



HQLite Library

Requirements

- SQLite lacks strong type checking.
- SQLite does not handle SQL Injection attack.
- So we need a SQLite interface library to handle this

Design

Haskell Application

Parser

- Parses query and create appropriate object .
- It will catch parsing error in sql query.

↑
Query



HQLite Library

- This library is used for interfacing Haskell applications with SQLite
- HQLite Library contains functions to validate the different SQLite queries and execute them if validated else throws appropriate error.
- Validation is mainly done for two aspects : 1. Type Checking and 2. SQL Injection Handling



- Weakly Typed
- Can't Handle Sql Injection



HQLite Library

Implementation

- HQLite library implements the different sql queries. It exports data types and functions corresponding to each query. It also implements the where clause in general way which is used in most of the queries. We perform type and sql injection checks here.
- Parser uses Parsec library and HQLite library. It parses the input query and creates the corresponding instance of data type and calls the HQLite function to validate the query and run on database.
- We use SQLite as a backend database.

Sample Case :

Parsing phase:

Input - "select * from emp where name = 'prashant' or '1' = '1';"

Object –Select (HqlSelectTable tableName [Hqlcolum] HqlExp)

Execute Function –
execSelect Table object query



Validate the object HqlSelectTable
If successfully validated then send
select query to SQLite database else
throw exception

Haskell Features Used

- Libraries like Parsec, Language are used in Parser,
- SQLite , Monad, Functor, Applicative libraries are used in implementation of HQLite library