SUDAN UNIVERSITY OF SCIENCE & TECHNOLOGY
COLLEGE OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

DATABASE ABSTRACTION LAYER

A PROJECT SUBMITTED AS ONE OF THE REQUIREMENT FOR OBTAINING A BACHELOR FOR OF HONOR IN COMPUTER SYSTEM AND INFORMATION SYSTEM.

OCTOBER 2016

بسم الله الرحمن الرحيم

SUDAN UNIVERSITY OF SCIENCE & TECHNOLOGY
COLLEGE OF COMPUTER SCIENCE & INFORMATIONTECHNOLOGY
COMPUTER SCIENCE & INFORMATION SYSTEM DEPARTMENT

DATABASE ABSTRACTION LAYER

OCTOBER 2016

PREPARED BY:
MOHAMED ABDALLA SAADALDEEN
HOZAIFA MUSTAFA HAMAD

A PROJECT SUBMITTED AS ONE OF THE REQUIREMENT FOR OBTAINING A BACHELOR FOR OF HONOR IN COMPUTER SYSTEM AND INFORMATION SYSTEM.

SUPERVISOR: SIGNATURE OF SUPERVISOR
Mr. AYMEN ABDALAZIZ
Mr. MOHAMED NAFI DATE

الآيــــــــــــــــــــة

قال تعالى:

وَيَرَى الَّذِينَ أُوتُوا الْعِلْمَ الَّذِي أُنْزِلَ إِلَيْكَ مِنْ رَبِّكَ هُوَ الْحَقَّ وَيَهْدِي إِلَى صِرَاطِ الْعَزِيزِ الْحَمِيدِ))

(سبأ (الآية 6

الحمد لله

الحمد لله رب العالمين، أعطى اللسان، وعلَّم البيان، وخلق الإنسان، فبأي ألآء ربكما تكذبان لک الحمد يا من هو للحمد أهل، أهل الثناء والمجد، أحقُ ما قال العبد وكلنا لک عبد
لک الحمد.. من ضعيف يطلبُ نصرتک
لک الحمد .. من ف قير يطلبُ غناک
لک الحمد.. من ذليل يطلبُ عزک
لک الحمد ..من طالب علم يطلبُ توفي قک
لک الحمد.. ما دعوناک إلا حسنَ ظنٍ بک.. وما رجوناک إلا ث قةً فيک، وما خفناک إلا تصدي قاً بوعدک ووعيک.. فلک الحمد . والصلاة والسلام على علم الأعلام، وإمام كل إمام، محمد بن عبد الله وعلى آله وصحبه أجمعين

الحمد لله حمدا ليس منحصرا على أياديه ما يخفى وما ظهرا
ثم الصلاة وتسليم المهيمن ما هب الصبا فأدر العارض المطرا
على الذي شاد بنيان الهدى فسما وساد كل الورى فخرا وما افتخرا
نبينا أحمد الهادي وعترته وصحبه كل من آوى ومن نصرا
وبعد فالعلم لم يظفر به أحد إلا سما وبأسباب العلى ظفرا

الإهـــداء

إلى من جرع الكأس فارغاً ليس قيني قطرة حب إلى من كلّت أنامله لي قدم لنا لحظة سعادة إلى من

حصد الأشواک عن دربي ليمهد لي طريق العلم

أبي

وإلى من تتسابق الكلمات لتخرج معبرة عن مكنون ذاتهامن علمتني وعانت الصعاب لأصل إلى ما أنا

فيه

وعندما تكسوني الهموم أسبح في بحر حنانها ليخفف من آلامي ..

أمي

إلى من كانوا يضيئون لي الطريق ويساندوني ويتنازلون عن ح قو قهم لإرضائي والعيش في هناء

أحبكم حبا لو مر على أرض قاحلة

لتفجرت منها ينابيع المحبة

إخوتي

إلي كل من ساهم في هذا العمل المتواضح له جزيل الشكر الوافي إلى الذين بذلوا كل جهدٍ وعطاء لكي
أصل إلى هذه اللحظة أساتذتي الكرام .

الي كل من اشت قنا لهم وهم تحت الثرى ينعمون ان شاء الله الي احباب ورفاق ذهبوا وتركونا وحدنا
الي كل من افت قدناهم وافت قدنا لمساتهم وبصماتهم في هذه الحياه وهي ذكري لنا الي المبتسم دائما
رحمه الله أخونا وزميلنا محمد عثمان (جانفيس)نسأل الله ان يبلغه و يبلغهم دار خيراً من دارنا جميعاً.

Acknowledgement

First and above all, praise God, Peace and blessings be upon the Messenger of Allah, Mohammad is and prayers be upon him, Then thank God that guided me to what I am for providing me this opportunity and granting me the capability to proceed successfully I want to express my deep thanks for all Who have helped me in taking out this work from my teacher Special teacher Maria, and my friends and all those who gave me the idea, to the completion of this work.

Special thanks: To Everything you did for us, Thank you, respect and praise. Wish to say thank you, deep in my heart, God bless you Mr. MOHAMED NAFI and AYMEN ABDALAZIZ Who have given us a tender and the insightful discussion, offering valuable advice, for support from this project during the whole period of the study, and especially for your patience and guidance during the writing process. Again thank and gratitude to all of our colleagues and friends who joined us or helped us from this project.

ABSTRACT

Competition between producing companies RDBMS led to the difference in the SQL commands, making the programs work on a certain number of databases.

The project goal is to make the programs work with many database, and do not need to learn to syntax of SQL.

Since we were able to possible through building a library to develop software, which works as a layer between the program and the database, so the software will choose the process that wants to have with the database, and the class composition of the appropriate syntax of the database.

In the end, through the library we have the capability to deal with three types of data bases easily, and the library provide security on the database, and make the developers are able to build programs dealing with the database without having to refer to syntax of SQL Standard.

المستخلص

التنافس بين الشركات المنتجة لل RDBMS أدى إلى الأختلاف في أوامر ال SQL، مما جعل البرامج تعمل على عدد معين من قواعد البيانات.

الهدف من المشروع هو جعل البرامج تعمل مع قاعدة بيانات عديدة، و عدم الحاجة إلى تعلم بناء جمل الـSQL.

وبما أننا تمكنا من خلال بناء مكتبة لتطوير البرمجيات، والتي تعمل كطبقة بين البرنامج و قاعدة البيانات، وبالتالي فإن البرنامج سوف يختار العملية التي يريد أن يكونها مع قاعدة البيانات، وتكوين فئة من بناء الجملة الملائمة لـ قاعدة البيانات.

في النهاية، من خلال المكتبة لدينا الـ قدرة على التعامل مع ثلاثة أنواع من قواعد البيانات بسهولة (Sql Server, MySql, Oracle) والمكتبة يمكنها توفير الأمن على قاعدة البيانات هذه ، وجعل **المطورين قادرين على بناء برامج للتعامل مع قواعد البيانات دون الحاجة إلي الرجوع لأوامر الـ** SQL الـ قياسية .

LIST OF KEY TERMS

Term	Description
DBAL	**Database Abstraction Layer.**
RDBMS	**A relational database management system is a program that lets you create, update, and administer a relational database.**
ANSI	**The American National Standards Institute has served in its capacity as administrator and coordinator of the United States private sector voluntary standardization system.**
T-SQL	**Transact-SQL is a set of programming extensions from Sybase and Microsoft that add several features to the Structured Query**

	Language (SQL) including transaction control, exception and error handling, row processing, and declared variables.
IBM	International Business Machines is by far the world's largest information technology companies to products include hardware and software for a line of business servers, storage products, custom-designed microchips, and application software etc...
API	An application-programming interface is a set of programming instructions and standards for accessing a Web-based software application or Web tool.
DDL	Data Definition Language. These SQL statements define the structure of a database
DML	Data Manipulation Language. These SQL statements are used to retrieve and manipulate data
DCL	Data Control Language. These SQL statements control the security and permissions of the objects or parts of the database(s).
SSDT	SQL Server Data Tools used to Transact-SQL design capabilities to build, debug, maintain, You can work with a database project, or directly with a connected database instance on or off-premise.
PDO	The Hypertext Preprocessor Data Objects extension defines a lightweight consistent interface for accessing databases in PHP.
JVM	Java Virtual Machine is an abstract machine. It is a specification that provides runtime environment in which java byte code can be executed.
OOP	Object-oriented programming is a programming language model organized around objects rather than "actions" and data rather than logic.
JDBC	Java Database Connectivity is an application programming interface (API) which allows the programmer to connect and interact with databases

UML	**Unified Modeling Language in process analysis and standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology.**
BLOB	**Binary Large Object is a collection of binary data stored as a single entity in a database management system.**
CLOB	**Character Large Object is a collection of character data in a database management system.**

LIST OF FIGURES

Table of Contents

CHAPTER 1
INTRODUCTION

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION

These days, access to databases has become a frequent and widely, all daily transactions done with databases, exchange money, travel, registration of the university, even talking in social media need to databases, This made databases important and a way to earn profits, some companies earn all has profits from databases, such as Oracle , Microsoft, PostgreSQL Global Development Group etc. ,these companies produce what is called relational database management system(RDBMS) (1), RDBMS is a database management system (DBMS) that is based on the relational model, RDBMS is the basis for SQL, in 2014 - 2016, many of the databases in widespread use are based on the relational database model, so these companies focused on the production of the RDBMS.

The leading companies in this domain has become much, and to get a competitive advantage in the market, this company has become improve the performance of RDBMS, by making transactions performance faster, and make its own SQL commands more easier, the last feature to entice developers to develop programs using theirs RDBMS SQL commends.

1.2 PROBLEM STATEMENT

The difference in RDBMS’s SQL makes change the database type is a big problem, If the company decided to change the database that it use for some reason, this requires re-building the programs those based on the current database, This means increase cost wasting time.

1.3 RESEARCH QUESTION

How can we make a program deals with more than one database?

1.4 RESEARCH OBJECTIVE

1- Main focus of the project is to create maintainable, flexible, easy to use, fast and scalable database abstraction layer.

2- Providing a layer containing a number of databases allows the program to Perform same queries on different database types.

3- Identify the most popular types of databases and how to connect and handle them for use in building effective systems software.

4- Make the building of SQL syntax more secure by avoiding the SQL flaws.

5-Facilitate the process of changing databases present system a new database.

1.5 RESEARCH SCOPE

We will build layer between the software and the database in the form of Java library, library receives database type and the required query , and it build the required query for the specific type of database, the library will cover the DML commands, and it will includes three types of databases MySQL, Oracle and SQL Server.

1.6 EXPECTED CONTRIBUTION

A java library work in the form of a layer between the program and the database, the layer built the appropriate SQL command for the three types of data bases

1.7 DOCUMENT ORGANIZATION

In this research will be the order of the chapters as follows: -

The First chapter includes introduction, including clarifying Why the SQL commends are different, and the problem that for it want to build this system as well as also the question of research and the goals achieved by this system and the scope of research that we want to accomplish. The Second chapter includes previous studies about the data abstraction layer or similar systems operate on the same concept It includes of part .Third Chapter will be analysis and tools used in this system, for example, Java specially Java Database Connectivity (JDBC) and extensions of this other chapter. The Four chapter will be Unified Modeling Language (UML), and the analysis of the system but by use case, sequence diagram ……..etc. , Other tools to the content UML .The Fifth chapter is a part the contents software or the code. In Addition the Sixth chapter is final chapter in which it will be References and recommendations about the system

CHAPTER2
LITERATURE REVIEW

CHAPTER2 LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter we will talk about the history of SQL began when a standard of the American National Standards Institute (ANSI).How is a standardized computer language that was originally developed by IBM (International Business Machines) for querying, some types of databases different database systems, or DBMS (Database Management Systems) such as: Microsoft Access, IBM DB2, Sybase, My SQL, Oracle, Microsoft SQL Server but we focus on commonly used by companies to use the systems the three types are the last.

SQL is a programming language for Relational Databases. It is designed over relational algebra and tuple relational calculus. SQL comes as a package with all major distributions of RDBMS.SQL comprises both data definition and data manipulation languages. Using the data definition properties of SQL(2), one can design and modify database schema, whereas data manipulation properties allows SQL to store and retrieve data from database.SQL Database is a relational database service in the cloud based on the market-leading Microsoft SQL Server engine, with mission-critical capabilities. SQL Database delivers predictable performance, scalability with no downtime, business continuity and data protection—all with near-zero administration. You can focus on rapid app development and accelerating your time to market, rather than managing virtual machines and infrastructure. Because it's based on the SQL Server engine, SQL Database supports existing SQL Server tools, libraries and APIs, which makes it easier for you to move and extend to the cloud .And in this chapter we talk in detail about the types of databases that we have mentioned previously. In addition previous studies on Data abstraction layer is usually the first step in database design. A complete database is much too complex a system to be developed without first creating a simplified framework. Data abstraction makes it possible for the developer to start from essential elements -- data abstractions -- and incrementally add data detail to create the final system.

2.2 SQL (Structured Query Language)

The language of SQL (Structured Query Language) as they are known to many is the standard language which is a computer language for storing, manipulating and retrieving data stored in relational database. It was the beginning of Structured Query Language in 1974, where he d / Chamberlain definition of a language called the Structured Query Language English) SEQUEL (and was made up of words and symbols of the English language, and in 1976 was the definition of a revised version dubbed SEQUEL / 2, and developed until it became her name to SQL.SQL is the standard language for Relation Database System. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix and SQL Server use SQL as standard database language(3). It has evolved, and many versions and languages began to emerge. To unify SQL for best practices, the American National Standards Institute (ANSI) created specific standards for database query languages.

We use SQL because allows users to access data in relational database management systems, Allows users to define the data in database and manipulate that data and allows embedding within other languages using SQL modules, libraries pre-compilers.

2.2.1 SQL Process

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task.

There are various components included in the process. These components are Query Dispatcher, Optimization Engines, Classic Query Engine and SQL Query Engine, etc. Classic query engine handles all non-SQL queries, but SQL query engine won't handle logical files.

2.2.2 SQL statements are often divided into three categories:

Data Definition Language (DDL). These SQL statements define the structure of a database, including rows, columns; tables, indexes, and database specifics such as file locations. DDL SQL statements are more part of the DBMS and have large differences between the SQL variations (4).Data Manipulation Language (DML).These SQL statements are used to retrieve and manipulate data. This category encompasses the most fundamental commands including DELETE, INSERT, SELECT, and UPDATE. DML SQL statements have only minor differences between SQL variations .Data Control Language (DCL). These SQL statements control the security and permissions of the objects or parts of the database(s). DCL SQL statements are also more part of the DBMS and have large differences between the SQL variations. But we are in this system will be our work and our focus on Data Manipulation Language.

2.2.3 SQL Constraints

SQL Constraints are rules used to limit the type of data that can go into a table, to maintain the accuracy and integrity of the data inside table (5).

Constraints can be divided into following two types,

- **Column level constraints :** limits only column data
- **Table level constraints :** limits whole table data

Constraints are used to make sure that the integrity of data is maintained in the database. Following are the most used constraints that can be applied to a table.

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT

2.3 MYSQL

MySQL is a leading open source database management system. It is a multi-user, multithreaded database management system. MySQL is especially popular on the web. It is one of the parts of the very popular LAMP platform. Linux, Apache, MySQL and PHP. Currently MySQL is owned by Oracle. MySQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows or Mac. Wikipedia, YouTube, Face book use MySQL (6). These sites manage millions of queries each day. MySQL comes in two versions. MySQL server system and MySQL embedded system. The development of MySQL begun in 1994 by a Swedish company MySQL AB. Sun Microsystems acquired MySQL AB in 2008. Sun was bought by Oracle in 2010. So today, Oracle Corporation is the owner of the MySQL database. MySQL, PostgreSQL, Firebird, SQLite, Derby, and HSQLDB are the most well known open source database systems. MySQL is developed in C/C++. Except of the C/C++, APIs exist for PHP, Python, Java, C#. MySQL is the most popular Relational SQL database management system (7). It is one of the best RDBMS being used for developing web-based software applications. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is developed, marketed. MySQL is released under an open-source licenses we mentioned. So you have nothing to pay to use it and it is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.

2.3.1 Administrative MySQL Command:

Here is the list of important MySQL commands, which you will use time to time to work with MySQL database (8):

- **USE Database name**: This will be used to select a particular database in MySQL work area.
- **SHOW DATABASES:** Lists the databases that are accessible by the MySQL DBMS.
- **SHOW TABLES:** Shows the tables in the database once a database has been selected with the use command.
- **SHOW COLUMNS FROM table name:** Shows the attributes, types of attributes, key information, whether NULL is permitted, defaults, and other information for a table.
- **SHOW TABLE STATUS LIKE table name\G:** Reports details of the MySQL DBMS performance and statistics.

2.3.2 Some Reasons to Use MySQL:

- **Scalability and Flexibility:** The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. Platform flexibility is a stalwart feature of MySQL with all flavors of Linux, UNIX, and Windows being supported. And, of course, the open source nature of MySQL allows complete customization for those wanting to add unique requirements to the database server.
- **High** Performance: A unique storage-engine architecture allows database professionals to configure the MySQL database server specifically for particular applications, with the end result being amazing performance results. Whether the intended application is a high-speed transactional processing system or a high-volume web site that services a billion queries a day, MySQL can meet the most demanding performance expectations of any system. With high-speed load utilities, distinctive memory caches, full text indexes, and other performance-enhancing mechanisms, MySQL offers all the right ammunition for today's critical business systems.
- .High Availability: Rock-solid reliability and constant availability are hallmarks of MySQL, with customers relying on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability

options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

- Management Ease: MySQL offers exceptional quick-start capability with the average time from software download to installation completion being less than fifteen minutes. This rule holds true whether the platform is Microsoft Windows, Linux, Macintosh, or UNIX. Once installed, self-management features like automatic space expansion, auto-restart, and dynamic configuration changes take much of the burden off already overworked database administrators. MySQL also provides a complete suite of graphical management and migration tools that allow a DBA to manage, troubleshoot, and control the operation of many MySQL servers from a single workstation. Many third party software vendor tools are also available for MySQL that handle tasks ranging from data design and ETL, to complete database administration, job management, and performance monitoring.
- Lowest Total Cost of Ownership: By migrating current database-drive applications to MySQL, or using MySQL for new development projects, corporations are realizing cost savings that many times stretch into seven figures. Accomplished through the use of the MySQL database server and scale-out architectures that utilize low-cost commodity hardware, corporations are finding that they can achieve amazing levels of scalability and performance, all at a cost that is far less than those offered by proprietary and scale-up software vendors. In addition, the reliability and easy maintainability of MySQL means that database administrators don't waste time troubleshooting performance or downtime issues, but instead can concentrate on making a positive impact on higher level tasks that involve the business side of data(9).

2.4 ORACLE SQL

Structured Query Language (SQL) is the set of statements with which all programs and users access data in an Oracle database. Application programs and Oracle tools often allow users access to the database without using SQL directly, but these applications in turn must use SQL when executing the user's request.

