Figura 11) para que estas retornem atributos da tabela “instituições” (Figura 6). Por exemplo: Quando o usuário cadastrar os atributos da tabela “doador” (Figura 10), automaticamente a chave primária da tabela “instituições” (Figura 6) será guardada, para eventuais ocasiões como retornar um nome ou até o proprio código.

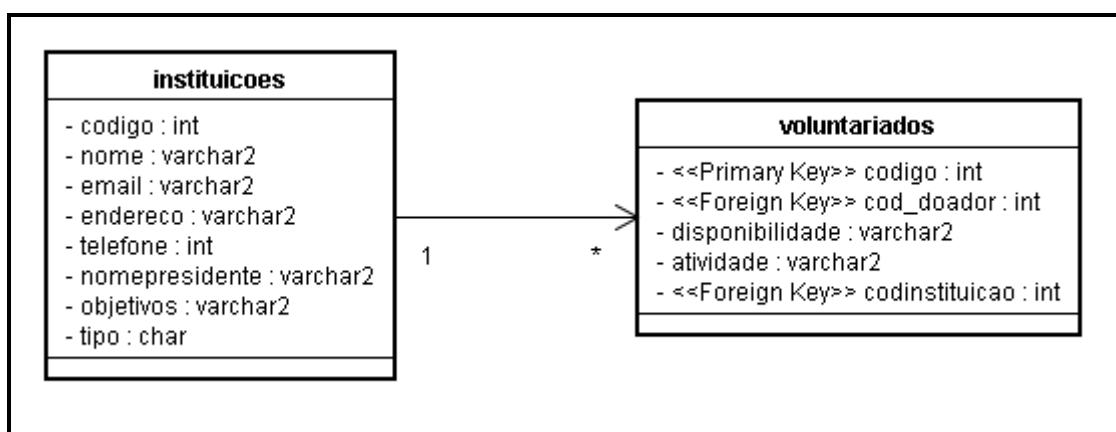


Figura 14: Relacionamento instituiçoes para voluntariados

Por fim essas tabelas descritas a cima, tornam-se relacionais umas com as outras. A chave primária da tabela “doador” (Figura 10), será chave estrangeira na tabela “doações” (Figura 11) formando assim a chave estrangeira “cod\_doador”, assim como acontece na tabela de “voluntariados” (Figura 13) onde esta também terá o atributo “cod\_doador”.

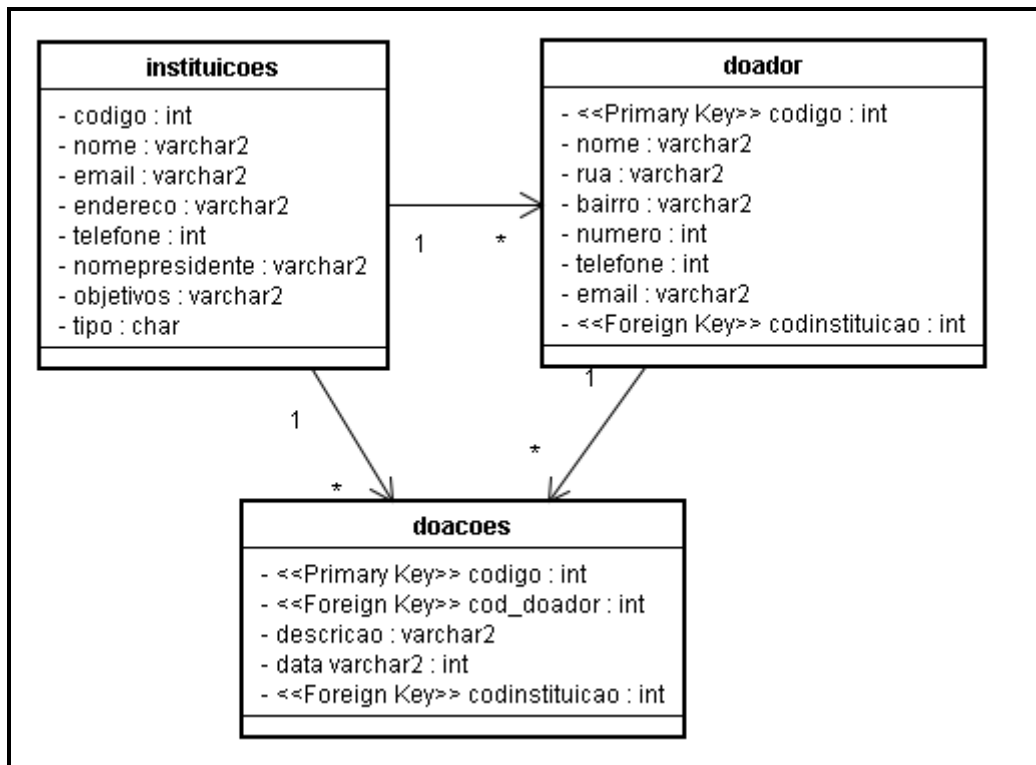


Figura 15: Relacionamento instituicoes, doador, doacoes

Outras tabelas sofrem relacionamentos como “tipoeventos” (Figura 8) para “evento” (Figura 9) e “evento” (Figura 9) para “participantes” (Figura 7), onde evento adicionará um atributo “cod\_tipoeventos” e a tabela “participantes” receberá um atributo “cod\_evento” como chave estrangeira.

