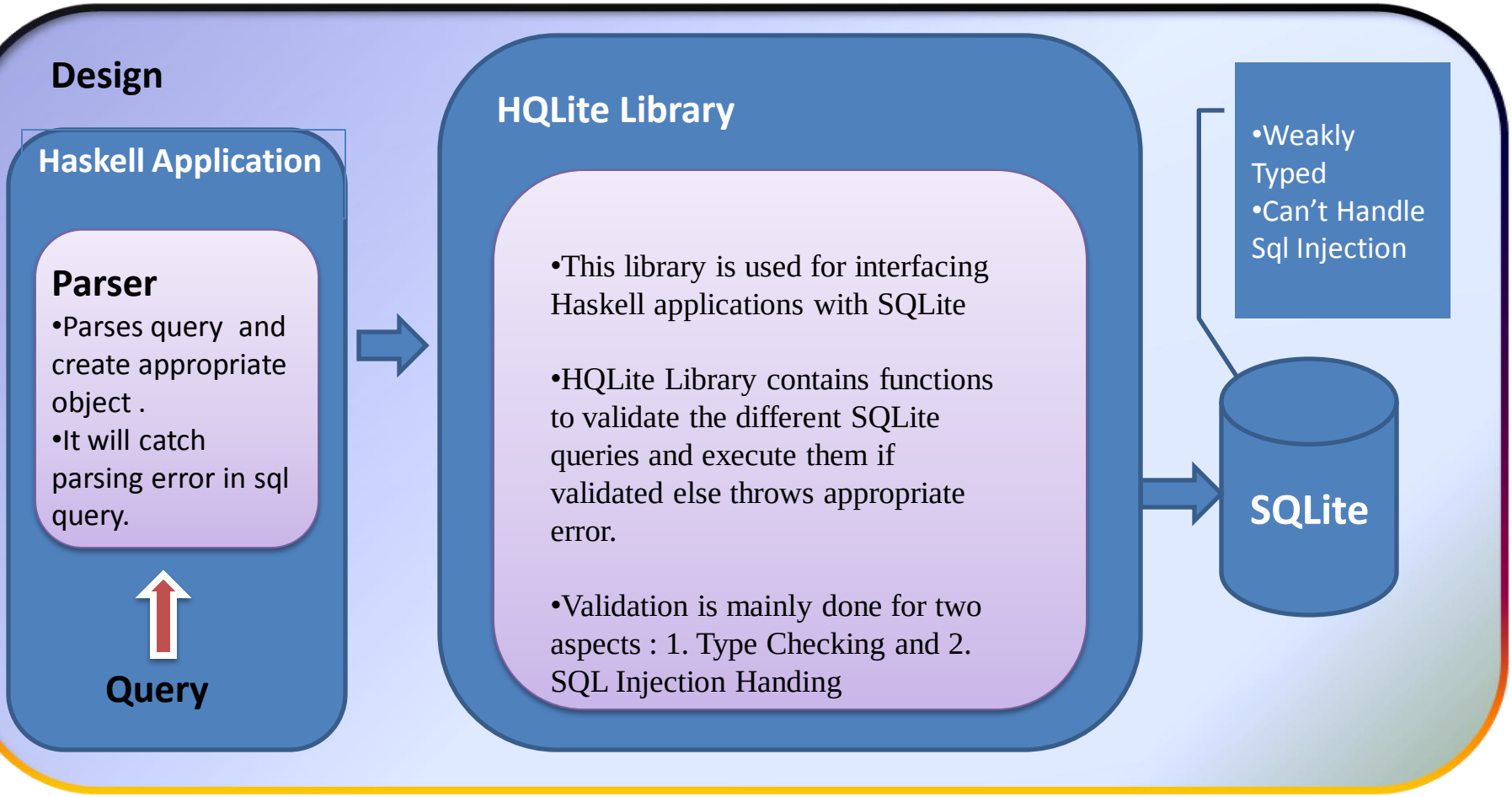
HQLite Library

Requirements

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

Design



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 is 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 tabName [Hqlcolum] HqlExp)
Execute Function –
execSelect Table object query

Validate the object HqlSelectTable
If successfully validated then send select query to SQLite databse 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