2.4.1 ANSI Standards and Oracle

The standard format of SQL was developed by the American National Standards Institute (ANSI). ANSI works with companies like Oracle Corporation to develop its standards, thus helping to gain support among competitors for a unified standard that benefits everyone(10). Oracle Database provides full support for ANSI standard SQL and, like most database vendors, adds extra features making SQL more robust and versatile as a database access language. For example, Oracle Database contains a rich set of functions. These functions can be used to alter column data within queries. For instance, the UPPER function can convert all of the letters in a word to capitals, and ADD_MONTH can add a month to a date, among a comprehensive multitude of other function options

2.4.2 Oracle database

Every Oracle Database Contains Logical and Physical Structures. Logical Structures are table spaces, Schema objects, extents and segments. Physical Structures are Data files, Redo Log Files, Control File(10). A database is divided into logical storage units called table spaces, which group related logical structures together. Each Table space in turn consists of one are more data files. All the tables and other objects in Oracle are stored in table space logically, but physically they are stored in the data files associated with the table space. Every Oracle database has a set of two or more redo log files. The set of redo log files for a database is collectively known as the database's redo log. A redo log is made up of redo entries (also called redo records). The primary function of the redo log is to record all changes made to data. If a failure prevents modified data from being permanently written to the data files, the changes can be obtained from the redo log so work is never lost. Every Oracle database has a control file. Control files contain the database name and locations of all data files and redo log files. Every Oracle database also has a Parameter File. Parameter file contains the name of the Database, Memory Settings and Location of Control file.

2.4.3 SQL Tools

Oracle has provided a user-friendly interactive tool for running SQL since its first release (11). The SQL*Plus tool today has four variations from which to choose:

- SQL*Plus Command Line. Use this when you don’t have a Windows interface, such as when using telnet to reach a remote UNIX database server.
- SQL*Plus Windows. Use this in a Windows-capable environment(can be invoked using a network name from a client or directly on the database server, regardless of the operating system).
- SQL*Plus. This gives you the same interface as SQL*Plus Windows, except it runs in a Web browser. Use this to run SQL commands and automatically generate a report in HTML format.

2.4.4 Oracle Database Architecture

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance (12). A database server also prevents unauthorized access and provides efficient solutions for failure recovery. Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed. The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

2.5 MICROSOFT SQL SERVER

SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. TSQL is very similar to standard SQL, but in addition it supports some extra functionality, built in functions, etc. T-SQL expands on the SQL standard to include procedural programming, local variables, and various support functions for string processing, data processing mathematics, etc.

2.5.1. SQL Server Data Tools

SQL Server Data Tools (SSDT) transforms database development by introducing A ubiquitous, declarative model that spans all the phases of database development inside Visual Studio. You can use SSDT Transact-SQL design capabilities to build, debug, maintain. You can work with a database project, or directly with a connected database instance on or off-premise.

Developers can use familiar Visual Studio tools for database development. Tools such as: code navigation, IntelliSense, language support that parallels what is available for C# and Visual Basic, platform-specific validation, debugging, and declarative editing in the Transact-SQL editor. SSDT also provides a visual Table Designer for creating and editing tables in either database projects or connected database instances. While you are working on your database projects in a team-based environment, you can use version control for all the files. When it’s time to publish your project, you can publish to all supported SQL platforms; including SQL Database and SQL Server. SSDT platform validation capability ensures that your scripts work on the target you specify.

The SQL Server Object Explorer in Visual Studio offers a view of your database objects similar to SQL Server Management Studio. SQL Server Object Explorer allows you to do light-duty database administration and design work. You can easily create, edit, rename and delete tables, stored procedures, types, and functions. You can also edit table data, compare schemas, or execute queries by using contextual menus right from the SQL Server Object Explorer (13).

2.6 PREVIOUS STUDIES

There are previous three studies in database abstraction layer.

2.6.1 ADOdb Database Abstraction Layer (John Lim).

Allow the use of different database servers using the same code base. ADOdb is a fast, easy to use, popular database abstraction layer for PHP. It allows the same code to be used when accessing a wide range of databases. It has been actively maintained since 2000 by the project's founder and numerous community contributors (14). Adobe contains components for querying and updating databases, as well as an Object Orientated Active Record library, schema management and performance monitoring. It also contains the following standalone extensions:

- A Date/Time library to handle dates outside of the normal PHP limits.
- A Session Management library that extends the normal PHP functionality to allow storing of session management data in a database, or in encrypted values.

Drupal's database abstraction layer provides a unified database query API (Application Programming Interface*)* that can query different underlying databases. It is built upon PHP's PDO (PHP Data Objects) database API, and inherits much of its syntax and semantics. Besides providing a unified API for database queries, the database abstraction layer also provides a structured way to construct complex queries.

But This Study Use PHP as Programming Language which suffers from?

- Security : Since it is open sourced, so all people can see the source code, if there are bugs in the source code, it can be used by people to explore the weakness of PHP
- Not suitable for large applications: Hard to maintain since it is not very modular.
- Weak type: Implicit conversion may surprise unwary programmers and lead to unexpected bugs. For example, the strings “1000” and “1e3” compare equal because they are implicitly cast to floating point numbers.

2.6.2 Database abstraction layer for C++.

It describes a set of classes which can be used to access an arbitrary SQL based relational database system. There exist a lot of different database systems in the world, with a lot of different API’s for accessing data within a C++ program. The most API’s are in plain old C, which make the usage even harder for novice programmers. So we need a simple harmonized layer which can be easily extended for new database systems and also easily used by programmers (15).

Accessing different database systems in a C++ program can be very cumbersome, because of the many different API's; almost all of them are in plain old C. There exist a number of abstractions to access different database systems with the same API. For example ODBC (Open Database Connectivity). But ODBC on its own has a very large, verbose and complex structure, and it also has been done in plain old C." Different C++ Frameworks provide different abstraction layers for database access, which try to solve the problem described above. But they can't be used outside of the context of the frameworks.

But This Study Use C++ as Programming Language which suffers from?

- Very complex! The learning curve is steep and takes a long time to climb.
- Poor in Multitasking.
- Has some design flaws, although they are largely fixed by boost libraries and the new language standard.
- It is not platform independent.

2.6.3 Database abstraction layer for Python.

Database Abstraction Layer (DAL) an API that maps Python objects into database objects such as queries, tables, and records. The DAL dynamically generates the SQL in real time using the specified dialect for the database back end; so that you do not have to write SQL code or learn different SQL dialects (the term SQL is used generically)(16).Python is a general-purpose high-level programming language. Its design philosophy emphasizes programmer productivity and code readability. It has minimalist core syntax with very few basic commands and simple semantics, but it also has a large and comprehensive standard library, including an Application Programming Interface (API).

2.6.3.1 Python used at database abstraction layer to:

- Python is object oriented with class-based inheritance. Everything is an object (including classes, functions, modules, etc), in the sense that they can be passed around as arguments, have methods and attributes, and so on.

- Python is multipurpose: it is not specialized to a specific target of users (like R for statistics, or PHP for web programming). It is extended through modules and libraries that hook very easily into the C programming language.
- Python is dynamically typed.

2.6.3.2 But in Python problems:

Python is much slower than some other languages and although it is dynamically but strongly-typed language, it can do no compile-time checks for type consistency. Many programmers who are used to static languages find this distinctly unnerving, imagining the proliferation of hard-to-detect errors.

2.7 SUMMARY

In this chapter, we talked about the Introduction to Structured Query Language and how it began and how it has been certified by the American National Standards Institute As we explained SQL process and when you are executing an SQL command for any RDBMS .In addition we reminded SQL statements (DDL, DML, and DCL) indicate to focus on the DML .Then some SQL Constraints. We talked in detail about three types of databases MySql, Oracle, and Microsoft Sql Server Finally some previous studies in database abstraction layer.

CHAPTER3

RESEARCH METHODOLOGY

CHAPTER3 RESEARCHMETHODOLOGY

INTRODUCTION

As is well known that the systems and software used and built on the database certain difficult overlap and dealing with other database in the system on but we will walk another approach and easy way multi-use purposes, in reconciling between a number of well-known databases and traded to use in the current systems, which will It includes a series of commands relevant to the designated databases that we use and this will be in the form of his libraries like Java in what is known as Database Abstraction Layer. Designing of database is most important responsibility of the software professionals who are dealing with the database related projects. For this they follow the Design Methodology. It helps the designer to plan, manage, control, and evaluate database development projects.

In this chapter the aim is to offer a new approach outlined as well as the old approach to databases and how to use tools that help in building databases have the ability to deal with each other and explain more of the orders, instructions environment and the appropriate build a library that we want to build to help many people in coping with full of change for the database system concerned appointed.

ADAPTATION APPROACHES

Writing deterministic such as writing a library programs is often difficult for problems whose optimal solutions depend on unpredictable properties of the programs' inputs. Difficulty is also encountered for problems where the programmer is uncertain about how to best implement certain aspects of a solution. For such problems a mixed strategy of deterministic programming and machine learning can often be very helpful: Initially, define those parts of the program that are well understood and leave the other parts loosely defined through default actions, but also define how those actions can be improved depending on results from actual program runs. Then run the program repeatedly and let the loosely defined parts adapt.

3.2.1 Benefits of library

- Reduce the size of your class files
- You can reuse a library in several projects.
- Cleaner API (Application program interface) since you can't leak internal fields.
- You can test your library independent of your application. If the library is good, then the bug must be in your app.

FEATURES OF APPROACHES

3.3.1 Features of SQL

- Can contain SQL Procedural Language statements and features which
- Support input parameters.
- Support nested functions calls to other SQL functions or functions implemented in other languages.

3.3.2 Features of MySQL

- Management Ease.
- High Performance.
- High Availability.

3.3.3 Features of MS SQL Server

- High Performance.
- High Availability.
- Row version-based isolation level.

3.3.4 Features of Oracle

- Portability.
- Self-managing database.
- Data mining.

3.3.5 System Scenario

Every framework should have a database abstraction layer. You can create insert, update and delete statements without needing to write raw SQL. The coolest thing about using the database abstraction layer provided in, you can change the database provider by simply changing a line in your comfit file. Currently supports MySQL, MS SQL, and Oracle.

MS Sql

Oracle

MySql

DB Abstraction Layer

Java

Query ()

EMPLOYED TECHNOLOGIES:

We are using several technologies such as:

3.4.1 Java Technologies:

Java technology is programming language and a platform that is programming language is a high-level, object-oriented language. Java programs are both compiled and interpreted. Compilation translates Java code into an intermediate language called Java byte code (17). It

Figure Database Abstraction Layer

is in turn parsed and run (interpreted) by the Java Virtual Machine (JVM) — a translator between the language and the underlying operating system and hardware. A compiled Java program can run on any system that has a version of the JVM.

The Java language was designed with the following properties:

- Object Oriented: In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- Platform independent: Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by virtual Machine (JVM) on whichever platform it is being run.
- Simple: Java is designed to be easy to learn. If you understand the basic concept of OOP Java would be easy to master.
- Secure: With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- Architectural-neutral: Java compiler generates an architecture-neutral object file format which makes the compiled code to be executable on many processors, with the presence of Java runtime system.
- Robust: Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- Multithreaded: With Java's multithreaded feature it is possible to write programs that can do many tasks simultaneously. This design feature allows developers to construct smoothly running interactive applications.
- Interpreted: Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light weight process.
- High Performance: With the use of Just-In-Time compilers, Java enables high performance.
- Distributed: Java is designed for the distributed environment of the internet.

- Dynamic: Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

3.4.2 But we are in our project we chose the programming language Java and so me:

- Java is platform-independent. One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.
- Java is secure. Considers security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.
- Java is easy to learn. Was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.

DEVELOPMENT TOOLS:

Java tools and technology help application developers build better code faster. Some Java tools are aimed at increasing the efficiency of applications running on the Java platform in terms of the computing and storage resources necessary to run the application (18). Other Java technologies are focused on making Java applications more maintainable via modularity and component reuse or other strategies. Still other tools and technologies are designed to make each application developer more effective by increasing communication (collaboration tools) or reducing the need for repetitive boilerplate code (high level abstractions and Java code libraries).

3.5.1 Java Database Connectivity

Java Database Connectivity (JDBC) is an application programming interface (API) which allows the programmer to connect and interact with databases. It provides methods to query and update data in the database through update statements like SQL's CREATE, UPDATE, DELETE and INSERT and query statements such as SELECT (19). Additionally, JDBC can run stored procedures.

JDBC actually has two levels of interface. In addition to the main interface, there is also an API from a JDBC "manager" that in turn communicates with individual database product "drivers," the JDBC-ODBC Bridge if necessary, and a JDBC network driver when the Java program is running in a network environment (that is, accessing a remote database). When accessing a remote database, JDBC takes advantage of the Internet's file addressing scheme and a file name looks much like a Web page address (or Uniform Resource Locator).

JDBC specifies a set of object-oriented classes for the programmer to use in building SQL requests. An additional set of classes describes the JDBC driver API. The most common SQL data types, mapped to Java data types, are supported. The API provides for implementation-specific support for Microsoft Transaction Server requests and the ability to commit or roll back to the beginning of a transaction.

3.5.2 JDBC Characteristics:

- A collection of classes and interfaces for Java, written in the Java programming language
- An aid for writing database applications since it provides a standard API for database programmers
- A mechanism through which Java applications can talk to a wide range of databases
- A program which allows a programmer to write a single program once using the JDBC API, and then run it on any platform with the combination of Java and JDBC
- A way to easily send SQL statements to any relational database
- A low level interface using JDBC SQL commands (These commands can be invoked directly because JDBC is designed to be a base to build higher level interfaces and tools. JDBC also aids in establishing a connection with the databases and facilitates sending SQL statements and processing the result.)

3.5.3 JDBC Architecture

Java Database Connectivity (JDBC) architecture is an API specifying interfaces for accessing relational databases. JDBC helps to connect to a database, send queries and updates to the database, and retrieve and process the results obtained from the database for queries (20).The JDBC architecture supports two-tier and three-tier processing models for accessing a database.

In the two-tier model, a Java applet or application communicates directly to the data source. The JDBC driver enables communication between the application and the data source. When a user sends a query to the data source, the answers for those queries are sent back to the user in the form of results. The data source may not always be a single machine located at a single place. It can be located on a different machine on a network to which a user is connected. This is known as a client/server configuration, where the user's machine acts as a client and the machine having the data source running acts as the server. The network connection can be intranet or Internet.

In the three-tier model, the user's commands or queries are sent to middle-tier services, from which the commands are again sent to the data source. The results are sent back to the middle tier and from there to the user. This type of model is found very useful by management information system directors, as it makes it simple to maintain access control and make updates to corporate data. Application deployment also becomes easy and provides performance benefits.

3.5.4 Net Beans IDE version 8.1

Net Beans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. (21). The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems. The platform offers many features to the developers so they can better enhance and improve their applications.

- **Some of the features of platform are:**
- Integrated Development Tools
- Debugging Tools
- Building Project into .jar

SUMMARY

In this chapter, we explained how a methodology for building and problems the library where experience some of the benefits of this library addition to the qualities and attributes appropriate, which include Sql, MySQL, SqlServer, Oracle, Series of commands relevant to the designated databases and figure explained model of data abstraction layer in java . In addition Techniques used to build and develop the tools used in it and the environment in which we use for they contain that library.

The next chapter will be analysis and design and the inclusion of more details of the library and explain their work in the form or in the form of diagrams showing how to deal with it.

CHAPTER 4
SYSTEM ANALYSIS

CHAPTER 4 SYSTEM ANALYSIS

4.1 INTRODUCTION

Most people refer to the Unified Modeling Language (UML) in process analysis. The UML is an international industry standard graphical notation for describing software analysis and designs (22). UML models: use case, sequence, activity and class diagrams.

4.2 USE CASE DIAGRAMS

Which is use case illustrates a unit of functionality provided by the system and describe the operations that can be performed by system users. Which is library of Java Database abstraction layer (DBAL) include all operations of data manipulation language (Insert, Select, Update, Delete).

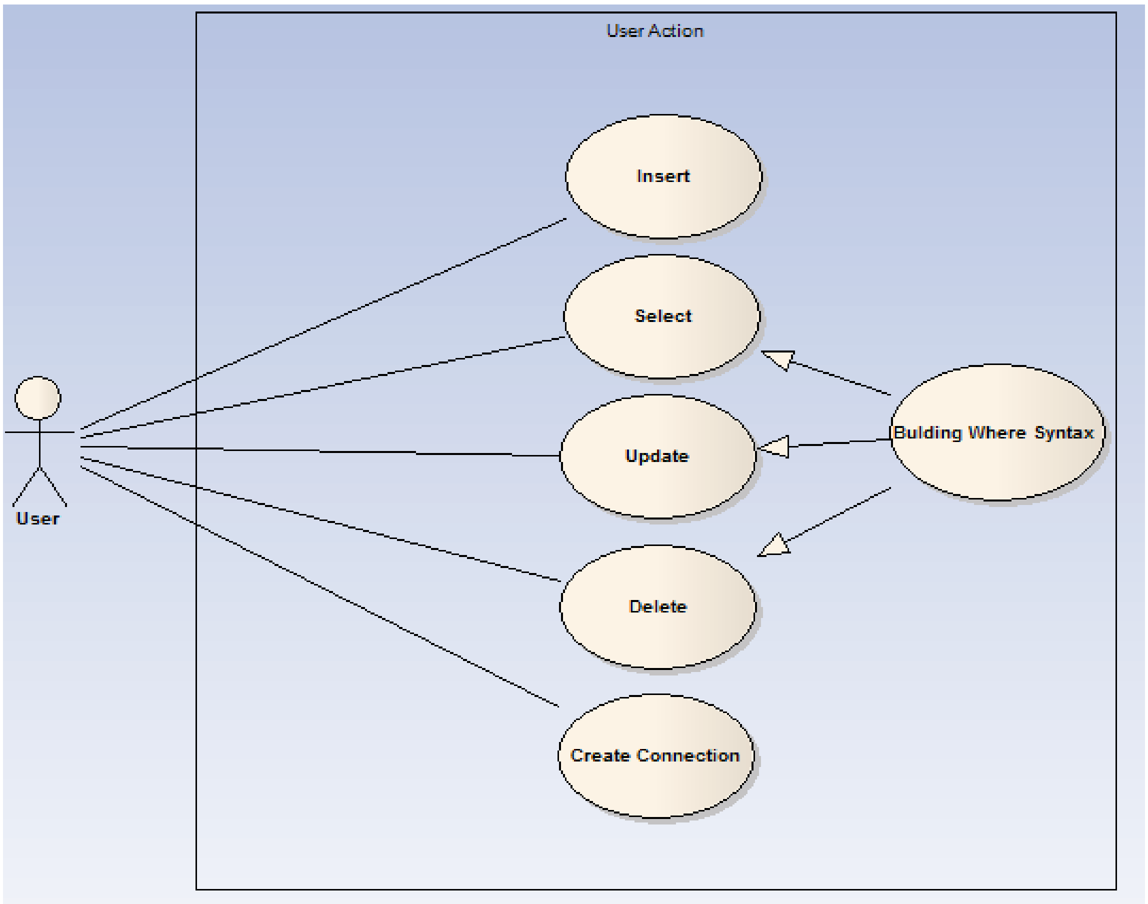


Figure : Illustrates the Use Case of process DBAL

4.3 SEQUENCE DIAGRAM

Sequence diagrams show a detailed flow for a specific use case or even just part of a specific use case.