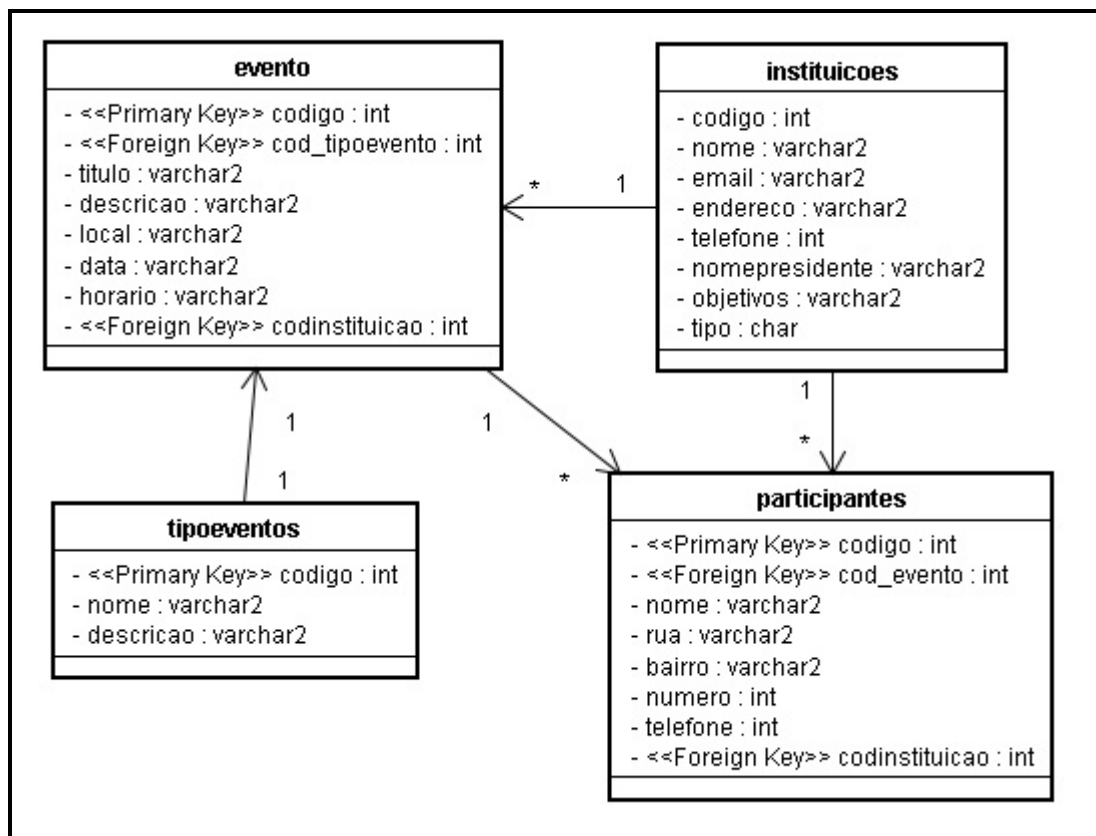


Figura 16: Relacionamento instituicoes, participantes, evento, tipoeventos

Nesses relacionamentos existiram três tabelas onde só ocorrerá o relacionamento com a tabela “instituições” (Figura 6), nesse caso será a unidade “notícia” (Figura 12), “doador” (Figura 10) e “tipoeventos” (Figura 8). Elas não terão nenhum outro relacionamento que acontece por exemplo em “evento” (Figura 9) e “tipoevento” (Figura 8) descrito no parágrafo acima, entretanto assim, só proporcionando para elas o atributo “codinstituição”.

#### 6.4 Aplicação do Passo 4: Mostrando os Dados

Os dados a seguir mostrados neste passo, serão para a utilização das ferramentas Flashback, são dados aleatórios com o intuito de serem modificados se forem precisos.

## Tabela Instituições:

codigo – 1

nome – Apae

email – [apae@teste.com](mailto:apae@teste.com)

endereco – end1

telefone – 0000000

nomepresidente – José

codigo – 2

nome – Casa das Crianças

email – [Crianças@teste.com](mailto:Crianças@teste.com)

endereco – end2

telefone – 0000000

nomepresidente - Francisco

codigo – 3

nome – Broto Verde

email – [BrotoVerde@teste.com](mailto:BrotoVerde@teste.com)

endereco – end3

telefone – 0000000

nomepresidente – Bruno

## Tabela Notícia:

codigo – 1

titulo – Apresentação dos Trabalhos

descricao – desc1

codinstituicao – 1

codigo – 2

titulo – Festas Finais

descricao – desc2

codinstituicao – 2

codigo – 3

titulo – Titulo3

descricao – desc3

codinstituicao - 1

Tabela TipoEventos:

codigo – 1

nome – Evento Beneficiente

descricao – Evento beneficente para instituições

codigo – 2

nome – Evento Festivo

descricao – Evento voltado para festas

Tabela Evento:

codigo – 1

cod\_tipoevento – 2

titulo – Festa Junina

descricao – Festa de integração das instituições

horario – sexta-feira, 21 horas

codinstituicao – 1

codigo – 2

cod\_tipoevento – 1

titulo – Festa do Milho

descricao – Festa para arrecadas fundos

horario – sabado, 21 horas

codinstituicao - 1

Tabela Doador:

codigo – 1

nome – doador1

rua – TesteRua1

bairro – Testebairro1

numero – 11

telefone – 0000000

codinstituicao – 1

codigo – 2

nome – doador2

rua – Testesua2

bairro – Testebairro2

numero – 11

telefone – 0000000

codinstituicao - 2

Tabela Doações:

codigo – 1

cod\_doador – 1

descricao – Descrição da doação1

data – 12/12/2009

codinstituicao – 1

codigo – 2

cod\_doador – 1

descricao – Descrição da doação2

data – 24/11/2009

codinstituicao – 2

codigo – 3

cod\_doador – 1

descricao – Descrição da doação3

data – 12/1/2009

codinstituicao - 1

Tabela Participantes

codigo – 1

cod\_doador – 1

nome – José Henrique

rua – Osório

bairro – Vila Claudia

numero – 180

telefone – 3333333

codinstituicao – 1

codigo – 2

cod\_doador – 2

nome – Fabricio Carlos

rua – Rua Vicente

bairro – Vila Maria

numero – 350

telefone – 33228484

codinstituicao – 2

Tabela Voluntariados:

codigo – 1

cod\_doador – 1

disponibilidade – Todos os dias

atividade – estágio

codinstituicao - 1

codigo – 2

cod\_doador – 2

disponibilidade – Segundas e Sextas

atividade – Analista

codinstituicao - 2



## 6.5 Aplicação do Passo 5: Usando as Ferramentas

### 6.5.1 Flashback Query

Neste item serão mostrados dois tipos básicos de utilização da ferramenta Flashback Query.

Primeiramente serão criadas algumas tabelas, e após serão efetuados os comandos da ferramenta. Estas tabelas serão as seguintes:

#### - instituicoes

```
<sql>create table instituicoes (  
codigo number(8) not null, nome varchar2(25), email varchar2(25),  
endereco varchar2(25), telefone varchar2(25), nomepresidente varchar2(25),  
primary key (codigo));  
  
<sql>insert into instituicoes values (1, 'APAE', 'apae@teste.com', 'endereco1',  
'0000000', 'José');  
  
<sql>insert into instituicoes values (2, 'Casa das Crianças',  
'Criancas@teste.com', 'end2', 0000000, Francisco);  
  
<sql>insert into instituicoes values (3, 'Broto Verde', 'BrotoVerde@teste.com',  
'end3', '0000000', 'Bruno');
```

Figura 17: Criação e Inserção da Tabela Instituições

#### - noticia;

```
<sql>create table noticia (  
codigo number(8) not null, titulo varchar2(25),  
descricao varchar2(25), codinstituicao number(8),  
primary key (codigo),
```

```
foreign key (codinstituicao) references instituicoes (codigo));
<sql>Insert into noticia values (1, 'Apresentação do Pic', 'descricao1', 1);
<sql>Insert into noticia values (2, 'Festas Finais', 'desc2', 2);
<sql>insert into noticia values (3, 'Titulo3', 'desc3', 1);
```

Figura 18: Criação e Inserção da Tabela Notícia

Utilizando um select para verificar a tabela antes das alterações;

```
<sql>select * from noticia;
```

CODIGO	TITULO	DESCRICAO	CODINSTITUICAO
1	Apresentação do Pic	descricao1	1
2	Festas Finais	Desc2	2
3	Titulo3	Desc3	1

Tabela 1: Tabela de Notícia

Agora serão realizados alguns “updates” para que o Flashback Query visualize as modificações:

```
<sql>update noticia set titulo = 'Confraternização' where codigo = 1;
```

```
<sql>update noticia set titulo = 'Apresentação dos Trabalhos' where codigo = 2;
```

```
<sql>delete from noticia where codigo = 3;
```

```
<sql>select * from noticia;
```

CODIGO	TITULO	DESCRICAO	CODINSTITUICAO
1	Confraternização	descricao1	1
2	Apresentação dos Trabalhos	Desc2	2

Tabela 2: Tabela de Notícia alterada

Agora é possível visualizar os dados da tabela, que existiam antes de utilizar o update, em um intervalo de tempo qualquer escolhido. Neste caso será utilizado um intervalo de 2 minutos:

```
<sql>select * from noticia
```

```
as of timestamp (systimestamp – interval '2' minute);
```

CODIGO	TITULO	DESCRICAO	CODINSTITUICAO
1	Apresentação do Pic	descricao1	1
2	Festas Finais	Desc2	2
3	Titulo3	Desc3	1

Tabela 3: Tabela de Notícia recuperada

O Flashback Query será acionado com a cláusula “AS OF” forçando o Oracle visualizar os dados antes do “update” em um intervalo de 3 minutos.

