**Abstract**

Database Interface is an industry-standard tool for application development. Using Database Interface, developers can interact with any back end software (i.e. Oracle, MS-SQL Server, MS-Access, My-SQL etc). The database interface can be used to build, test, and debug PL/SQL packages, procedures, triggers, and functions. Database Interface users can create and edit database objects such as tables, views, indexes, constraints, and users. Database Interface's SQL Editor provides an easy and efficient way to write and test scripts and queries, and its powerful data grids provide an easy way to view and edit data related to any DBMS/RDBMS tool.

The innovative idea behind this project is to build a Generic Database Interface. Any user who has expertise in any Database can login through this Interface to connect to his acquainted Database can do all database operations.

**Advantages:**

* **Flexibility**. The end-user will be able to use any major commercial (or open-source) database on the market.
* **Maintenance**. Since the project is going to be developed using most of the industrial standards coding will be very much understandable and documentation shows the points of extensibility.
* **Ease of Use**. Installation is simple; you just have to create a .war file and drop it in the right directory/folder. The Java search engine interface is text-based, and intended to provide easy access to search operations.

INTRODUCTION & OBJECTIVES

Database Interface is an industry-standard tool for application development. Using Database Interface, developers can interact with any back end software (i.e. Oracle, MS-SQL Server, MS-Access, My-SQL etc). The database interface can be used to build, test, and debug PL/SQL packages, procedures, triggers, and functions. Database Interface users can create and edit database objects such as tables, views, indexes, constraints, and users. Database Interface's SQL Editor provides an easy and efficient way to write and test scripts and queries, and its powerful data grids provide an easy way to view and edit data related to any DBMS/RDBMS tool.

The requirements of a database application developer will vary from project to project. On a large team where DBAs manage the DDL, a developer may spend 90% of development time coding and testing SELECT queries to issue from 3GL or 4GL application code. In such an environment, a developer might be concerned only with viewing the DDL and database code. On smaller teams, a developer might be responsible for maintenance of the development schema, movement of test data between schemas, writing procedure code, populating tables from legacy sources, and more. Database Interface facilitates all of these needs.

For example if you are working with oracle, you don't have to be a PL/SQL expert to access database objects with Database Interface. You can view the Oracle Dictionary, tables, indexes, stored procedures, and more - all through a multi-tabbed browser. Database Interface utilizes direct Oracle OCI calls for full access to the Oracle API.

Advanced editing features save time and increase productivity. Code can be created from shortcuts and code templates. You can even create your own code templates.

**Use Database Interface to**

1. Create, browse, or alter objects (tables, views, indexes, etc.) including Oracle8 TYPE objects
2. Graphically build, execute, and tune queries
3. Edit and Debug PL/SQL and profile "stored procedures" including functions, packages, and triggers
4. Search for objects
5. Find and fix database problems with constraints, triggers, extents, indexes, and grants

**Modules:**

1. Structures Module
2. Properties Module
3. SQL Querying Module
4. Import Database Module
5. Export Database Module
6. DML Operations Module
7. Search Operations Module

The present application can be differentiated into the following modules, which are closely integrated with one another.

1. Structure: It gives the list of all tables which are present in the current/selected user. This module is used to browse and view the structure of an existing database object.
2. Properties: The properties module enables us to modify data types, size and constraints assigned to each field.
3. SQL: Using this module, we can use and implement different kinds of SQL statements. The result of the SQL statement will be displayed immediately after executing the statement.
4. Import: This module gives an exclusive feature of importing the existing SQL scripts to the existing schema.
5. Export: This module gives the feature of exporting existing schema object(s) to an SQL file. Here the facility is there to export only tables, only single user or the entire schema.
6. Operations: This module will be used to create new database objects.
7. Search: Search module provides searching facilities for searching for the given column names, data or both options.

**Technical Environment:**

|  |  |  |
| --- | --- | --- |
| Web Server | : | Tomcat 5.5 |
| Server-side Technologies | : | Java, Java Server Pages, Servlets |
| Client-side Technologies | : | Hyper Text Markup Language, Cascading Style Sheets, Java Script, AJAX |
| Operating System | : | Windows (or) Linux (or) Mac any version |