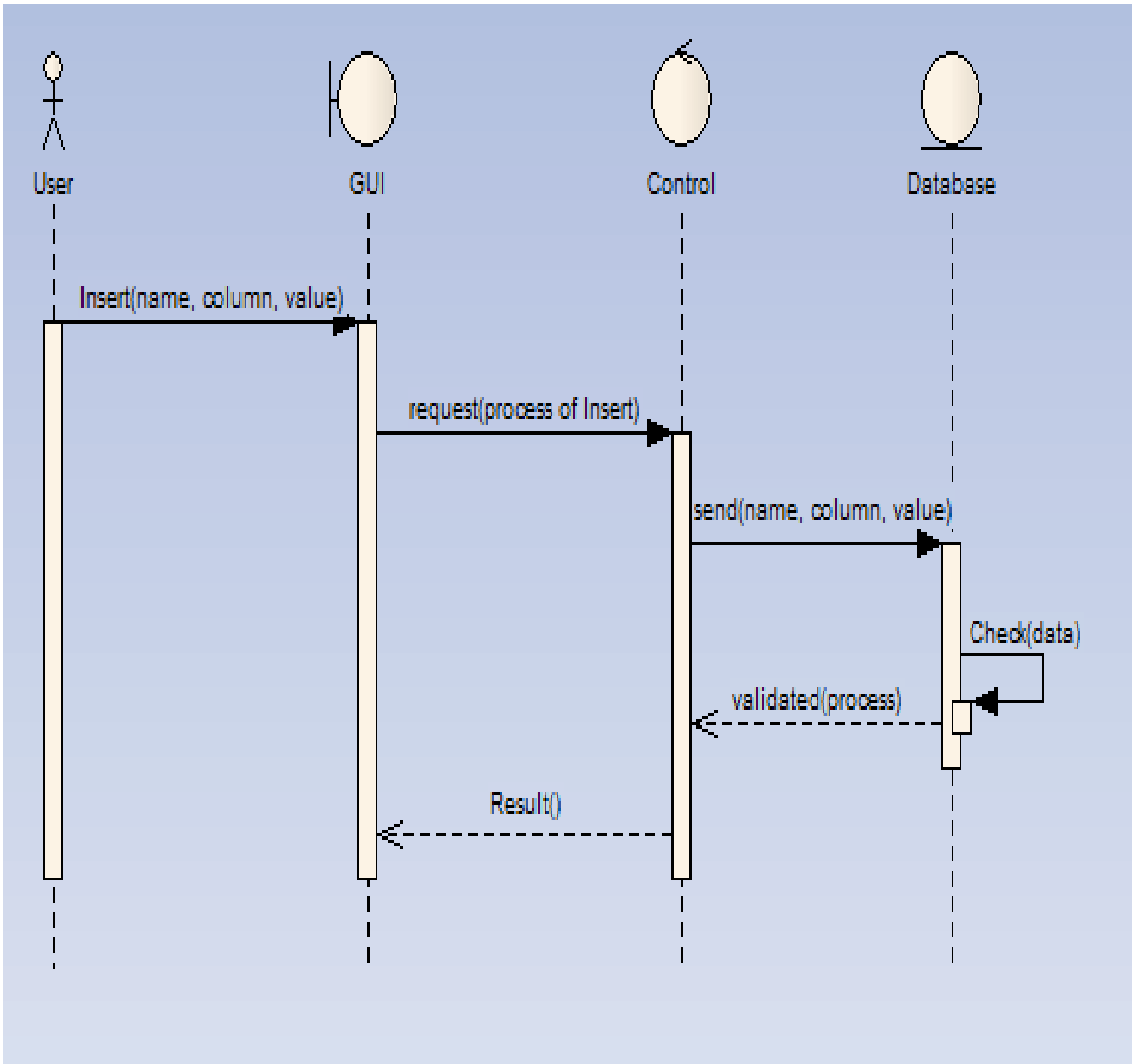

User
GUI
Control
Database
Insert(name, column, value)
request(process of Insert)
send(name, column, value)
Check(data)
validated(process)
Result()

Figure : Illustrates the sequence of process to insert.

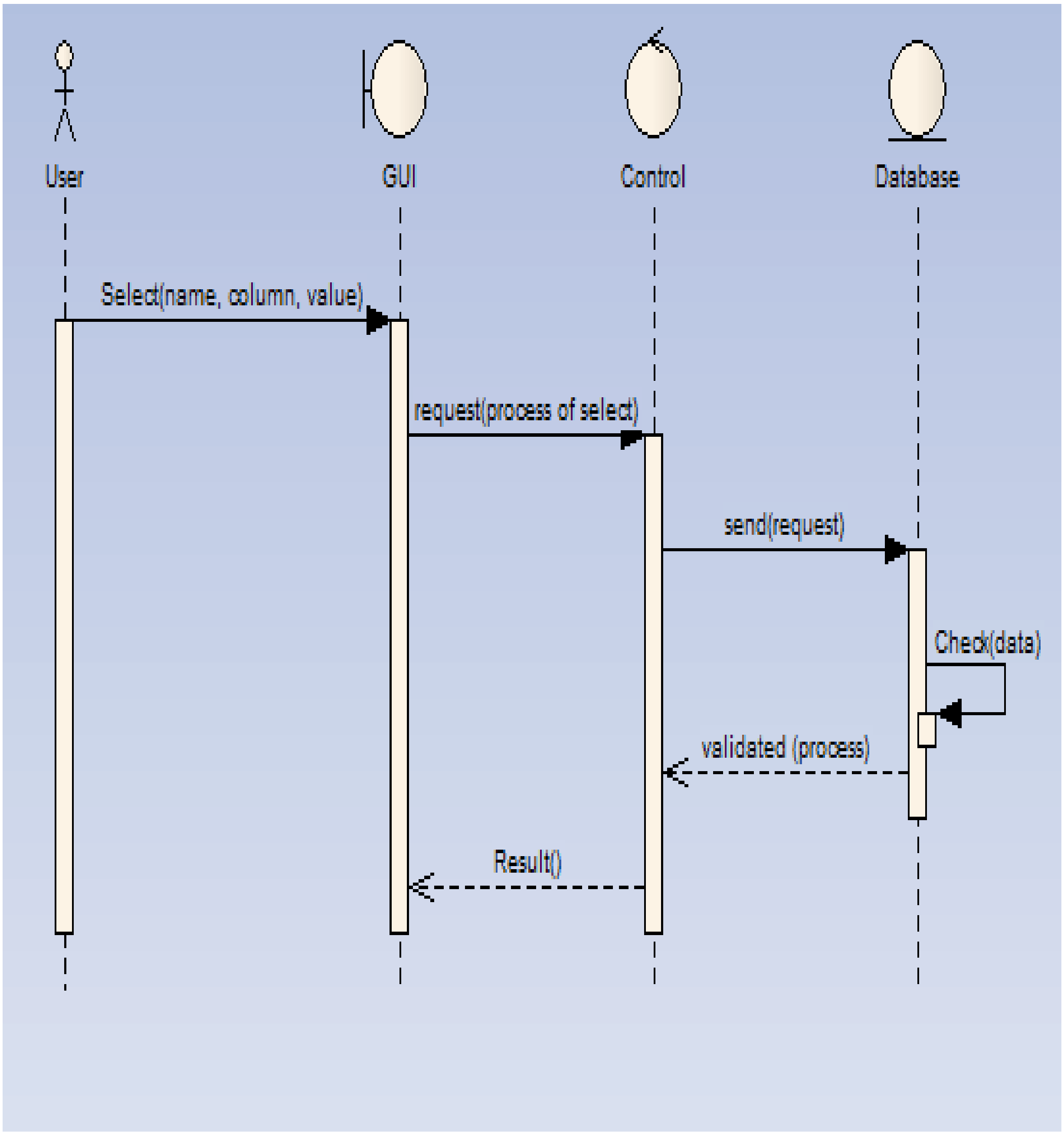


Figure : Illustrates the sequence of process to select.

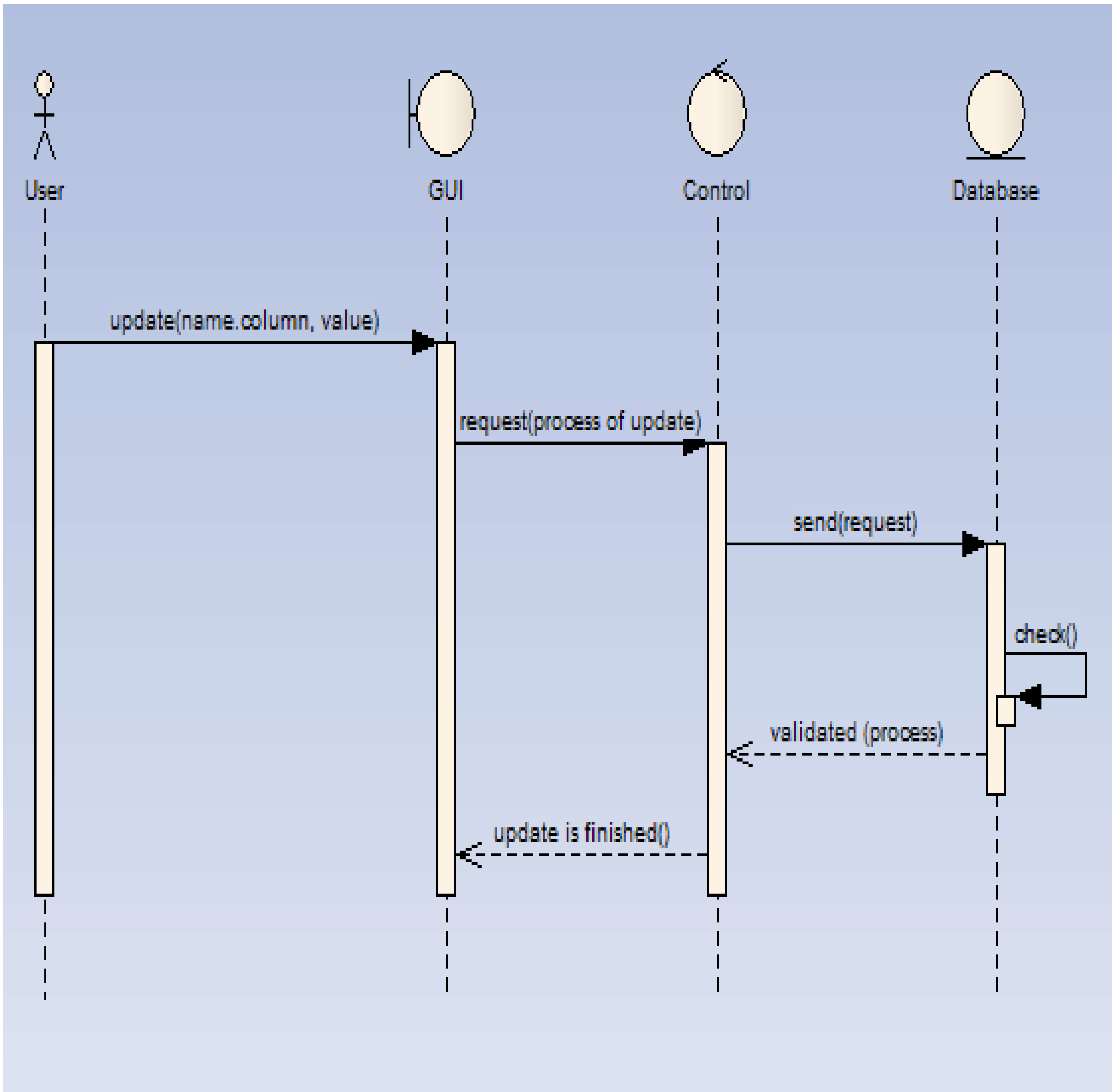


Figure : Illustrates the sequence of process to update.

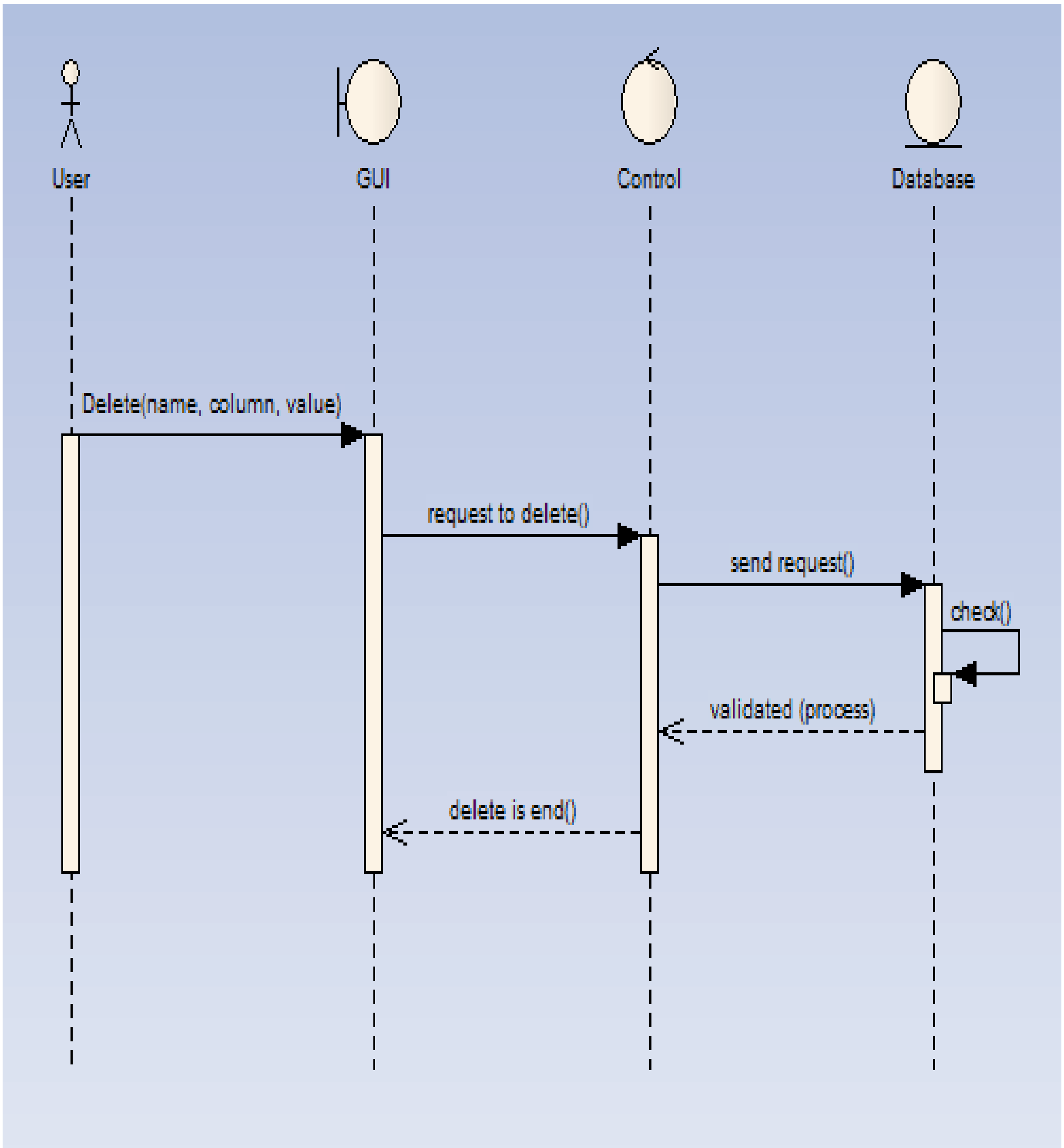


Figure : Illustrates the sequence of process to delete.

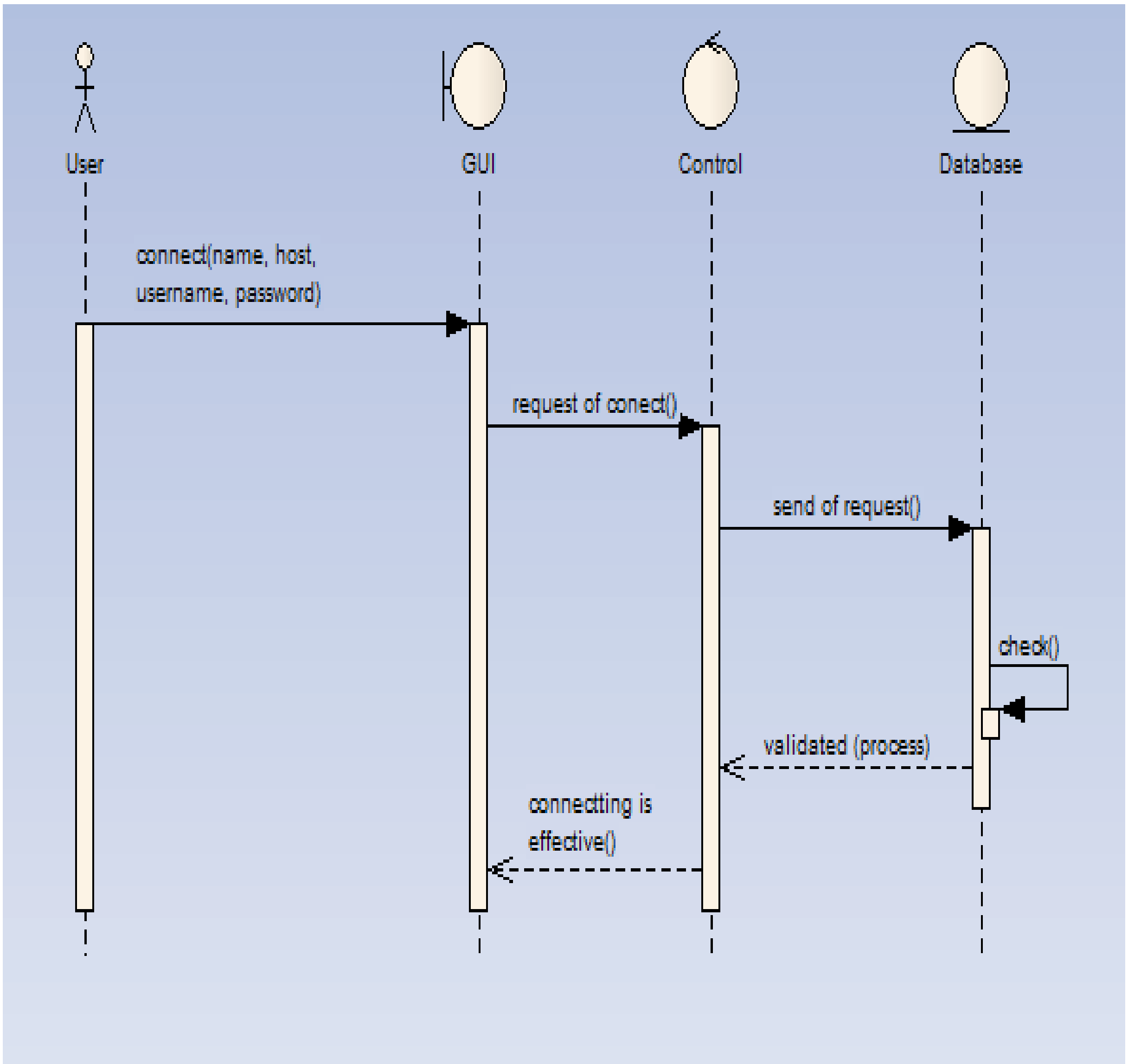


Figure : Illustrates the sequence of process to connect.

4.4 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamic aspects of the system of all process in DML in this library.