Outra forma de se utilizar o Flashback Query é usando o pacote DBMS\_FLASHBACK, no qual o mesmo opera em um nível de sessão.

Para isso, será necessário dar o privilégio de EXECUTE no pacote para um determinado usuário.

```
<sql> connect / as sysdba;
```

Disponibilizando o acesso ao pacote DBMS\_FLASHBACK;

```
<sql> grant EXECUTE on dbms_flashback to bonan;
```

Conectando com o usuário “bonan” para efetuar as SQLs;

```
<sql> connect bonan/bonan;
```

Selecionando os dados da tabela instituições.

```
<sql>select * from instituicoes;
```

CODIGO	NOME	EMAIL	ENDERECO	TELEFONE	NOMEPRESIDENTE
1	Apae	<a href="mailto:apae@teste.com">apae@teste.com</a>	End1	0000000	José
2	Casa das Crianças	<a href="mailto:Criancas@teste.com">Criancas@teste.com</a>	End2	0000000	Francisco
3	Broto Verde	<a href="mailto:BrotoVerde@teste.com">BrotoVerde@teste.com</a>	End3	0000000	Bruno

Tabela 4: Tabela de Instituições

Verificando os dados da tabela ‘instituições’, o usuário efetuara algumas transações;

Atualizando as linhas inseridas para a utilização do DBMS\_FLASHBACK;

```
<sql>update instituicoes set telefone = 33223322, nomepresidente = 'Francisco Augusto' where codigo = 2;
```

```
<sql>update instituicoes set nome = 'Nosso Lar', nomepresidente = 'Caio Cesar' where codigo = 1;
```

```
<sql>select * from instituicoes;
```

CODIGO	NOME	EMAIL	ENDereco	TELEFONE	NOMEPRESIDENTE
1	Nosso Lar	<a href="mailto:Apae@teste.com">Apae@teste.com</a>	End1	0000000	Caio Cesar
2	Casa das Crianças	<a href="mailto:Crianca@teste.com">Crianca@teste.com</a>	End2	33223322	Francisco Augusto
3	Broto Verde	<a href="mailto:BrotoVerde@teste.com">BrotoVerde@teste.com</a>	End3	0000000	Bruno

Tabela 5: Tabela de Instituições alterada

Após Atualizar as linhas da tabela instituições, será preciso comitar essas atualizações para poder utilizar o pacote dbms\_flashback:

```
<sql> commit;
```

```
<sql> execute dbms_flashback.enable_at_time(to_timestamp(sysdate – interval '2' minute));
```

“Procedimento PL/SQL concluído com sucesso.”

Em algum imprevisto, o usuário de banco de dados atualizou as linhas por engano, então através do dbms\_flashback é possível visualizar os dados que foram atualizados antes do update;

```
<sql>select * from instituicoes;
```

CODIGO	NOME	EMAIL	ENDereco	TELEFONE	NOME PRESIDENTE
1	Apae	<a href="mailto:Apae@teste.com">Apae@teste.com</a>	End1	0000000	José
2	Casa das Crianças	<a href="mailto:Crianca@teste.com">Crianca@teste.com</a>	End2	0000000	Francisco
3	Broto Verde	<a href="mailto:BrotoVerde@teste.com">BrotoVerde@teste.com</a>	End3	0000000	Bruno

Tabela 6: Tabela de Instituições recuperada

Assim selecionando as linhas antes de serem atualizadas, da tabela “instituições” para visualização.

### 6.5.2 Flashback Database

Antes de usar a ferramenta, é preciso conectar com o privilégio de SYSDBA para usar o comando flashback database. O banco de dados deve ser colocado no modo FLASHBACK com o comando **alter database flashback on**. Logo o banco de dados não poderá estar aberto, sendo assim, realiza-se um shutdown para que isso não ocorra.

```
<sql> connect / as sysdba;
```

```
<sql> shutdown;
```

Banco de dados fechado.

Banco de dados desmontado.

Instância ORACLE iniciado.

```
<sql> startup mount exclusive;
```

```
<sql> alter database archivelog;
```

Banco de dados alterado.

```
<sql>alter database flashback on;
```

Banco de dados alterado.

```
<sql>alter database open;
```

Banco de dados alterado.

Para que o flashback entre no modo **on**, é preciso ativar a recuperação de mídia por meio do comando “alter database archivelog”.

