

Índice general

1. Introducción	1
2. Diseño	3
2.1. Bases	3
2.2. JDBC y su envoltorio	3
2.3. Abstracción de SQL	3
2.3.1. Diseño de CREATE TABLE	4
2.3.2. Diseño de UPDATE e INSERT	4
Apéndices	5

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

trabajando con mysql 5.1 reference manual

trabajando con PostgreSQL 8.4 reference manual

sqlite con el de la pagina

Capítulo 2

Diseño

Acá un resumen del capítulo

2.1. Bases

Para diseñar JDBGM se tomo como idea base el patrón de desarrollo DAO

2.2. JDBC y su envoltorio

y otro

2.3. Abstracción de SQL

Así que para brindar una abstracción sobre la base de datos es necesario que las consultas SQL (sin hacer diferencias sobre DDL ¹ y DML ²) no sean manejadas de forma explícita es decir que no seria conveniente, por ejemplo, manejar las consultas como cadenas de caracteres ya que esta es una estructura estática dependiente del DBMS que se este usando. Por lo que para poder manejar adecuadamente las consultas se considero usar una estructura de datos un poco mas compleja que contenga la consulta de manera desglosada. Es decir que en ves de tener una sentencia como la siguiente:

```
| nombre_comando [parametro] opcion1 [parametro] opcion2 [parametro];
```

Se tendría una estructura de datos como la siguiente:

Pseudocódigo de la estructura de dato que contiene la sentencia

```
1 class Sentencia{  
2     nombre_comando;  
3     opcion1;  
4     opcion2;  
5     devolver_sentencia;  
6 }
```

Este conjunto estará formado por:

1. CREATE TABLE - Permite crear tablas
2. ALTER TABLE - Modifica la estructura de una tablas ³

¹Data Definition Lenguaje

²Data Manipulation Lenguaje

³Solo se da un soporte básico para esta sentencia

3. UPDATE - Modifica las filas de una tabla
4. INSERT - Inserta nuevas filas a una tabla
5. DELETE - Elimina columnas de una tabla
6. SELECT - Realiza consulta sobre las tablas

2.3.1. Diseño de CREATE TABLE

Esta sentencia es usada para crear tablas en una base de datos relacional, un resumen de su sintaxis como se la define en el estándar SQL es la siguiente:

```
CREATE TABLE <table name> (  
  { <column name> [ <column type> ] [ PRIMARY KEY ] [ REFERENCES <foreign table> ] }...  
)  
[ PRIMARY KEY <indexed columns> ]  
[ FOREIGN KEY <columns and referenced table> ]
```

Así que finalmente tenemos la siguiente sintaxis para nuestro proyecto

```
CREATE [ TEMPORARY ] TABLE <database name> <dot> <table name>  
<table contents source>  
  
<table contents source> ::=  
  <left paren> <table element> [ { <comma> <table element> }... ] <right paren>  
  | AS <select stmt>
```

2.3.2. Diseño de UPDATE e INSERT

Y

Apéndices