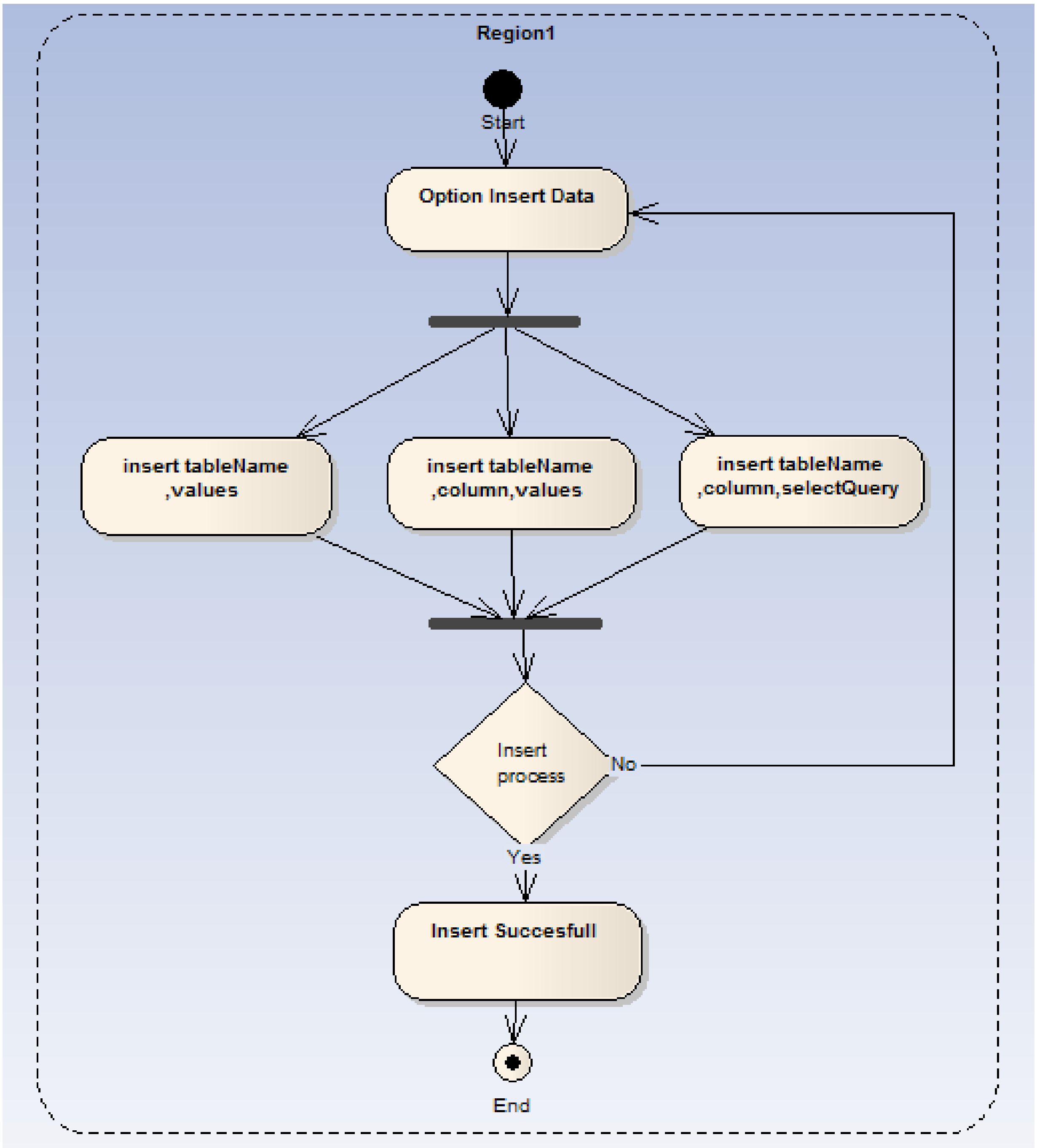


Figure   Illustrates the Activity of process to insert.

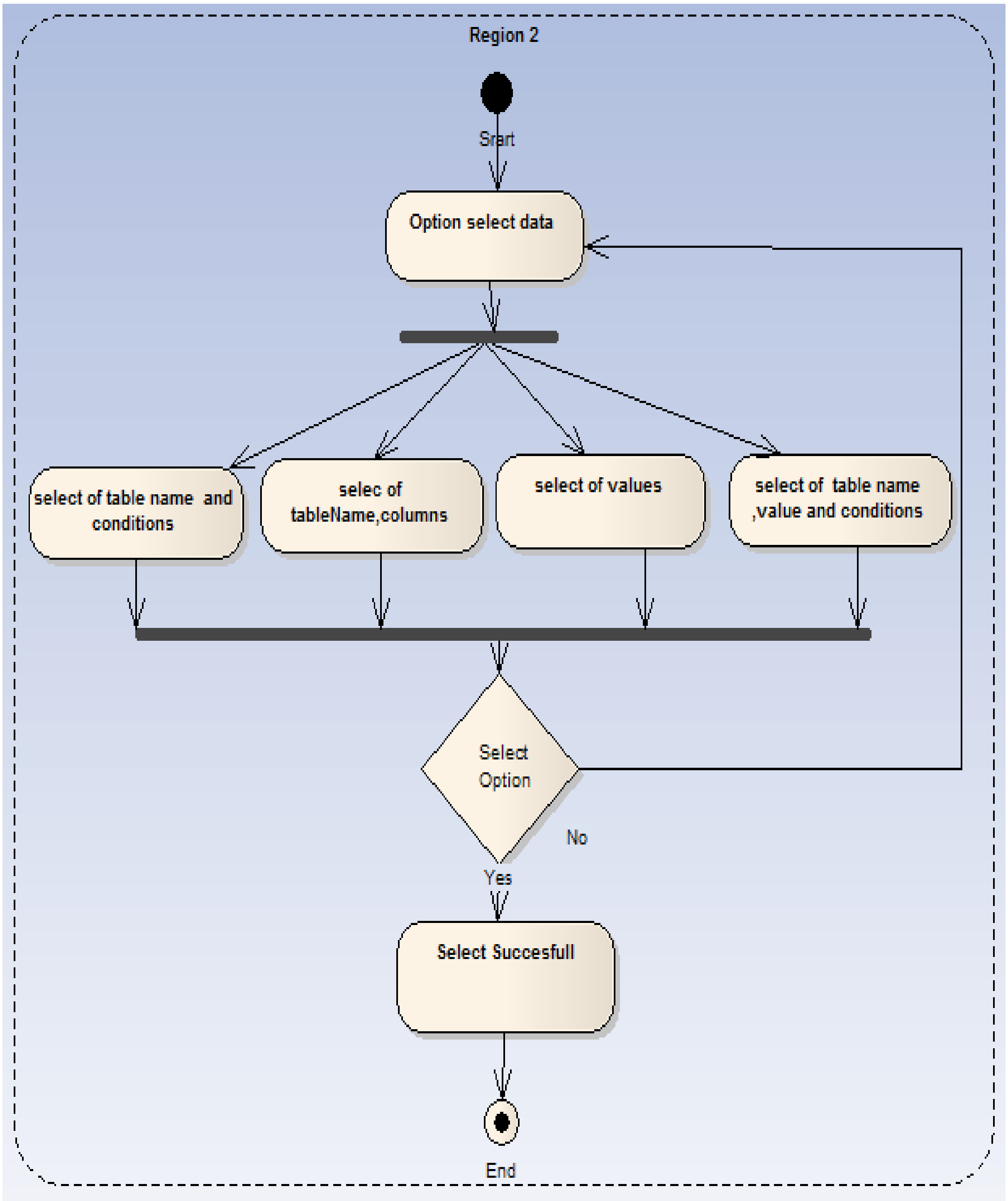

Region 2
Start
Option select data
select of table name and conditions
selec of tableName,columns
select of values
select of table name ,value and conditions
Select Option
No
Yes
Select Succesfull
End

Figure : Illustrates the Activity of process to select.

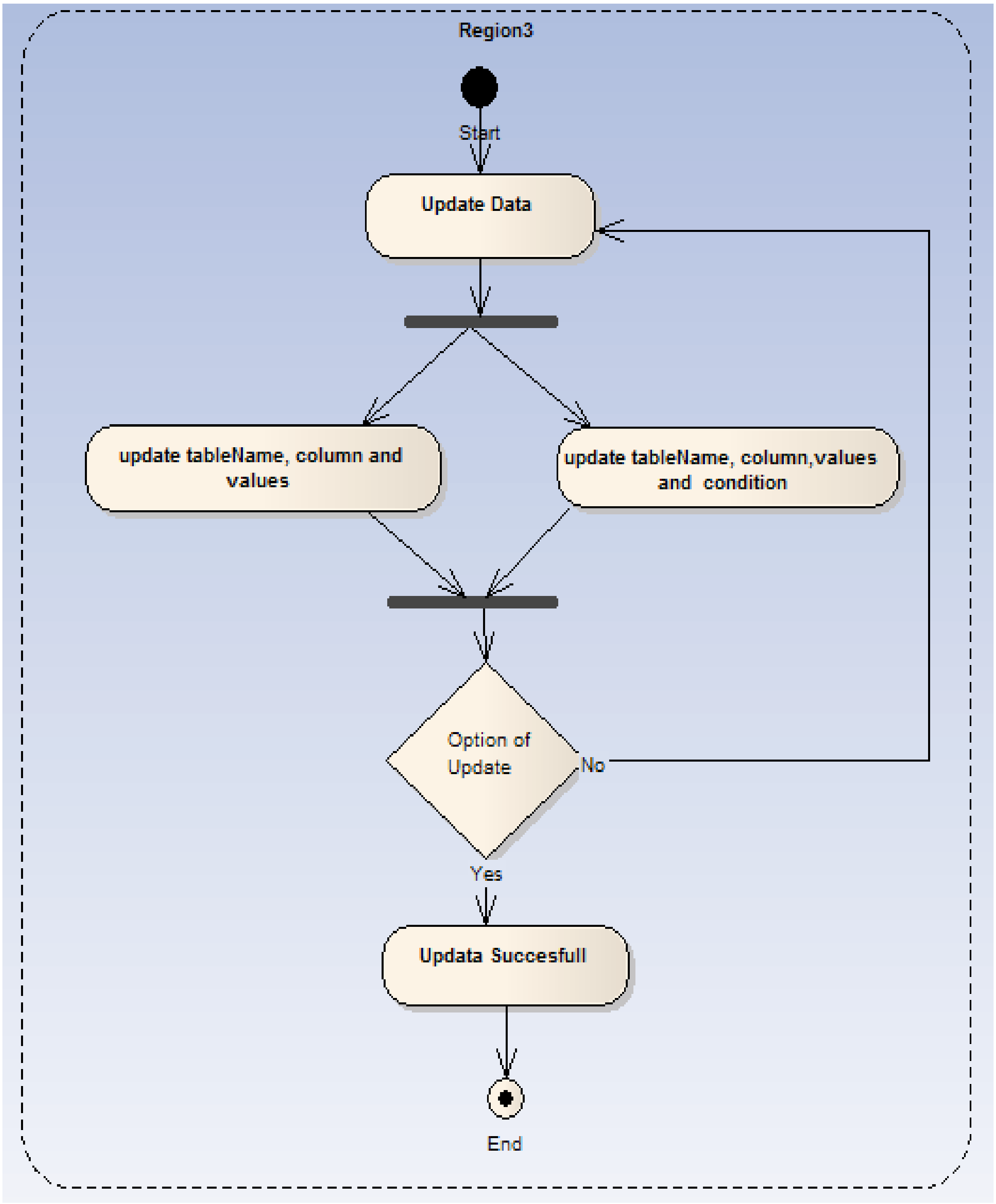

Region3
Start
Update Data
update tableName, column and values
update tableName, column,values and condition
Option of Update
No
Yes
Updata Succesfull
End

Figure : Illustrates the Activity of process to update.

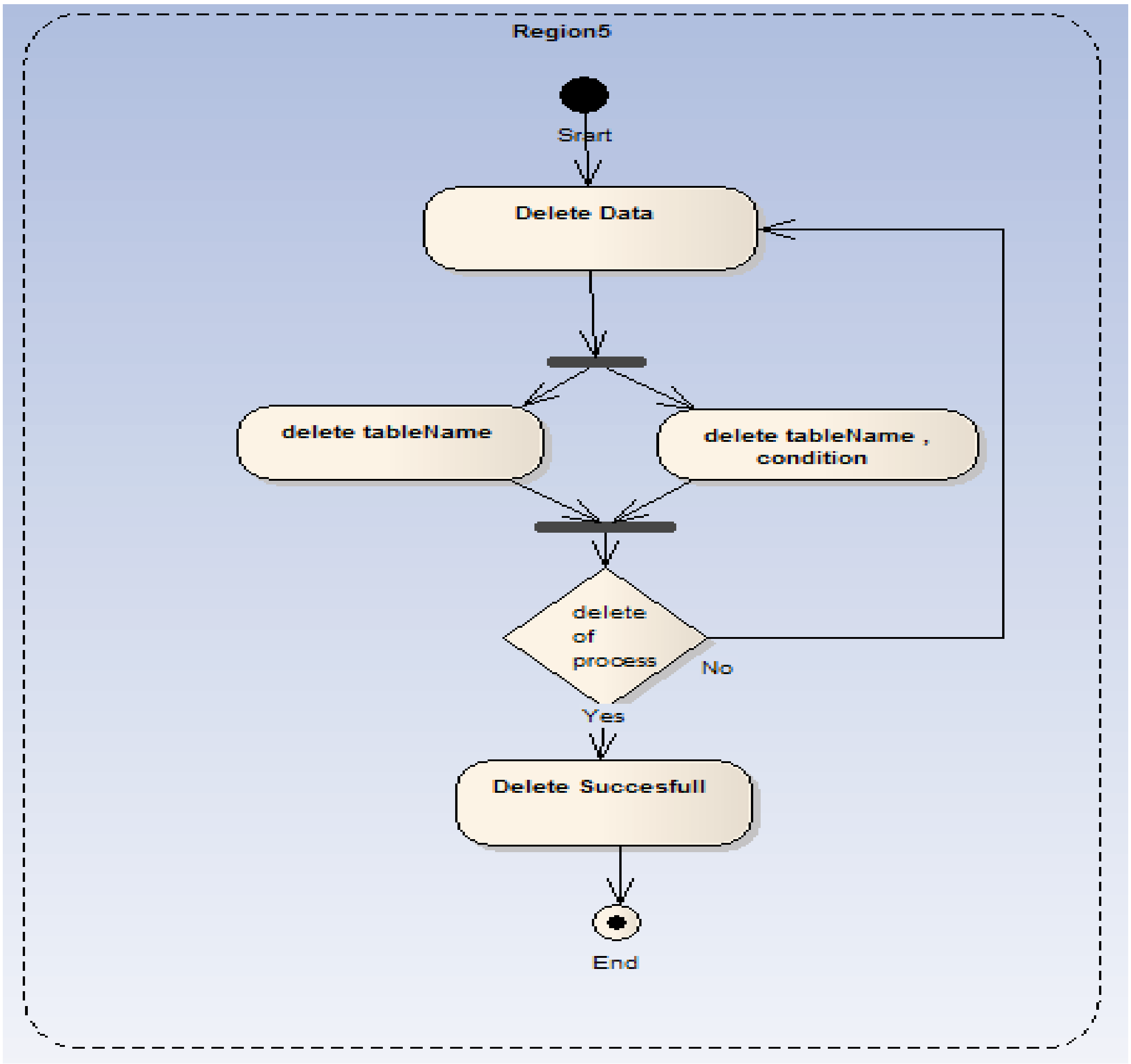


Figure : Illustrates the Activity of process to delete.

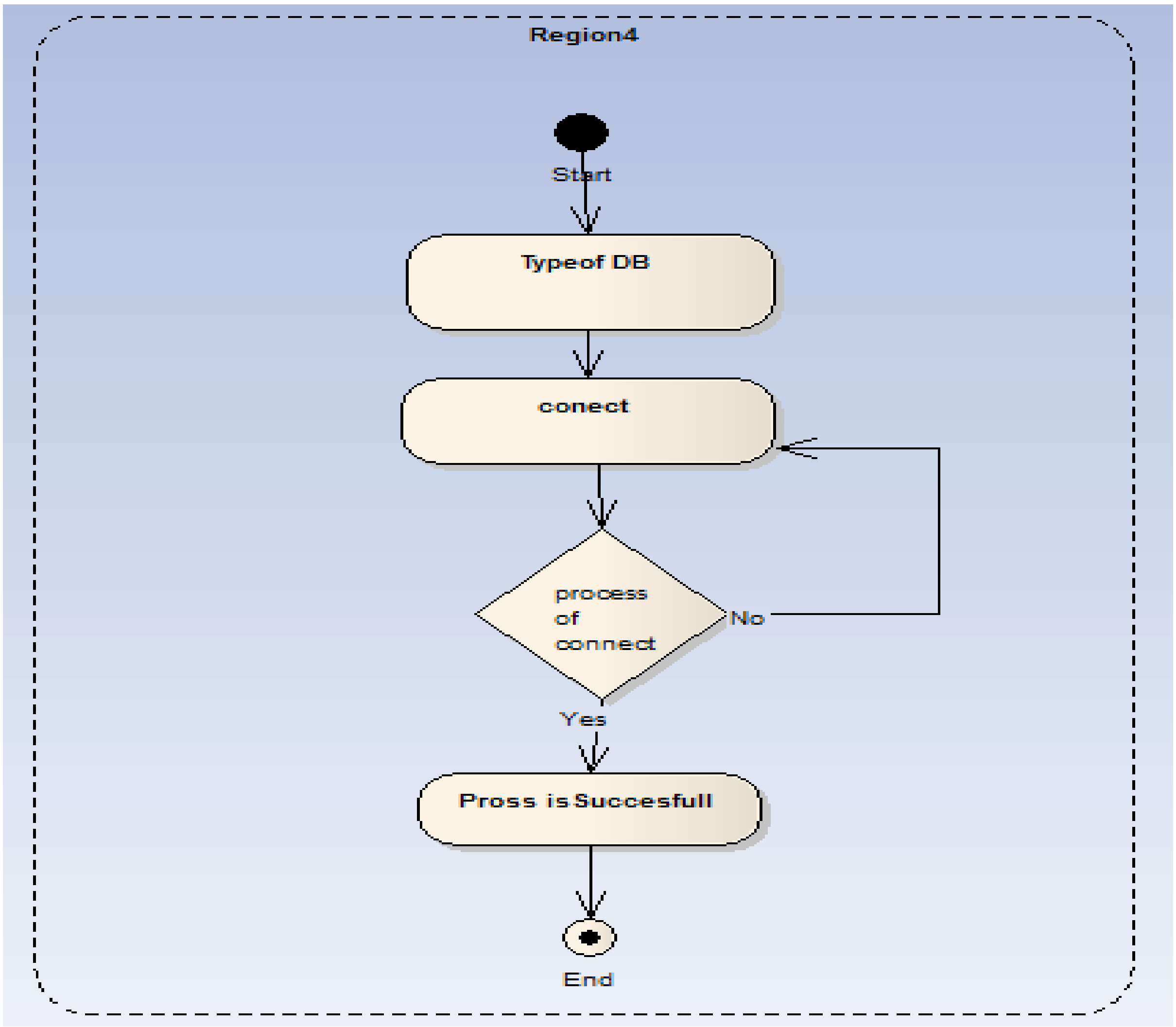


Figure Illustrates the Activity of process to connect.