Feito isto, verificar se o flashback foi ativado para o banco de dados, consultando a visão v\$database.

```
<sql> select current_scn, flashback_on from v$database;
```

```
CURRENT_SCN FLASHBACK_ON
```

```
-----
```

```
3347169 YES
```

Com o SCN (System Change Number) encontrado, faça algumas alterações no seu banco de dados.

```
<sql>drop table noticia;
```

```
<sql>drop table tipoeventos;
```

Após as alterações de um shutdown e monte o banco novamente.

```
<sql>shutdown;
```

```
<sql> startup mount exclusive;
```

```
<sql>flashback database to scn 3347269;
```

Abra o banco de dados no modo read only ;

```
<sql>alter database open read only;
```

Verifique se é nesse ponto que gostaria de voltar, em seguida abra o banco com “resetlogs” e feche o “flashback database”;

```
<sql> select table_name from user_tables;
```

TABLE_NAME
INSTITUICOES
NOTICIA
DOADOR
DOACOES
PARTICIPANTE
TIPOEVENTOS
EVENTO
VOLUNTARIADOS

Tabela 7: Banco de Dados recuperado

```
<sql>alter database open resetlogs;
```

```
<sql>alter database flashback off;
```

Quando for executado o comando flashback database, o Oracle verifica se todos os arquivos de REDO LOG arquivados necessários estão disponíveis. Se os *logs* estiverem disponíveis, os arquivos de dados **online** são revertidos ao tempo ou SCN especificados.



Se não houver dados *online* suficientes nos *logs* de arquivamento e na área de flashback, será preciso usar os métodos tradicionais de recuperação de banco de dados para recuperar os dados. (BRYLA, LONEY, 2008)

### 6.5.3 Flashback Table e Flashback Drop

O Flashback Table será usado para recuperar tabelas a partir de um TIMESTAMP ou SCN. Primeiramente cria-se as tabelas:

```
<sql>create table tipoeventos (  
codigo number(8) not null, nome varchar2(35), descricao varchar(35),  
primary key (codigo));  
  
<sql>insert into tipoeventos values (1, 'Evento Beneficiente', 'Evento  
beneficiente para instituições');  
  
<sql>insert into tipoeventos values (2, 'Evento Festivo', 'Evento voltado para  
festas');
```

Figura 19: Criação da tabela tipoeventos

```
<sql>create table evento (  
codigo number(8) not null, cod_tipoevento number(8), titulo varchar2(35),  
descricao varchar2(35), horario varchar2(25), codinstituicao number(8),  
primary key (codigo),  
foreign key (cod_tipoevento) references tipoeventos(codigo),  
foreign key (codinstituicao) references instituicoes (codigo));
```

```
<sql>insert into evento values (1, 2, 'Festa Junina', 'Festa de integração das
instituições', 'sexta-feira', '21 horas', 1);

<sql>insert into evento values (2, 1, 'Festa do Milho', 'Festa para arrecadar
fundos', 'sabado', '21 horas', 1);
```

Figura 20: Criação da tabela evento

Antes de usar a ferramenta em uma ou mais tabelas, é preciso ativar o movimento de linhas (row movement).

```
<sql> alter table tipoeventos enable row movement;
```

Agora, a tabela “evento” é eliminada e em seguida é recuperada com o uso da ferramenta Flashback Table;

```
<sql> drop table evento;
```

```
<sql> select * from evento;
```

SQL Error: ORA-00942: a tabela ou view não existe

Recuperando a tabela “evento” por intermédio de uma forma mais simplificada:

```
<sql>flashback table evento to before drop;
```

Flashback concluído.

```
<sql>select table_name from user_tables;
```

```
TABLE_NAME
```

```
-----
```

```
EVENTO
```

Ou utilizando um “interval”:

flashback table noticia to timestamp (systimestamp – interval '15' minute);

Por intermédio desses comandos é possível recuperar não só tabelas ou linhas, mas também, índices, *triggers* e *constraints*, com muita facilidade.

#### 6.5.4 Flashback Version Query

Será mostrado com essa ferramenta, históricos inteiros de uma determinada linha entre dois SCNs ou TIMESTAMP.

Abaixo a view da ferramenta Flashback Version Query:

PseudoColunas	Descrição
VERSIONS_START(SCN TIME)	O SCN ou timestamp inicial de quando a alteração foi feita na linha
VERSION_END(SCN TIME)	O SCN ou timestamp final de quando a alteração não estava mais válida para a linha. Se esse valor for NULL , a versão da linha ainda é atual ou a linha foi excluída.
VERSIONS_XID	O ID da transação que criou a versão da linha.
VERSIONS_OPERATION	A operação executada na linha (I=Insert, D=Delete, U=Update).

```

<sql> create table doador (
codigo number(8) not null, nome varchar2(25), rua varchar2(25),
bairro varchar2(25), numero number(8), telefone number(8),
codinstituicao number(8),
primary key (codigo),
foreign key (codinstituicao) references instituicoes(codigo));

```

Figura 21: Criação da Tabela Doador

```

<sql> create table doacoes (
codigo number(8) not null, cod_doador number(8), descricao varchar2(25),
data varchar2(25), codinstituicao number(8),
primary key (codigo),
foreign key (cod_doador) references doador (codigo),
foreign key (codinstituicao) references instituicoes(codigo));

