



UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

Tópicos Avançados de Bases de Dados

Relatório sobre Bases de Dados Relacionais

André Figueira m37280
João Santos l30558

Introdução:

Neste trabalho foi implementado um Sistema de Gestão de Histórico de PostgreSQL. O PostgreSQL é um software open source com um modelo de dados relacional. Tem mais de 15 anos de desenvolvimento e uma arquitectura com provas dadas e tem ganho uma forte reputação pela fiabilidade, integridade dos dados e correcção dos mesmos.

As bases de dados relacionais apresentam dados como relações e providenciam operadores relacionais para manipular os dados na forma de tabela. O Sistema de Histórico de bases de dados é um mecanismo que permite pesquisar a história das alterações, updates e deletes de uma base de dados particular.

Relatório sobre Sistema de Histórico:

Estes módulos foram implementados de forma dinâmica, sendo assim, podem ser introduzidos em qualquer Base de Dados postgresQL, não sendo limitada a uma base de dados. Se houverem alterações na base de dados, como por exemplo, criação de novas tabelas, basta correr novamente os módulos e ficam disponibilizadas as funcionalidades para essas novas tabelas, deixando intactas as já existentes.

Foi utilizada uma base de dados de teste que serviu para demonstrar as funções que se criaram. Esta base de dados consiste na representação de um sistema de informação de uma biblioteca e para isso foram criadas 3 tabelas, books, borrower e

loan.Books terá como atributos o **isbn** de um livro, o **título** de um livro e a **categoria**, sendo o **isbn** a chave primária da tabela.

A tabela borrower terá o **borrower_id** e o nome do requisitador do livro - **name**, **borrower_id** será a chave primária. A tabela loan terá o **borrower_id**, o **isbn** e o **return_date** que é a data de devolução de um livro, sendo a chave primária composta por **borrower_id** e **isbn**. Finalizando, a tabela loan terá duas chaves estrangeiras, o **isbn** e o **borrower_id**.

Dos quatro módulos apresentados, foram escolhidas três. Estes são:

1. Triggers para UPDATE, DELETE e INSERT
2. Recuperação do histórico de determinado tuplo
3. Obtenção da situação de tuplos num momento de tempo

A base de dados utilizada é corresponde a 3 tabelas,

Módulo - Triggers para UPDATE, DELETE e INSERT

Quando o módulo é introduzido no sistema este cria a tabela de histórico para cada uma das tabelas. Foram também criados triggers para operações de modificação da base de dados como update, delete e insert. Quando alguma destas operações se verifica, despoleta uma acção no servidor para manter histórico destas operações e insere a informação de que elas sucederam nas tabelas criadas inicialmente quando o módulo foi iniciado.

Este módulo consiste em três procedimentos que correm automaticamente no dado momento que o módulo é introduzido.

O primeiro procedimento cria todas as tabelas de histórico para todas as tabelas existentes na base de dados.

O segundo procedimento cria a função que irá permitir, dependendo da operação, enviar a informação de Utilizador que executou a operação, Tipo de Operação, Tempo de execução, e cópia das colunas da tabela que está sendo de histórico.

O terceiro procedimento faz então a criação de triggers para cada uma das tabelas da base de dados (excepto de histórico). Sempre que uma destas sofrer uma alteração o trigger é executado automaticamente, executando a função criada no segundo procedimento.

Módulo - Recuperação do histórico de determinado tuplo

Este módulo consiste num procedimento que executa no instante que o módulo é adicionado ao sistema.

Consiste na criação automática de funções para cada uma das tabelas. Cada tabela tem uma função que permite através da chave primária do tuplo saber o seu histórico desde a data de criação.

Como este processo é dinâmico, o procedimento necessita de encontrar todos os atributos e chaves primárias de cada tabela usando cursores (em cada iteração do loop) para que consiga criar uma função com os argumentos e campos de retorno correctos da tabela correspondente. No momento que a função está “construída” esta é executada e passa para a próxima iteração do loop até não haver tabelas no schema public.

Base de função gerada:

tuple_history_<table_name>(**pkey1** data_type, **pkey2** data_type, ..., **pkeyN** data_type)

Módulo - Recuperação do histórico de determinado tuplo

Este módulo consiste num procedimento que é executado quando é chamado no servidor.

Semelhante ao módulo anterior mas com campo adicional nas funções. Este campo corresponde a um timestamp, que quando chamada a função dado chave(s) primária(s) mais um timestamp devolve o último estado estado do tuplo anterior a esse timestamp.

O procedimento restringe mais a procura dos atributos das tabelas usando cursores de forma a que o retorno de tabela proveniente da função contenha só a informação do tuplo e não mais informação adicional.

Base de Função gerada:

given_tuple_time_at_<table_name>(**pkey1** data_type, ... ,**pkeyN** data_type, **ts** text)