4.5 CLASS DIAGRAM

The class diagram describes the attributes and operations of all this classes' main class Sql and dbConnection, MySQL, Oracle, and Sql Server).

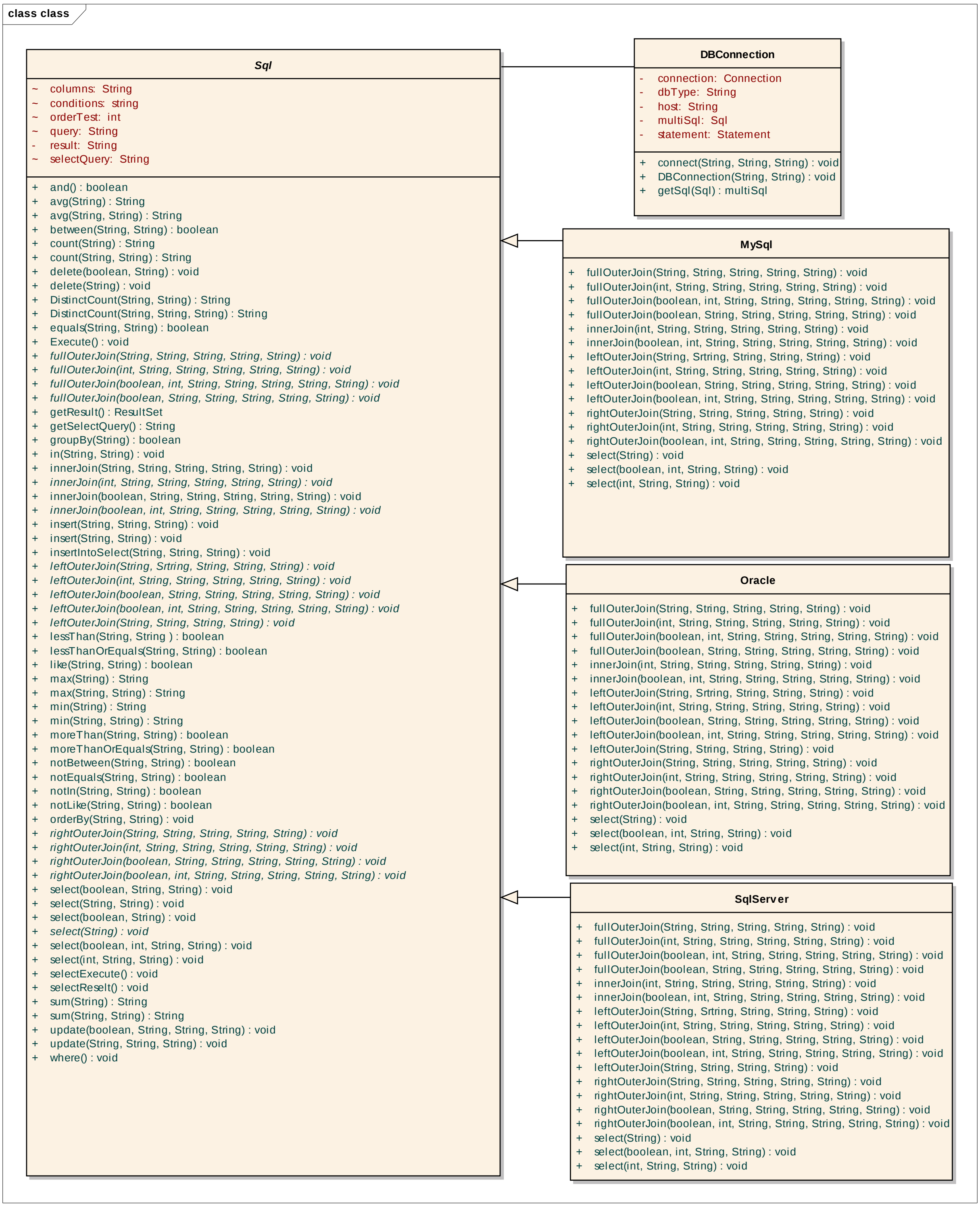

class class
Sql
~ columns: String
~ conditions: string
~ orderTest: int
~ query: String
- result: String
~ selectQuery: String
+ and() : boolean
+ avg(String) : String
+ avg(String, String) : String
+ between(String, String) : boolean
+ count(String) : String
+ count(String, String) : String
+ delete(boolean, String) : void
+ delete(String) : void
+ DistinctCount(String, String) : String
+ DistinctCount(String, String, String) : String
+ equals(String, String) : boolean
+ Execute() : void
+ fullOuterJoin(String, String, String, String, String) : void
+ fullOuterJoin(int, String, String, String, String, String) : void
+ fullOuterJoin(boolean, int, String, String, String, String, String) : void
+ fullOuterJoin(boolean, String, String, String, String, String) : void
+ getResult() : ResultSet
+ getSelectQuery() : String
+ groupBy(String) : boolean
+ in(String, String) : void
+ innerJoin(String, String, String, String, String) : void
+ innerJoin(int, String, String, String, String, String) : void
+ innerJoin(boolean, String, String, String, String, String) : void
+ innerJoin(boolean, int, String, String, String, String, String) : void
+ insert(String, String, String) : void
+ insert(String, String) : void
+ insertIntoSelect(String, String, String) : void
+ leftOuterJoin(String, Srtring, String, String, String) : void
+ leftOuterJoin(int, String, String, String, String, String) : void
+ leftOuterJoin(boolean, String, String, String, String, String) : void
+ leftOuterJoin(boolean, int, String, String, String, String, String) : void
+ leftOuterJoin(String, String, String, String) : void
+ lessThan(String, String) : boolean
+ lessThanOrEquals(String, String) : boolean
+ like(String, String) : boolean
+ max(String) : String
+ max(String, String) : String
+ min(String) : String
+ min(String, String) : String
+ moreThan(String, String) : boolean
+ moreThanOrEquals(String, String) : boolean
+ notBetween(String, String) : boolean
+ notEquals(String, String) : boolean
+ notIn(String, String) : boolean
+ notLike(String, String) : boolean
+ orderBy(String, String) : void
+ rightOuterJoin(String, String, String, String, String) : void
+ rightOuterJoin(int, String, String, String, String, String) : void
+ rightOuterJoin(boolean, String, String, String, String, String) : void
+ rightOuterJoin(boolean, int, String, String, String, String, String) : void
+ select(boolean, String, String) : void
+ select(String, String) : void
+ select(boolean, String) : void
+ select(String) : void
+ select(boolean, int, String, String) : void
+ select(int, String, String) : void
+ selectExecute() : void
+ selectReselt() : void
+ sum(String) : String
+ sum(String, String) : String
+ update(boolean, String, String, String) : void
+ update(String, String, String) : void
+ where() : void
DBConnection
- connection: Connection
- dbType: String
- host: String
- multiSql: Sql
- statement: Statement
+ connect(String, String, String) : void
+ DBConnection(String, String) : void
+ getSql(Sql) : multiSql
MySql
+ fullOuterJoin(String, String, String, String, String) : void
+ fullOuterJoin(int, String, String, String, String, String) : void
+ fullOuterJoin(boolean, int, String, String, String, String, String) : void
+ fullOuterJoin(boolean, String, String, String, String, String) : void
+ innerJoin(int, String, String, String, String, String) : void
+ innerJoin(boolean, int, String, String, String, String, String) : void
+ leftOuterJoin(String, Srtring, String, String, String) : void
+ leftOuterJoin(int, String, String, String, String, String) : void
+ leftOuterJoin(boolean, String, String, String, String, String) : void
+ leftOuterJoin(boolean, int, String, String, String, String, String) : void
+ rightOuterJoin(String, String, String, String, String) : void
+ rightOuterJoin(int, String, String, String, String, String) : void
+ rightOuterJoin(boolean, int, String, String, String, String, String) : void
+ select(String) : void
+ select(boolean, int, String, String) : void
+ select(int, String, String) : void
Oracle
+ fullOuterJoin(String, String, String, String, String) : void
+ fullOuterJoin(int, String, String, String, String, String) : void
+ fullOuterJoin(boolean, int, String, String, String, String, String) : void
+ fullOuterJoin(boolean, String, String, String, String, String) : void
+ innerJoin(int, String, String, String, String, String) : void
+ innerJoin(boolean, int, String, String, String, String, String) : void
+ leftOuterJoin(String, Srtring, String, String, String) : void
+ leftOuterJoin(int, String, String, String, String, String) : void
+ leftOuterJoin(boolean, String, String, String, String, String) : void
+ leftOuterJoin(boolean, int, String, String, String, String, String) : void
+ leftOuterJoin(String, String, String, String) : void
+ rightOuterJoin(String, String, String, String, String) : void
+ rightOuterJoin(int, String, String, String, String, String) : void
+ rightOuterJoin(boolean, String, String, String, String, String) : void
+ rightOuterJoin(boolean, int, String, String, String, String, String) : void
+ select(String) : void
+ select(boolean, int, String, String) : void
+ select(int, String, String) : void
SqlServer
+ fullOuterJoin(String, String, String, String, String) : void
+ fullOuterJoin(int, String, String, String, String, String) : void
+ fullOuterJoin(boolean, int, String, String, String, String, String) : void
+ fullOuterJoin(boolean, String, String, String, String, String) : void
+ innerJoin(int, String, String, String, String, String) : void
+ innerJoin(boolean, int, String, String, String, String, String) : void
+ leftOuterJoin(String, Srtring, String, String, String) : void
+ leftOuterJoin(int, String, String, String, String, String) : void
+ leftOuterJoin(boolean, String, String, String, String, String) : void
+ leftOuterJoin(boolean, int, String, String, String, String, String) : void
+ leftOuterJoin(String, String, String, String) : void
+ rightOuterJoin(String, String, String, String, String) : void
+ rightOuterJoin(int, String, String, String, String, String) : void
+ rightOuterJoin(boolean, String, String, String, String, String) : void
+ rightOuterJoin(boolean, int, String, String, String, String, String) : void
+ select(String) : void
+ select(boolean, int, String, String) : void
+ select(int, String, String) : void

Figure : Illustrates the class to DBAL.

CHAPTER 5
EXECUTION

CHAPTER 5EXECUTION

5.1 INTRODUCTION

In this Chapter we will know how to use the library, and how to deal with it. As we explained previously, the library used as abstraction layer between the database and the program , it will make the developer apple to connect to database and selecting, inserting, deleting and updating data in a database.

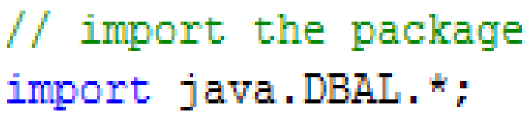
5.2 INSTALLATION

To use the library:

- You must have java JDK 1.8.
- Put the jar file of library (“DBAL”) in path “Java\jdk1.8.0_66\jre\lib\ext”

5.3 USES THE LIBRARY

5.3.1 **Import library (package)**: Illustrates the



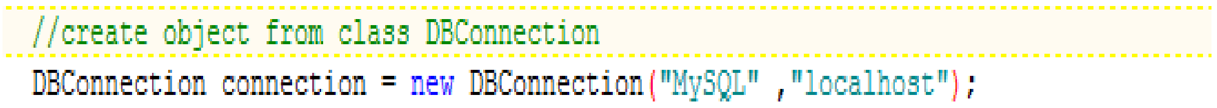
```
// import the package
import java.DBAL.*;
```

Figure Illustrates Import Library

5.3.2 Connect to database

You must create object from class java.DBAL.DBConnection refers to the same class, and pass to Constructor the type of database and host, the type of database have three options.

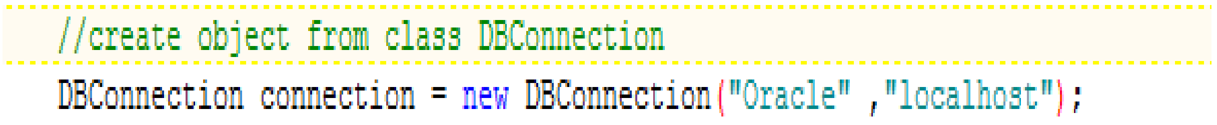
- **For MySql, the data type will be “MySQL”**



```
//create object from class DBConnection
DBConnection connection = new DBConnection("MySQL" ,"localhost");
```

Figure : Illustrates the Connect Of MySql

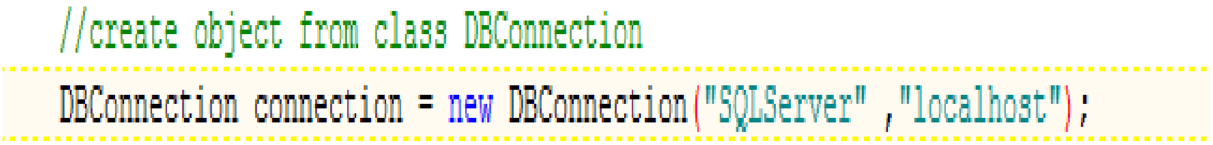
- **For oracle , the data type will be “Oracle”**



```
//create object from class DBConnection
DBConnection connection = new DBConnection("Oracle" ,"localhost");
```

Figure Illustrates the Connect Of Oracle

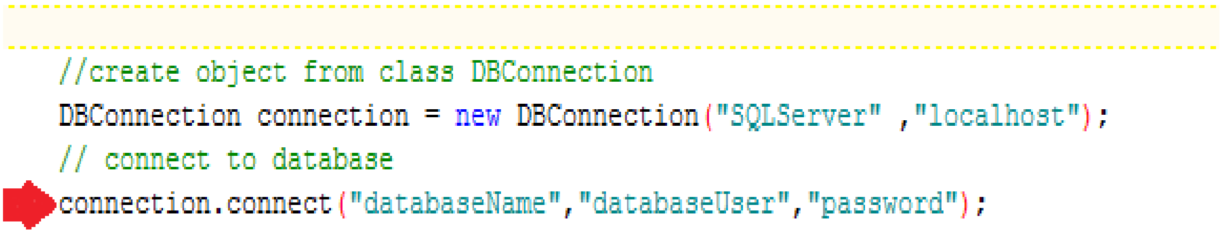
- **For Sql server , the data type will be “SQLServer”**



```
//create object from class DBConnection
DBConnection connection = new DBConnection("SQLServer" ,"localhost");
```

Figure Illustrates the Connect Of SqlServer

Call the method connect (), that existing in class DBConnection, and pass to it database name, database user name and password, after calling the method connect (), it has been physically connect to the database.



```
//create object from class DBConnection
DBConnection connection = new DBConnection("SQLServer" ,"localhost");
// connect to database
connection.connect("databaseName","databaseUser","password");
```

Figure : Illustrates Method Connect ().

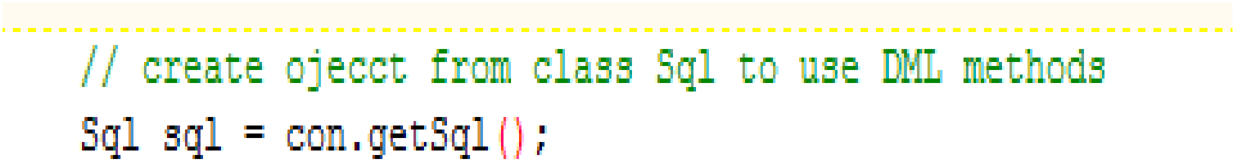
5.3.3 Disconnect

To disconnect call the method disconnect (), that existing in class DBConnection.

5.3.4 DML methods (insert , select , update , delete)

- **Use the DML methods**

Create object from class java.DBAL.Sql refers to the object that returned by Method getSql () .this method existing in class java.DBAL.DBConenction.



```
// create ojecct from class Sql to use DML methods
Sql sql = con.getSql();
```

Figure : Illustrates Method getSql ().

After the previous step, you can use all DML methods by the object from class java.DBAL.Sql.

To use DBAL method you must use columns array and values arrayDepending on the type of operation. The columns and values array existing in Sql object .Both columns and values array are objects from class Array List .If the columns or values was empty, that occur an runtime error.

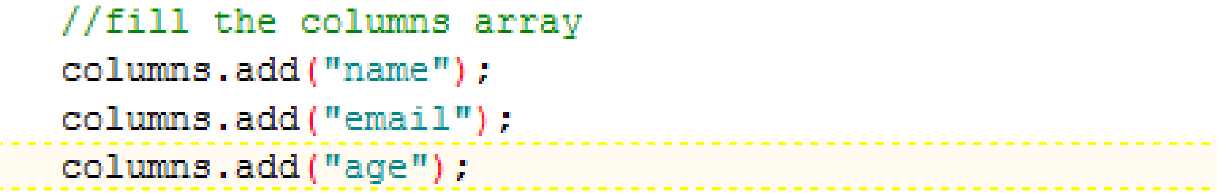
The columns and values clear the data after execute any DML method (insert, select, update, delete).

- **Columns array**

Columns array fills by using add () or addColumns (), last methods existing in java.DBAL.Sql .

Columns array fills just by String values.

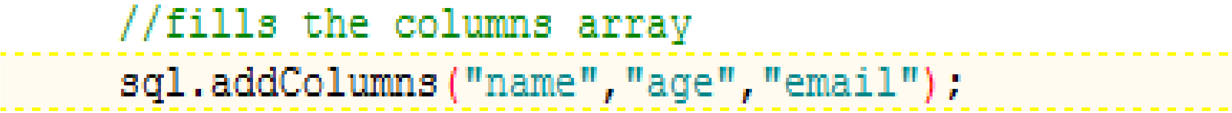
Use the method add () to add single column.



```
//fill the columns array
columns.add("name");
columns.add("email");
columns.add("age");
```

Figure : Illustrates add().

Use the method addColumns() to add group of columns.



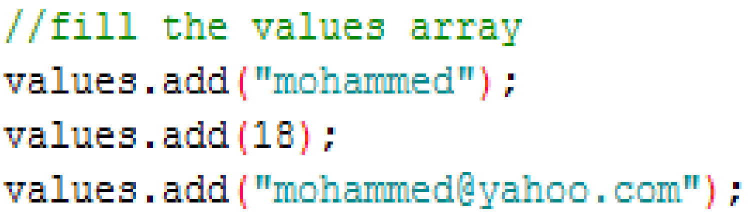
```
//fills the columns array
sql.addColumns("name","age","email");
```

Figure : Illustrates addColumns ().

- **Values array**

Values array fills by using add()values array , and it fills by any values (String , Numeric , BLOB , CLOB , Boolean , Date) , in values array , you can add single value in time .

Use the method add() .



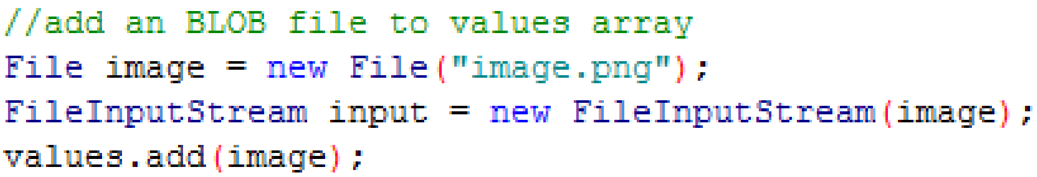
```
//fill the values array
values.add("mohammed");
values.add(18);
values.add("mohammed@yahoo.com");
```

Figure Illustrates method add() ,Values.