```

Figura 22: Criação da Tabela Doações

Inserindo dados e verificando o SCN(System Change Number) para utilizar com o Flashback:

```

<sql> select dbms_flashback.get_system_change_number from dual;

```

```

GET_SYSTEM_CHANGE_NUMBER

```

```

-----

```

```

4136683

```

```

<sql>insert into doador values (1, 'doador1', 'Testerua1', 'Testebairro1', 11,
3333333, 1);

```

```
<sql>insert into doador values (2, 'doador2', 'Testerua2', 'Testebairro2', 11,
4444444, 2);
```

```
<sql>insert into doador values (3, 'doador3', 'Testerua3', 'Testebairro3', 11,
5555555, 2);
```

```
<sql>commit;
```

```
<sql> select dbms_flashback.get_system_change_number from dual;
```

```
GET_SYSTEM_CHANGE_NUMBER
```

```
-----
```

```
4136866
```

Depois das transações feitas e os SCN guardados verificam-se as seguintes versões, serão passados só alguns dados para não visualizar uma tabela muito extensa:

```
<sql>select versions_startscn startscn, versions_xid xid, versions_operation
oper, codigo cod, nome nome, rua rua, bairro bairro, telefone tel from doador
versions between scn 'SCNINICIAL' and 'SCNFINAL';
```

SSCN	XID	OPER	COD	NOME	RUA	BAIRRO	TEL
4136861	03001f00f f060000	I	3	Doador3	Testerua3	Testebairro3	5555555
4136861	03001f00f f060000	I	2	Doador2	Testerua2	Testebairro2	4444444
4136861	03001f00f f060000	I	1	Doador1	Testerua1	Testebairro1	3333333

Tabela 8: Tabela doador utilizando Flashback Version Query

Por intermédio desse “select” o Flashback irá mostrar todas as transações feitas entre o SCN inicial e o SCN final, no caso os “inserts” na tabela doador.

Flashback Version Query com TIMESTAMP;

Inserindo dados na tabela “doações”:

```
<sql> insert into doacoes values (1, 1, 'Descrição da doação1', '12/12/2009', 1);
```

```
<sql> insert into doacoes values (2, 1, 'Descrição da doação2', '24/11/2009', 2);
```

```
<sql> insert into doacoes values (3, 1, 'Descrição da doação3', '12/1/2009', 1);
```

```
<sql>update doacoes set descricao = 'Doação para instituição' data = '10/04/2010' where codigo = 3;
```

```
<sql>select versions_starttime stime, versions_endtime endtime,
```

```
versions_xid xid,
```

```
case
```

```
when versions_operation = 'I' then 'INSERT'
```

```
when versions_operation = 'U' then 'UPDATE'
```

```
end as operation,
```

```
codigo cod, cod_doador coddoador,
```

```
descricao desc, data data from doacoes versions between timestamp minvalue and maxvalue;
```

STIME	ETIME	XID	OPERAT	COD	CODDOADOR	DESC	DATA
29/10/09 09:44:50		01001200 D8050000	UPDATE	3	1	Doação para instituição	10/04/2010
29/10/09 09:40:17		08001E00 F2060000	INSERT	3	1	Descrição da doação3	12/01/2009
29/10/09 09:40:17	29/10/09 09:44:50	08001E00 F2060000	INSERT	2	1	Descrição da doação2	24/11/2009

29/10/09 09:40:17		08001E00 F2060000	INSERT	1	1	Descrição da doação1	12/12/2009
----------------------	--	----------------------	--------	---	---	-------------------------	------------

Tabela 9: Tabela doações utilizando Flashback Version Query

Por meio desse “select” o Flashback irá mostrar todas as transações feitas entre o TIMESTAMP mínimo ‘minvalue’ e o máximo ‘maxvalue’, no caso os “inserts” na tabela doacoes.

### 6.5.5 Flashback Transaction Query

Abaixo alguns “pseudonomes” da ferramenta Flashback Transaction Query:

Nome	Descrição
XID	Identificador da transação.
START_SCN	Número do SCN que inicia a transação.
START_TIMESTAMP	Horário de início da transação.
COMMIT_SCN	SCN gerado durante o COMMIT da transação.
COMMIT_TIMESTAMP	Horário do COMMIT.
UNDO_SQL	SQL que poderá ser usado para desfazer a operação DML realizada.

Utilizando agora as tabelas e as transações feitas no Flashback Version Query mostrados anteriormente.

Uma forma muito simplificada de se usar o Transaction Query é selecionando os DML's com o uso do UNDO\_SQL e passando o XID de uma determinada transação.

```
<sql>select undo_sql from flashback_transaction_query where xid = 'xid selecionado';
```

O resultado será o comando necessário que deverá ser utilizado para desfazer as alterações realizadas.

É mostrado o histórico de todos os comandos DML's feitos na tabela "doador" e "doações".

```
<sql>select to_char(commit_timestamp, 'hh24:mm:ss')time, logon_user, operation, undo_sql from flashback_transaction_query where table_name = 'doador';
```

Com esses "pseudonomes", como LOGON\_USER, pode-se ver o usuário que fez uma determinada transação. Por fim, verificasse o UNDO\_SQL, que mostrará as SQLs que serão usadas para reverter as transações que um usuário fez e que são indesejadas.

Outro método para usar a ferramenta é passando o XID (número do ID da transação).

```
<sql>select [pseudonomes] from flashback_transaction_query where XID = hextoraw ('XID BUSCADO ATRAVES DO VERSION_QUERY');
```

### 6.5.6 Flashback Data Archive

```
<sql>create table participantes (
codigo number(8), cod_doador number(8), nome varchar2(25),
rua varchar2(35), bairro varchar2(35), numero number(8), telefone number(8),
codinstituicao number(8),
primary key (codigo),
```



```
foreign key (cod_doador) references doador (codigo),
foreign key (codinstituicao) references instituicoes (codigo));

<sql>insert into participantes values (1, 1, 'José Henrique', 'Osório', 'Vila
Claudia', 180, 33333333, 1);
<sql>insert into participantes values (2, 2, 'Fabricio Carlos', 'Rua Vicente', 'Vila
Maria', 350, 33228484, 2);
```

Figura 23: Criação e Inserção da tabela Participantes

```
<sql>create table voluntariados (
codigo number(8) not null, cod_doador number(8),
disponibilidade varchar2(30), atividade varchar2(30),
codinstituicao number(8),
primary key (codigo),
foreign key (codinstituicao) references instituicoes (codigo),
foreign key (cod_doador) references doador (codigo));

<sql>insert into voluntariados values (1, 1, 'Todos os dias', 'Estágio', 1);
<sql>insert into voluntariados values (2, 2, 'Segundas e Sextas', 'Analista', 2);
<sql>insert into voluntariados values (3, 1, 'Terças', 'Programador', 3);
```

Figura 23: Criação e Inserção da tabela Voluntariados

Primeiramente para se trabalhar com Flashback Data Archive é necessário criar um ou mais tablespaces onde esses Data Archives serão armazenados automaticamente.

```
<sql> create tablespace tspace1 datafile 'diretório onde será armazenado\nome
do tablespace .dbf' size 150m;
```

Tablespace criado.

Em seguida, criam-se agora dois exemplos de Flashback Data Archive;

```
<sql>create flashback archive fa_participantes tablespace tspace1 retention 10
year;
```

```
<sql>create flashback archive fa_voluntariados tablespace tspace1 quota 500m
retention 7 year;
```

O primeiro flashback terá uma retenção de dados por 10 anos, enquanto o segundo terá uma retenção de 7 anos com uma cota de dados de 500m.

Por padrão só se pode especificar um tablespace no comando create flashback archive, tendo então, que utilizar o comando alter flashback archive para adicionar outro tablespace.

Portanto, após o exemplo de criação de um Flashback Data Archive utilizaremos os flashbacks archives criados para fazer a atribuição as tabelas desejadas;

Como já se tem as tabelas criadas, será utilizada a cláusula alter table.

```
<sql>alter table bonan.participantes flashback archive fa_participantes;
```

Tabela Alterada.

```
<sql>alter table bonan.voluntariados flashback archive fa_voluntariados;
```

Tabela Alterada.

Para verificação das tabelas e flashbacks;

```
<sql>select * from dba_flashback_archive_tables;
```

TABLE_NAME	OWNER_TABLE	FLASHBACK_ARCHIVE_NAME
PARTICIPANTES	BONAN	FA_PARTICIPANTES

Tabela 10: Verificação do Flashback Data Archive

Usando este select, será mostrado todas as tabelas atribuídas ao flashback archive, assim como usuário e os flashbacks com seus tablespaces, neste caso será mostrada somente a tabela “participantes”.

Antes das atualizações será feito um select para verificação dos dados;

```
<sql>select * from participantes;
```

CODIGO	COD_EV ENTO	NOME	RUA	BAIRRO	NUMERO	TELEFONE	CODINSTI TUICAO
1	1	José Henrique	Osório	Vila Claudia	180	3333333	1
2	2	Fabricio Carlos	Rua Vicente	Vila Maria	350	33228484	2

Tabela 11: Tabela participantes

Assim cada registro eliminado ou atualizado destas duas tabelas será armazenado nos Flashback Data Archive criados, protegendo assim os dados que podem ter sido modificados por engano. Por exemplo, se um registro for eliminado da tabela “participantes”;

```
<sql> delete from bonan.participantes where codigo = 1;
```

1 linha deletada.