Add BLOB file

BLOB file can be image ,video ,audio, document ...etc .You can add it with add() method .

Use the method add() to add a single BLOB file .



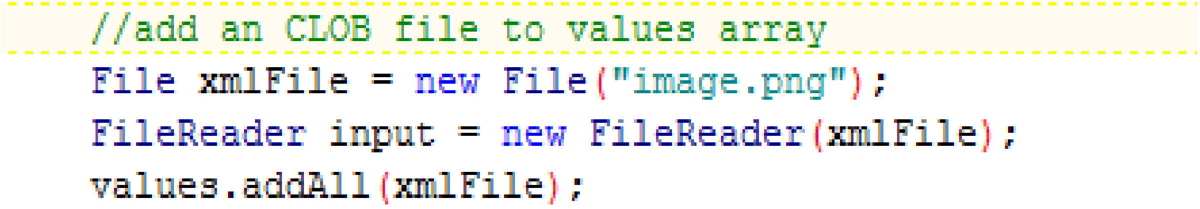
```
//add an BLOB file to values array
File image = new File("image.png");
FileInputStream input = new FileInputStream(image);
values.add(image);
```

Figure Illustrates BLOB File

Add CLOB file

CLOB file is a large text document like xml files , books (pdf) …etc .You can add it with add() method .

Use add() method to add a single CLOB file



```
//add an CLOB file to values array
File xmlFile = new File("image.png");
FileReader input = new FileReader(xmlFile);
values.addAll(xmlFile);
```

Figure:IllustratesCLOB File

5.3.5 SQL methods

- SQL has many built-in functions for performing calculations on data.
- The library provides this method.
- The SQL methods return Boolean value.

The SQL methods can be divided into three types depending on used:

1. **SQL Aggregate methods**

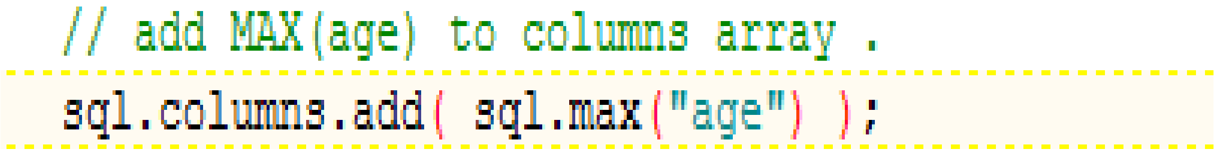
SQL aggregate methods return a single value, calculated from values in a column, it used on columns or conditions columns (23) .

- **Max() method**

  The MAX() method returns the largest value of the selected column

  It is takes max(String columnName).

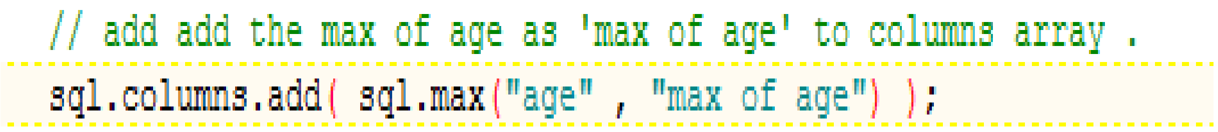
Use max() method



```
// add MAX(age) to columns array .
sql.columns.add( sql.max("age") );
```

Figure : Illustrates Max()

Use max() method and rename the column .



```
// add add the max of age as 'max of age' to columns array .
sql.columns.add( sql.max("age" , "max of age") );
```

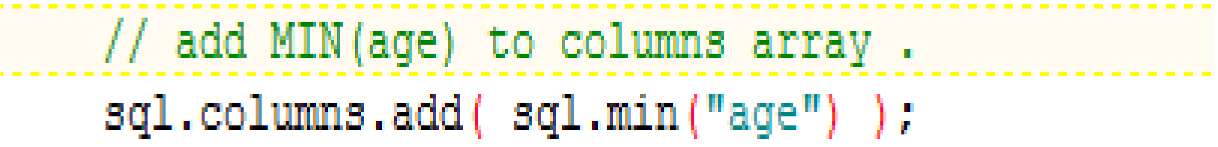
Figure : Illustrates rename Column max ().

- **Min() method**

  The MIN() method returns the smallest value of the selected column.

  It’s takes min(String columnName)

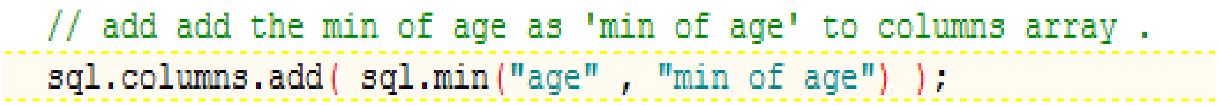
  Use min() method.



```
// add MIN(age) to columns array .
sql.columns.add( sql.min("age") );
```

Figure : Illustrates min ().

use min() method and rename the column.



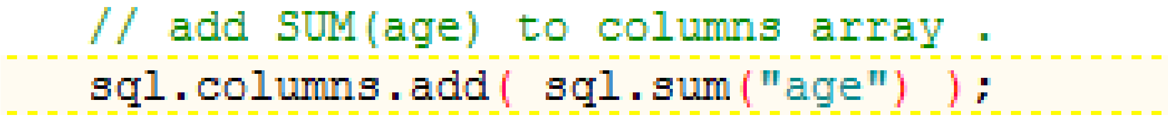
```
// add add the min of age as 'min of age' to columns array .
sql.columns.add( sql.min("age" , "min of age") );
```

Figure Illustrates rename Columnmin ().

- **Sum() method**

  The SUM() function returns the total sum of a numeric column.
  It’s take sum(String columnName)

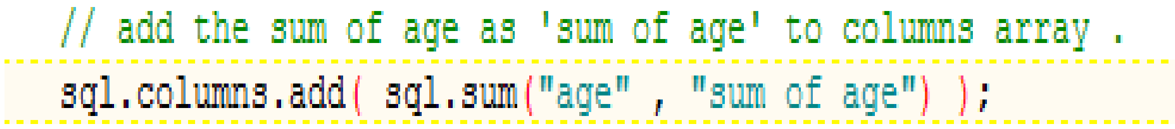
Use sum() method



```
// add SUM(age) to columns array .
sql.columns.add( sql.sum("age") );
```

Figure : Illustrates Sum ().

Use sum() method an rename the column .



```
// add the sum of age as 'sum of age' to columns array .
sql.columns.add( sql.sum("age" , "sum of age") );
```

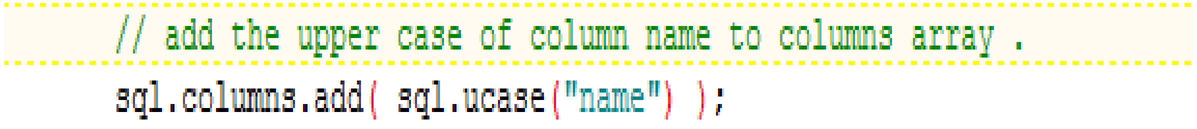
Figure : Illustrates rename column Sum ().

- **Ucase() method**

  The ucase() method converts the value of a field to uppercase.

  It’s take ucase(String columnName)

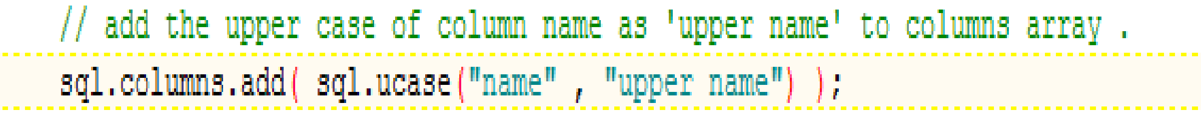
Use ucase() method



```
// add the upper case of column name to columns array .
sql.columns.add( sql.ucase("name") );
```

Figure Illustrates case().

Use ucase() method and rename the column .



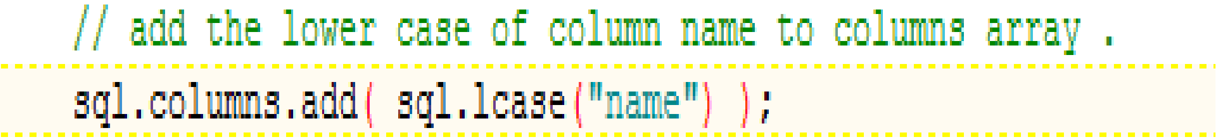
```
// add the upper case of column name as 'upper name' to columns array .
sql.columns.add( sql.ucase("name" , "upper name") );
```

Figure : Illustrates renameucase().

- **Lcase() method**

The lcase() method converts the value of a field to lowercase.
It’s take lcase(String columnName);

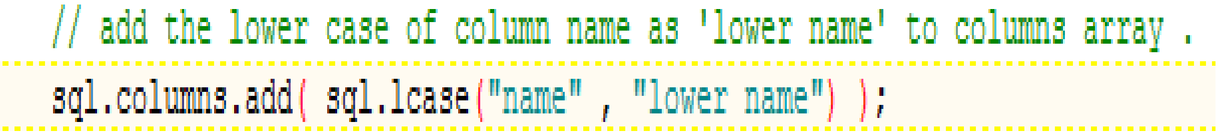
Use lcase() method



```
// add the lower case of column name to columns array .
sql.columns.add( sql.lcase("name") );
```

Figure : Illustrates lcase ().

Use lcase() method and rename the column .

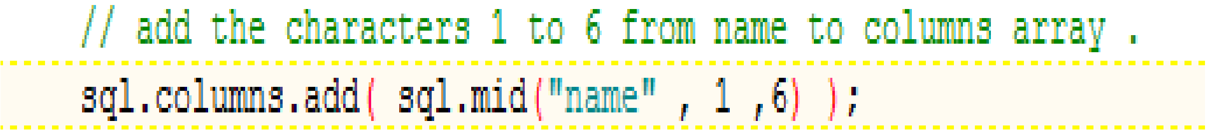


```
// add the lower case of column name as 'lower name' to columns array .
sql.columns.add( sql.lcase("name" , "lower name") );
```

Figure : Illustrates rename lcase

- **Mid() method**

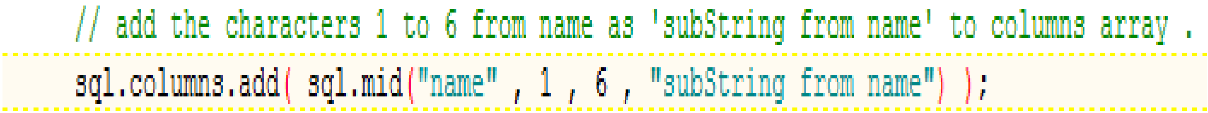
The mid() method is used to extract characters from a text field.
It’s take mid(String columnName , intstartIndex , intendIndex
Use mid() method



```
// add the characters 1 to 6 from name to columns array .
sql.columns.add( sql.mid("name" , 1 ,6) );
```

Figure Illustrates mid()

Use mid() method with rename column .

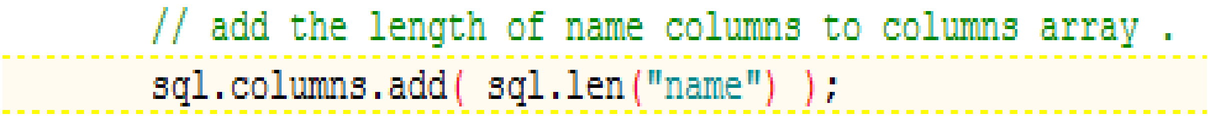


```
// add the characters 1 to 6 from name as 'subString from name' to columns array .
sql.columns.add( sql.mid("name" , 1 , 6 , "subString from name") );
```

Figure Illustrates rename mid().

- **Len() method**

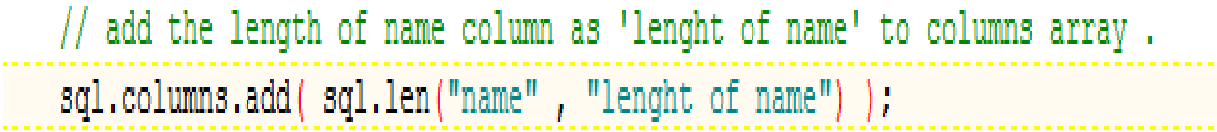
The len() method returns the length of the value in a text field.
It’s takes len(String columnName).
Use len() method



```
// add the length of name columns to columns array .
sql.columns.add( sql.len("name") );
```

Figure : Illustrates len().

Use len method and rename the column.

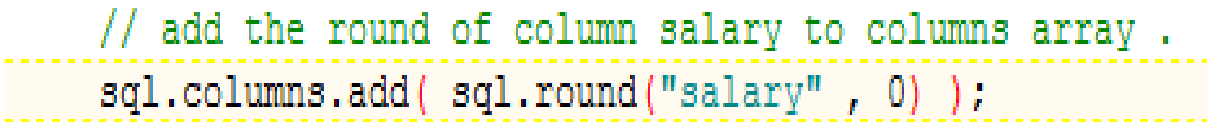


```
// add the length of name column as 'lenght of name' to columns array .
sql.columns.add( sql.len("name" , "lenght of name") );
```

Figure :Illustrates rename len().

- **round()method**

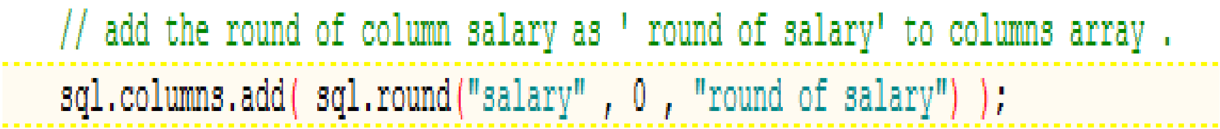
The round() round is used to round a numeric field to the number of decimals specified.
It’s take round(String columnName , intdecimals)
Use round() method.



```
// add the round of column salary to columns array .
sql.columns.add( sql.round("salary" , 0) );
```

Figure : Illustrates round().

Use round() method and rename column.



```
// add the round of column salary as ' round of salary' to columns array .
sql.columns.add( sql.round("salary" , 0 , "round of salary") );
```

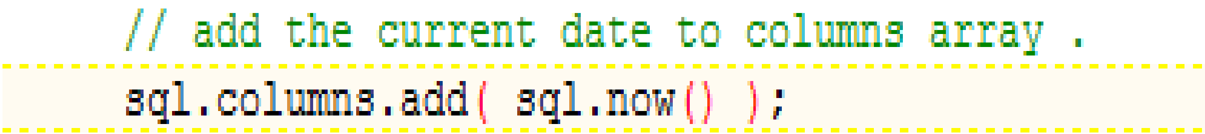
Figure : Illustrates rename round().

- **Now()method**

The now() function returns the current system date and time.

It's take no parameters now()

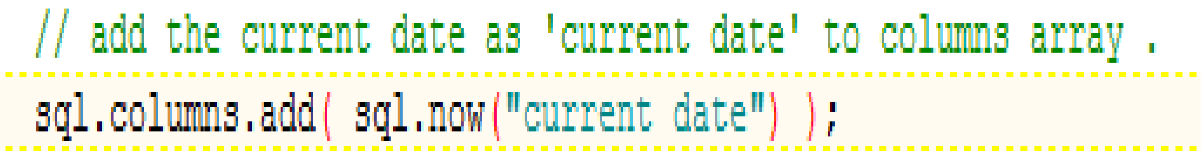
Use now() method .



```
// add the current date to columns array .
sql.columns.add( sql.now() );
```

Figure : Illustrates method now().

Use now() method and rename the column .



```
// add the current date as 'current date' to columns array .
sql.columns.add( sql.now("current date") );
```

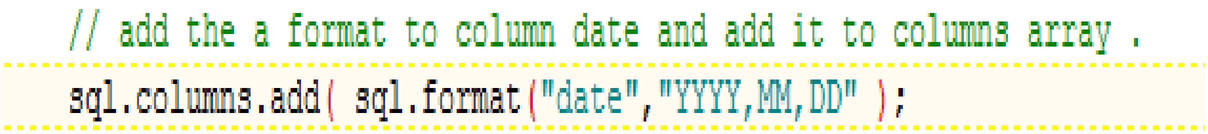
Figure :Illustrates rename now().

- **Format()method**

  The format() method is used to format how a field is to be displayed.

  It's take format(String columnName , String format

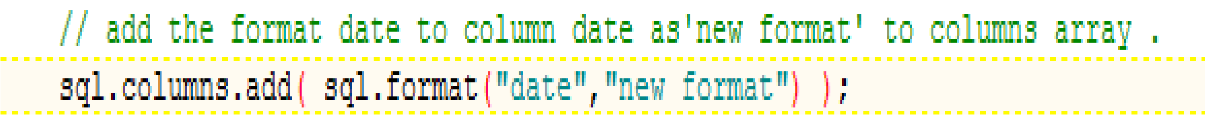
Use format() method.



```
// add the a format to column date and add it to columns array .
sql.columns.add( sql.format("date","YYYY,MM,DD" );
```

Figure : Illustrates format().

Use format() method and rename the column .



```
// add the format date to column date as'new format' to columns array .
sql.columns.add( sql.format("date","new format") );
```

Figure : Illustrates renameformat().

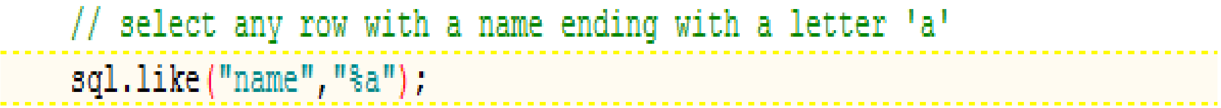
2. **SQL conditions methods**

- It is used to filter records
- Condition methods existing in object of class Sql.

- **like() method**

Use to search for a specified pattern in a column.It take like(String columnName , anyValue value)

Use like() method



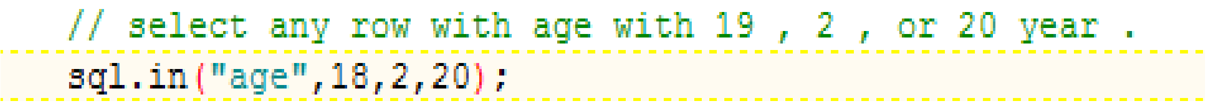
```
// select any row with a name ending with a letter 'a'
sql.like("name","%a");
```

Figure : Illustrates like().

- **in() method**

in() method allows you to specify multiple values in a WHERE clause.It take in (String columnName, anyValue… value).

Use in() method



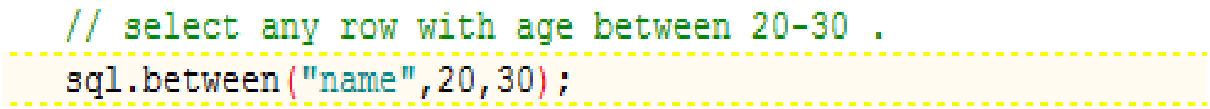
```
// select any row with age with 19 , 2 , or 20 year .
sql.in("age",18,2,20);
```

Figure : Illustrates in().

- **between method()**

The between() method is used to select values within a range.
It take between(String columnName, anyValuestartOfRange , anyValueendOfRange)

Use between() method



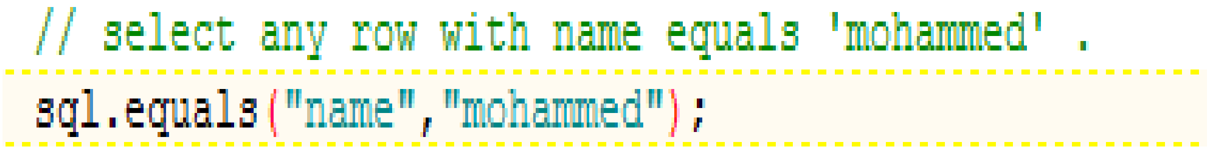
```
// select any row with age between 20-30 .
sql.between("name",20,30);
```

Figure : Illustrates between().

- **equals method()**

Equals method use to check if column value equals a specific value .it take the column name and the specific value

use equals() method

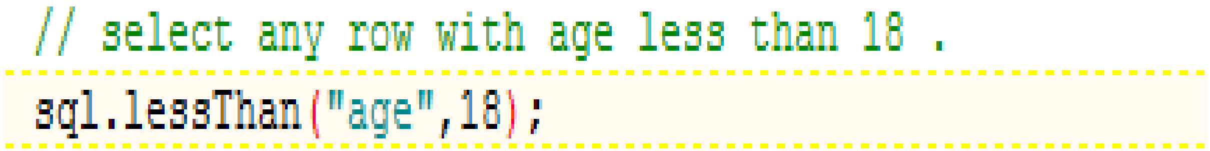


```
// select any row with name equals 'mohammed' .
sql.equals("name","mohammed");
```

Figure : Illustrates equals().

- **lessThan() method**

lessThan() method use to check if column value less than a specific value.it take the lessThan(String columnName , anyValue value).



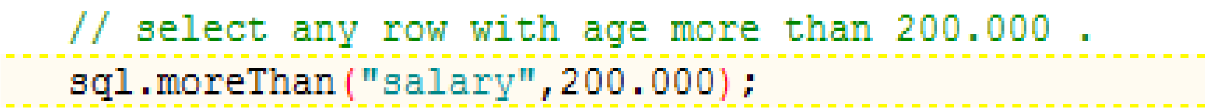
```
// select any row with age less than 18 .
sql.lessThan("age",18);
```

Figure : Illustrates lessTan().

- **moreThan() method**

moreThan() method use to check if column value more than a specific value.it take moreThan(String columnName , anyValue value).

usemoreThan() method



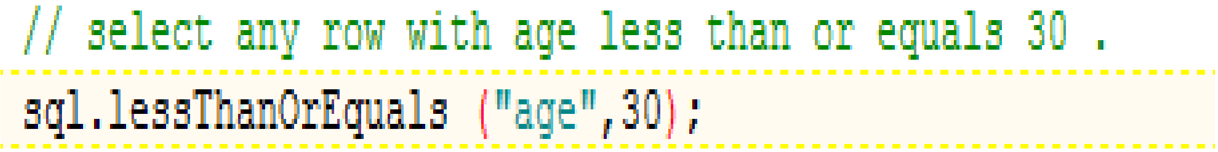
```
// select any row with age more than 200.000 .
sql.moreThan("salary",200.000);
```

Figure : Illustrates more than().

- **lessThanOrEquals() method**

lessThanOrEquals() method use to check if column value less than or equals a specific value .it take lessThanOrEquals(String columnName , anyValue value).

use lessThanOrEquals() method.



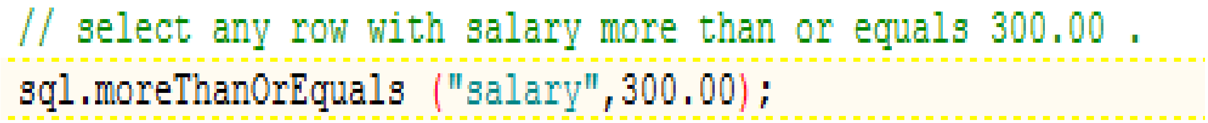
```
// select any row with age less than or equals 30 .
sql.lessThanOrEquals ("age",30);
```

Figure Illustrates lessThanOrEquals().

- **moreThanOrEquals() method**

moreThanOrEquals() method use to check if column value more than or equals a specific value .it take moreThanOrEquals(String columnName , anyValue value).

usemoreThanOrEquals() method



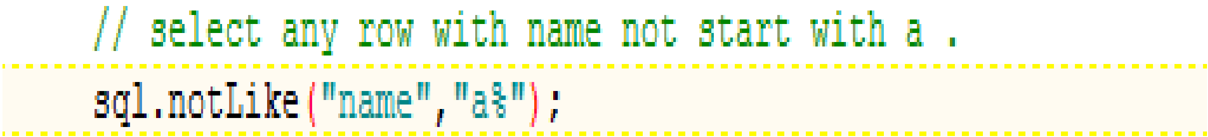
```
// select any row with salary more than or equals 300.00 .
sql.moreThanOrEquals ("salary",300.00);
```

Figure : Illustrates moreThanOrEquals().

- **notLike() method**

notLike () method used to negative like condition.it take notLike(String columnName , anyValue value).

UsenotLike()



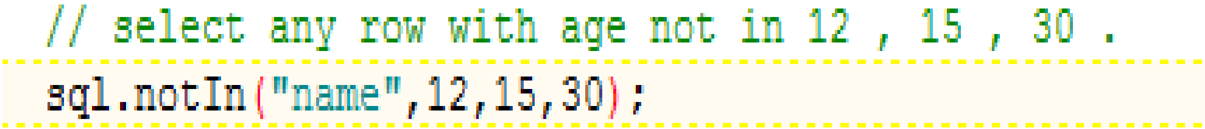
```
// select any row with name not start with a .
sql.notLike("name","a%");
```

Figure : Illustrates notLike().

- **notIn() method**

notIn() method use to negative in condition.it take noIn(String columnName , anyValue… value).

usenotIn() method



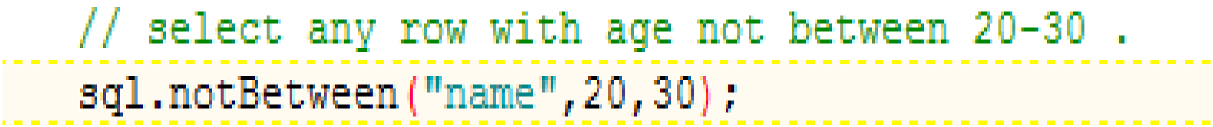
```
// select any row with age not in 12 , 15 , 30 .
sql.notIn("name",12,15,30);
```

Figure : Illustrates notIn().

- **notBetween() method**

notBetween() method use to negative between conditionit take notBetween(String columnName , anyValuestartOfRange , anyValueendOfRangeranage).

UsenotBetween() method



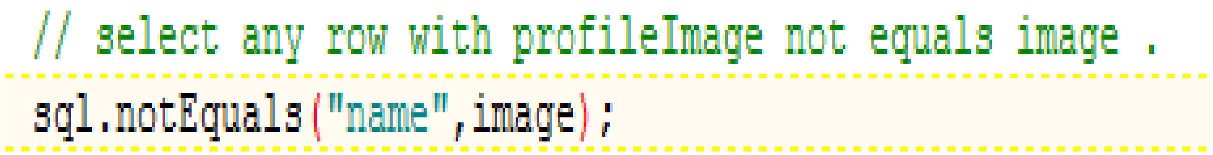
```
// select any row with age not between 20-30 .
sql.notBetween("name",20,30);
```

Figure : Illustrates notBetween().

- **notEquals() method**

notEquals() method use to check if column value not equals a specific value.It takes notEquals(String columnName , anyValue value).

Use notEquals() method

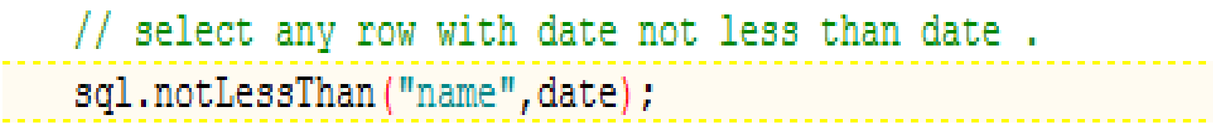


```
// select any row with profileImage not equals image .
sql.notEquals("name",image);
```

Figure : Illustrates notEquals().

- **notLessThan() method**

notLessThan () method use to check if column value not less than a specific value.
It takesnotLessThan(String columnName , anyValue value)
Use notLessThan() method



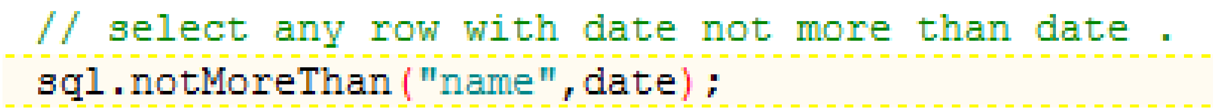
```
// select any row with date not less than date .
sql.notLessThan("name",date);
```

Figure :IllustratesnotlessThan().

- **notMoreThan() method**

notMoreThan () method use to check if column value not more than a specific value.It takes notLessThan(String columnName , anyValue value).

Use notMoreThan () method

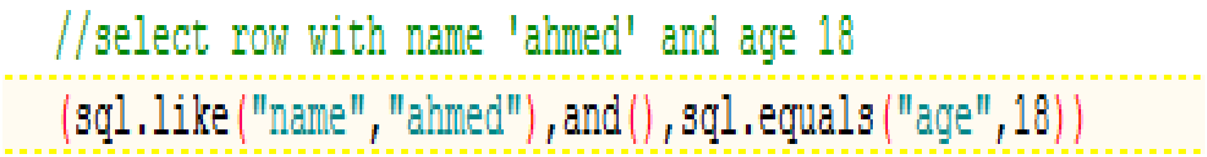


```
// select any row with date not more than date .
sql.notMoreThan("name",date);
```

Figure : Illustrates notMoreThan().

- **and() method**

and() method Used for linking conditions, it’s take no parameters and()
use and() method



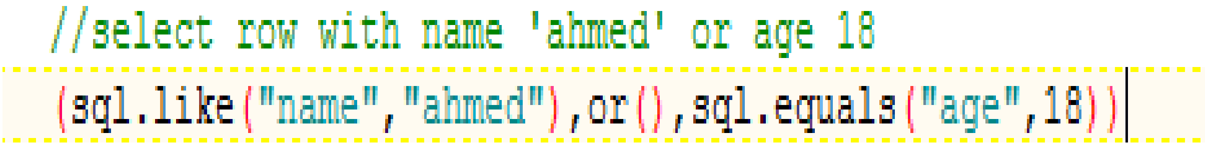
```
//select row with name 'ahmed' and age 18
(sql.like("name","ahmed"),and(),sql.equals("age",18))
```

Figure : Illustrates and().

- **or() method**

or() method Used for linking conditions.it’s take no parameters and().

Use or() method



```
//select row with name 'ahmed' or age 18
(sql.like("name","ahmed"),or(),sql.equals("age",18))
```

Figure : Illustrates or().

3. **Sorting method**

Sorting methods must call and the end of DML method , if it call befor condition that occur an error .

- **OrderBy() method**

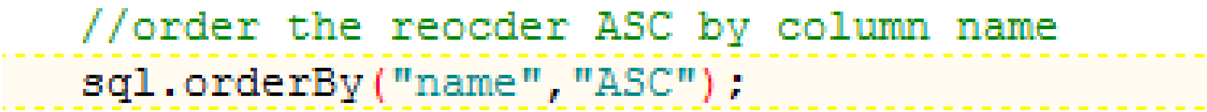
OrderBy() method used to sort the result-set by one or more columns.It’s take OrderBy(String columName , String sortType).

Use OrderBy() method

There are two type of Sort:

- ASC Sort
- DESC Sort

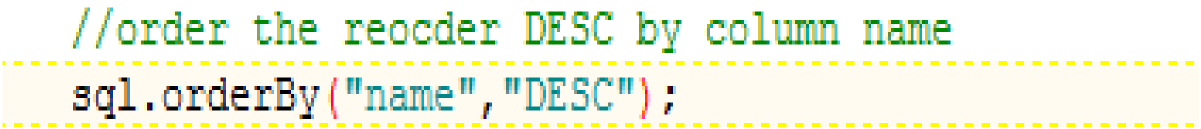
Use ASC sort



```
//order the reocder ASC by column name
sql.orderBy("name","ASC");
```

Figure : IllustratesASC Sort.

Use DESC sort



```
//order the reocder DESC by column name
sql.orderBy("name","DESC");
```

Figure : Illustrates DESC Sort.

- **groupBy() method**

groupBy() method used in conjunction with the aggregate functions to group the result-set by one or more columns.

- **MDL method**

insert() method

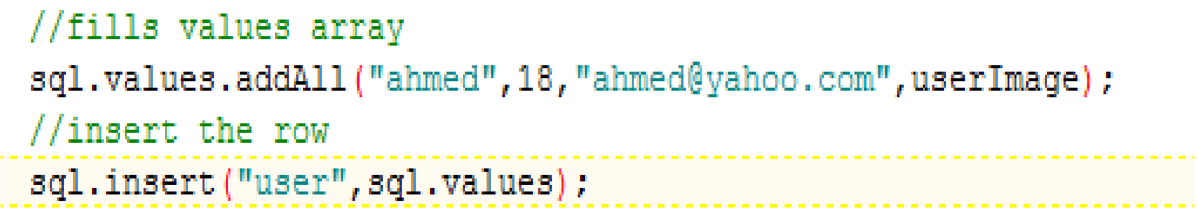
insert() method is used to insert new records in a table.

There are two types of insert method:

1- Insert row with all values

Insert(StraingcolumnName , ArrayList values)

Remember you must values array form class java.DBAL.Sql and must fill it.



```
//fills values array
sql.values.addAll("ahmed",18,"ahmed@yahoo.com",userImage);
//insert the row
sql.insert("user",sql.values);
```

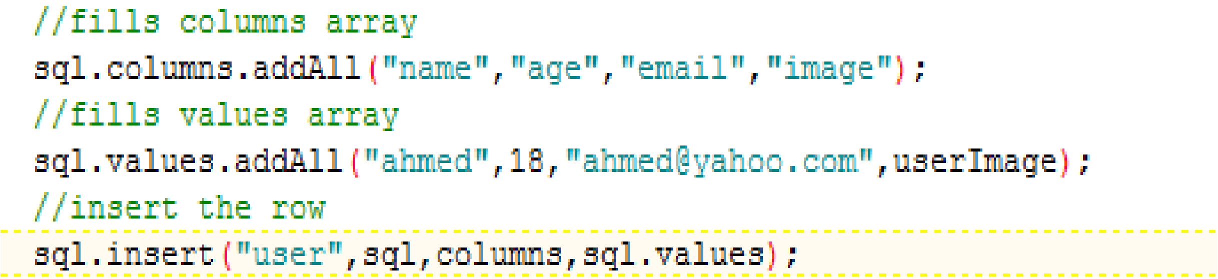
Figure : Illustrates Insert row with all values

Remember you must add values to values array in the same order of the columns of the table .if they don’t have the same order that will occur an error .

2-Insert row with specific values .

insert(String tableName , ArrayList<Stirng> columns , ArrayList<String> values)

Remember you must fill columns and values array first and then call the insert method .



```
//fills columns array
sql.columns.addAll("name","age","email","image");
//fills values array
sql.values.addAll("ahmed",18,"ahmed@yahoo.com",userImage);
//insert the row
sql.insert("user",sql,columns,sql.values);
```

Figure :Illustrates Insertrow with specific values.

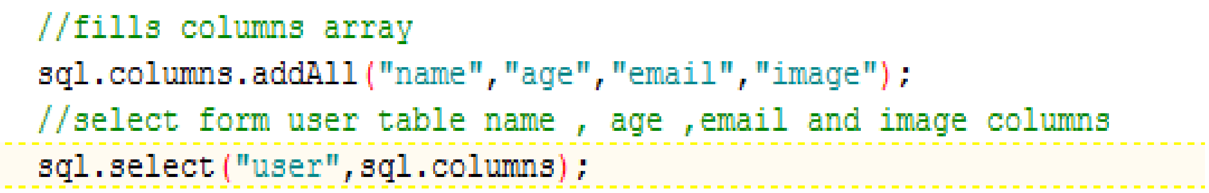
Remember you must add values to values array in the same order of the columns in the columns array .if they don’t have the same order that will occur an error .

select() mothod

select() method used to select data from a database .

1- select columns from table

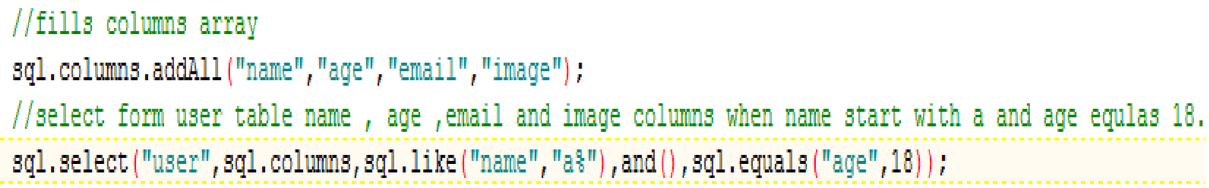
select (StirngtableName , ArrayList<String> columns)



```
//fills columns array
sql.columns.addAll("name","age","email","image");
//select form user table name , age ,email and image columns
sql.select("user",sql.columns);
```

Figure : Illustrates select columns from table

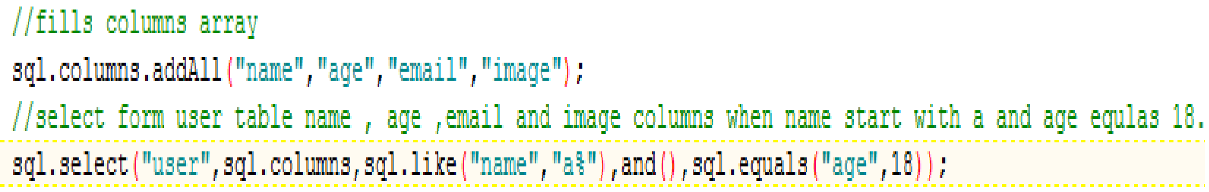
2- select columns form table with condition
select(String tableName , ArrayList<String>columns,boolean… conditions).



```
//fills columns array
sql.columns.addAll("name","age","email","image");
//select form user table name , age ,email and image columns when name start with a and age equlas 18.
sql.select("user",sql.columns,sql.like("name","a%"),and(),sql.equals("age",18));
```

Figure :Illustrates select columns from table with condition

3-select columns from table and return specific number of rows
Select(String tableName , ArrayList<String> columns , int limit).



```
//fills columns array
sql.columns.addAll("name","age","email","image");
//select form user table name , age ,email and image columns when name start with a and age equlas 18.
sql.select("user",sql.columns,sql.like("name","a%"),and(),sql.equals("age",18));
```

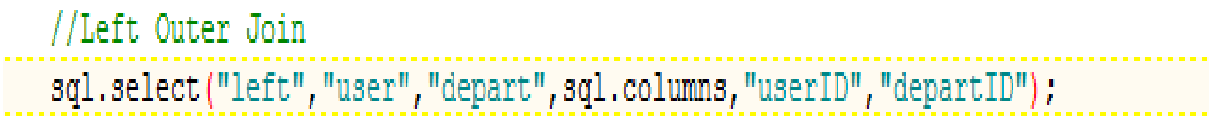
Figure : Illustrates select columns from table specific…

join select
Select(String joinType , String leftTableName , ArrayList<String> columns ,String rightTableName , String firstKey , String secondKey)

There are four type of join

- Left outer join, put the String “left” in parameter joinType to use the left outer join select.
- Right outer join, put the String “right” in parameter joinType to use the right outer join select.
- Full outer join, put the String “full” in parameter joinType to use the full outer join select.
- Inner join, put the String “inner” in parameter joinType to use the inner join select.