Por algum erro de usuário, a linha eliminada tinha que ser usada para determinadas transações, então os dados são procurados no flashback archive atribuído a tabela;

```
<sql> select codigo cod, cod_evento codeven, nome, rua, bairro, numero num,
telefone tel, codinstituicao codinst from participantes as of
timestamp(sysimestamp – interval '2' min);
```

COD	CODEVEN	NOME	RUA	BAIRRO	NUM	TEL	CODINST
1	1	José Henrique	Osório	Vila Claudia	180	3333333	1
2	1	Fabricio Carlos	Rua Vicente	Vila Maria	350	33228484	2

Tabela 12: Tabela participantes recuperada

Verificando os dados perdidos pode-se recuperar ou apenas visualizar esses dados.

## 7. CONCLUSÃO

Os backup's de bancos de dados são de primordial importância para as empresas. Manter históricos de dados e de todos os comandos DML's ocorridos nos database poderá ajudar, e facilitar o trabalho de cada administrador e desenvolvedor. Para isto, cada dia mais os usuários estão estudando novas formas de recuperar esses dados, assim como os Aspectos de Tempo com as ferramentas Flashbacks.

A utilização das ferramentas de Flashback está crescendo a cada dia, principalmente entre os usuários de Banco de Dados. As facilidades de manuseio, velocidade de recuperações, visualizações de dados e visualizações de históricos levam os DBA's a usufruírem dessas ferramentas.

Cada Flashback é de grande importância, elas tem suas funcionalidades desenvolvidas e aperfeiçoadas para melhor usabilidade como o Flashback Query quer por sua vez retorna visualizações dos dados, Flashback Table e Drop, que retornam as tabelas deletadas por algum engano, Flashback Database quer por sua vez volta a sua base de dados para o estado original, Flashback Version Query quer permite visualizar todas as versões de todas as linhas feitas, o Flashback Transaction Query que mostra as sql's que deveram ser feitas para retornar ao estado anterior das transações e o Flashback Data Archive que proporciona um repositório para os dados que são alterados.

A utilização dos Flashbacks citados está crescendo e por fim fazendo com que a Oracle Technologies atualize e desenvolva mais e mais ferramentas de recuperação, fazendo com que os problemas ocorridos nas empresas e as perdas dos dados, fiquem menos constantes ou podemos dizer de fácil retorno.

Esse trabalho de conclusão proporcionou algumas das diversas funcionalidades do Flashback, podendo ser usadas em determinadas tarefas de recuperações, retornos através de tempos e visualizações. Entretanto essas ferramentas contêm uma ampla gama para suprir inúmeros problemas e "salvar" os seus muitos utilizadores, assim, as pesquisas para se chegar a um ponto extremo de recuperação de dados, é a meta para que se possa utilizar por completo o Flashback.

## 8. REFERÊNCIAS

BRYLA, Bob; LONEY, Kevin. Oracle Database 11g Manual do DBA.

OSTERNE, Renato. FlashBack – Novos Recursos  
[http://imasters.uol.com.br/artigo/4884/oracle/flashback - novos recursos](http://imasters.uol.com.br/artigo/4884/oracle/flashback_-_novos_recursos).

LEGATTI, Eduardo. Utilizando Flashback Query no Oracle 9i – Belo Horizonte, MG  
<http://eduardolegatti.blogspot.com/2007/05/utilizando-flashback-query-no-oracle-9i.html?showComment=1234209420000>.

LEGATTI, Eduardo. Trabalhando com Flashback Drop no Oracle 10g – Belo Horizonte, MG <http://eduardolegatti.blogspot.com/2007/06/trabalhando-com-flashback-drop-no.html>

LEGATTI, Eduardo. Belo Horizonte – MG  
<http://eduardolegatti.blogspot.com/2008/07/um-pouco-do-flashback-versions-query-e.html>

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistemas de Banco de Dados. Tradução de Daniel Vieira. 5ª edição – Rio de Janeiro : Elsevier, 2006.

O.K. Takai, I.C. Italiano, J.E. Ferreira, Introdução a Banco de Dados, Fevereiro 2005.

ELMASRI, Ramez; NAVATHE, Shamkant B. Sistema de Banco de Dados. Revisor técnico Luíz Ricardo de Figueiredo. São Paulo: Pearson Addison Wesley, 2005.

SQL Magazine 29 - Flashback Drop no Oracle 10g Dos Santos, V. Flashback Drop  
<http://www.devmedia.com.br/articles/viewcomp.asp?comp=7895>

Legatti E. Introdução ao tipo de dados TIMESTAMP e suas variações, 2009  
<http://eduardolegatti.blogspot.com/2009/08/introducao-ao-tipo-de-dado-timestamp-e.html>

Oracle Database 11g. Oracle Flashback Technology  
[http://www.oracle.com/technology/deploy/availability/htdocs/Flashback\\_Overview.htm](http://www.oracle.com/technology/deploy/availability/htdocs/Flashback_Overview.htm)

Angonese S. F.; Edelweiss N. Gerenciador Temporal de Versões de Esquemas, (30 – 31 de Agosto, 1999)

INTEGR@ASSIS Descrição do Projeto Integr@Assis, Autor: Sociedade Filantrópica “Nosso Lar”., 2007.