Use join select method
In this example we will choice the left outer join

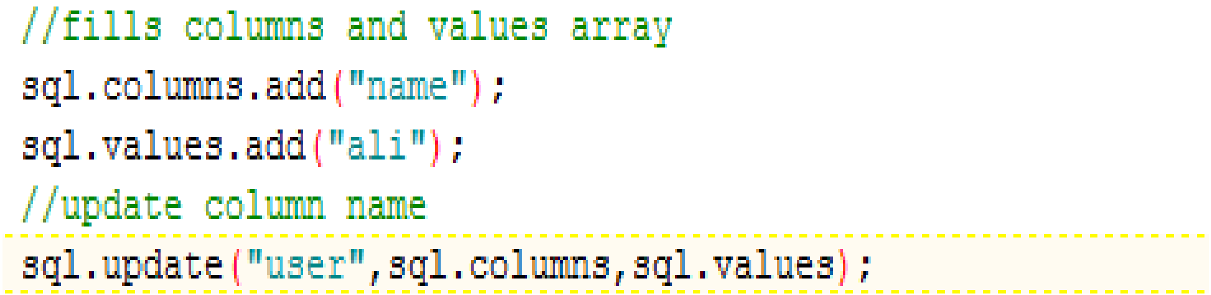


```
//Left Outer Join
sql.select("left","user","depart",sql.columns,"userID","departID");
```

Figure : Illustrates left Outer Join

update() method
update() method is used to update records in a table.
Update(String tableName,ArrayList,<String> columns , ArrayList<String> values>

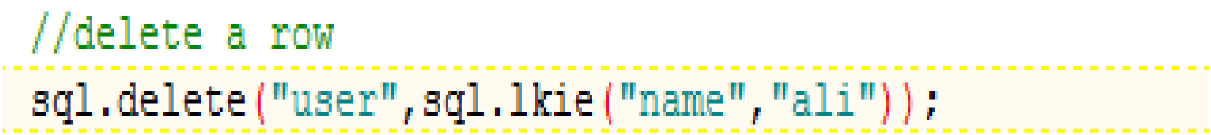
Use update() method



```
//fills columns and values array
sql.columns.add("name");
sql.values.add("ali");
//update column name
sql.update("user",sql.columns,sql.values);
```

Figure : IllustratesUpdate().

delete() method
delete() method is used to delete records in a table.
delete(String tableName , boolean… condtions)
use delete() method



```
//delete a row
sql.delete("user",sql.lkie("name","ali"));
```

Figure : Illustratesdelete().

- **Transaction**

A transaction is a unit of work that is performed against a database.To make a transaction you need to use transaction methods

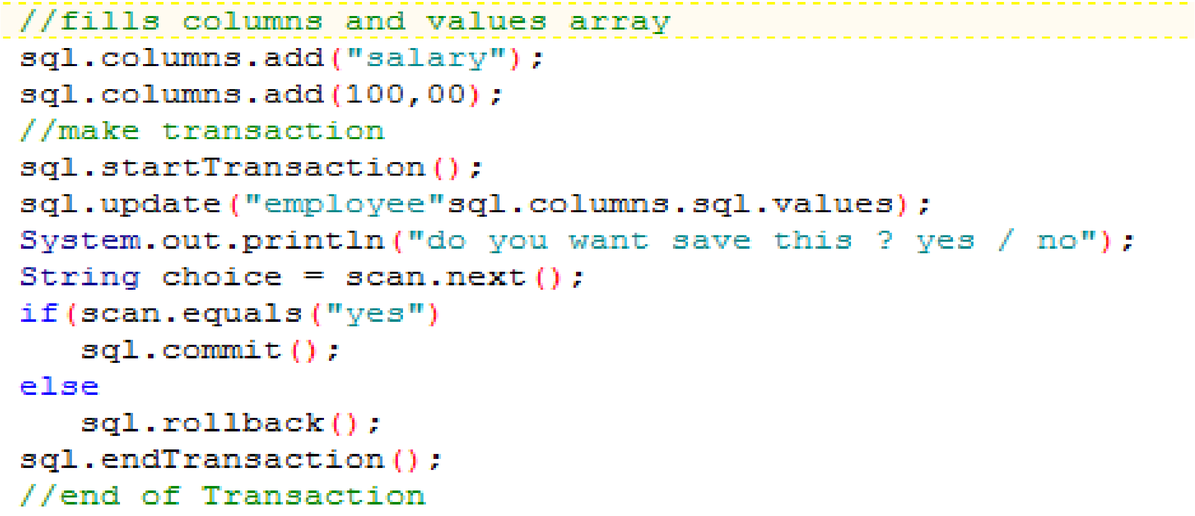
The transaction have a four methods

1- startTransaction() , use to tell the RDBMS to start a new transaction.
2- commit() , use to tell the RDBMS to commits the current transaction, making its changes permanent..

3- Rollback() , use to tell the RDBMS to rolls back the current transaction, canceling its changes.

4- EndTransaction() , use to tell the RDBMS to end the current transaction.

Use transaction methods

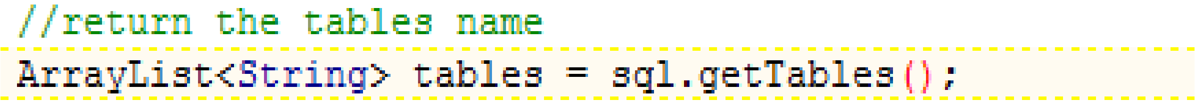


```
//fills columns and values array
sql.columns.add("salary");
sql.columns.add(100,00);
//make transaction
sql.startTransaction();
sql.update("employee"sql.columns.sql.values);
System.out.println("do you want save this ? yes / no");
String choice = scan.next();
if(scan.equals("yes")
   sql.commit();
else
   sql.rollback();
sql.endTransaction();
//end of Transaction
```

Figure : Illustrates transaction method

- **Meta data**

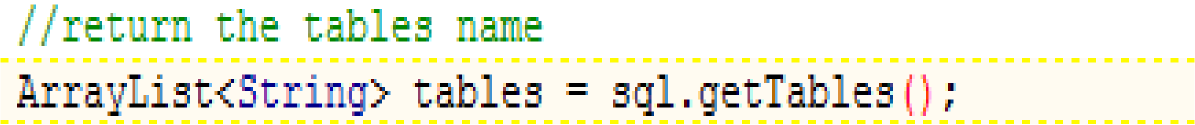
Metadata is literally data about data.
Meta data methods()
getTables() method
return an array list with all tables Name



```
//return the tables name
ArrayList<String> tables = sql.getTables();
```

Figure : Illustrates Meta datamethods().

getColumns(String tableName) methods
return an array list with all columns of specific table.



```
//return the tables name
ArrayList<String> tables = sql.getTables();
```

Figure IllustratesgetTables() method

5.4 TEST THE LIBRARY (PROGRAM DEMO)

We have developed a simple program by our library, the program can be deal to three types of data bases by the same methods and variables, just it need to change parameters of DBConnection constructer and connect method.

As example we well select name, id and email from table programmer and return just 4 rows.
By default the RDBMS is My SQL and all tables have the same data .
1-**we added the table name "user" to table name field and columns "name , email , id " to column field.**

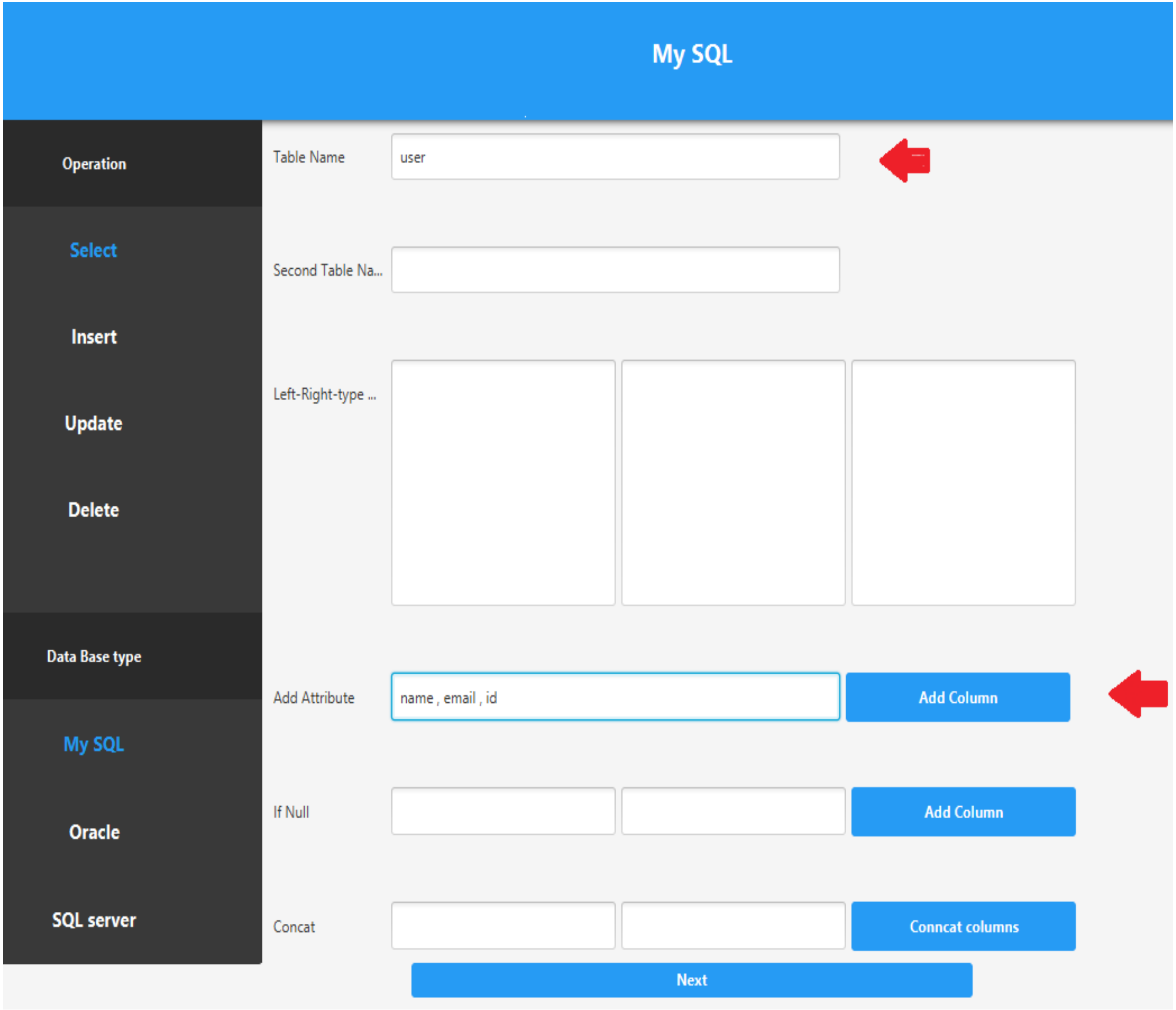
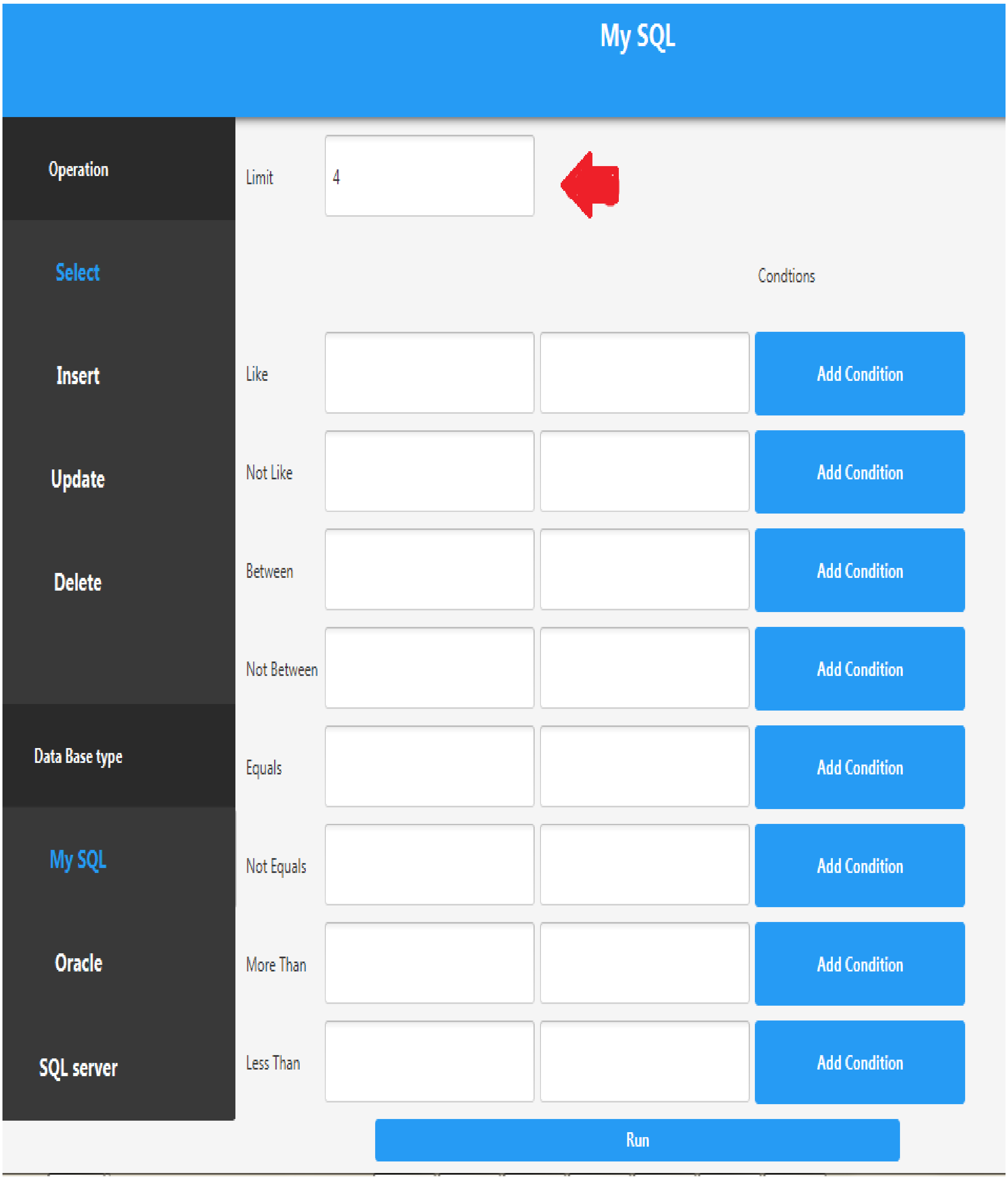


Table Name: user

Add Attribute: name , email , id

Figure : Illustrates added the table name

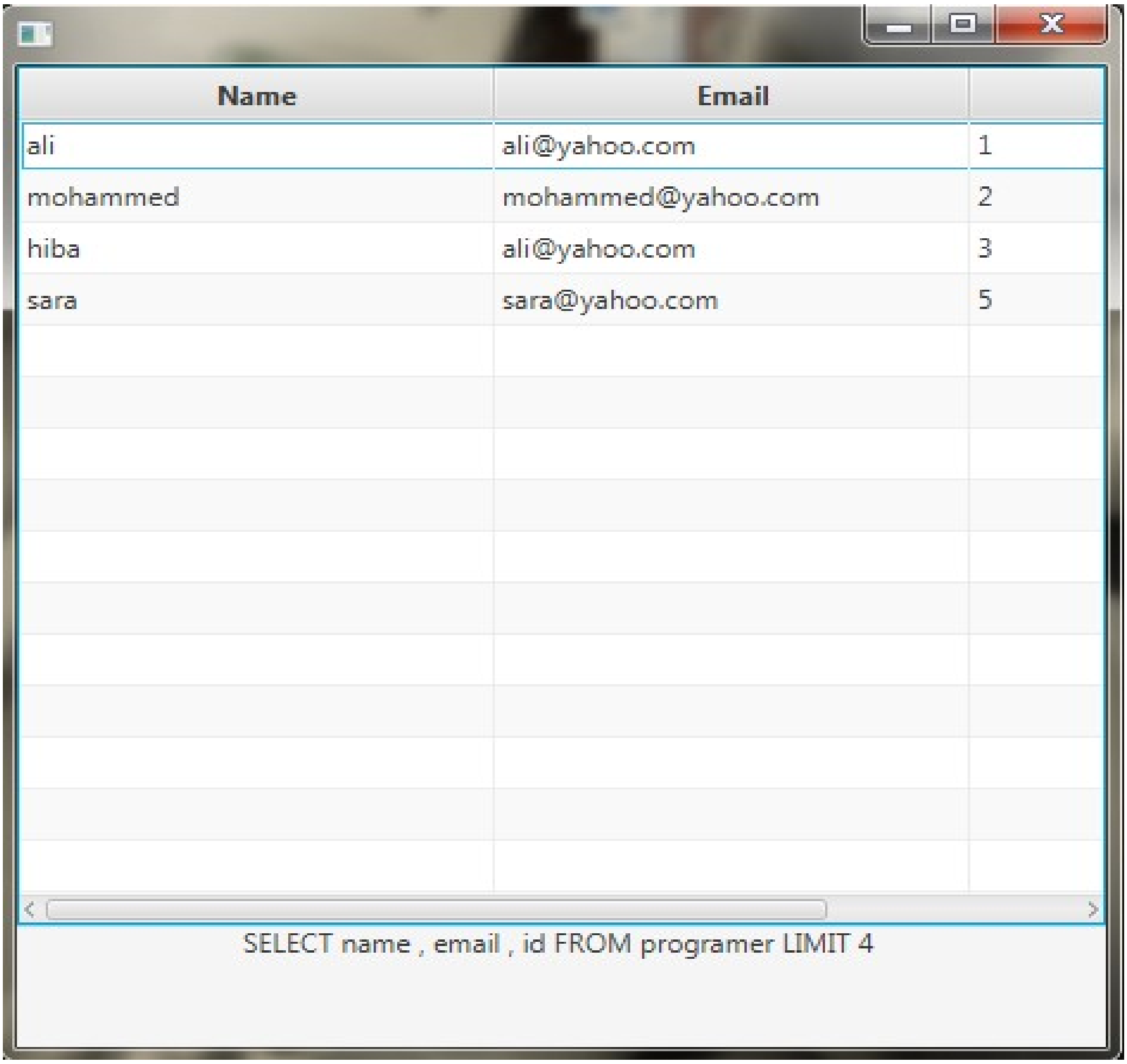
2-we added the limit of rows in limit field.



Limit 4

Figure : Illustrates **added the limit of rows**

3- **Run in MySql**

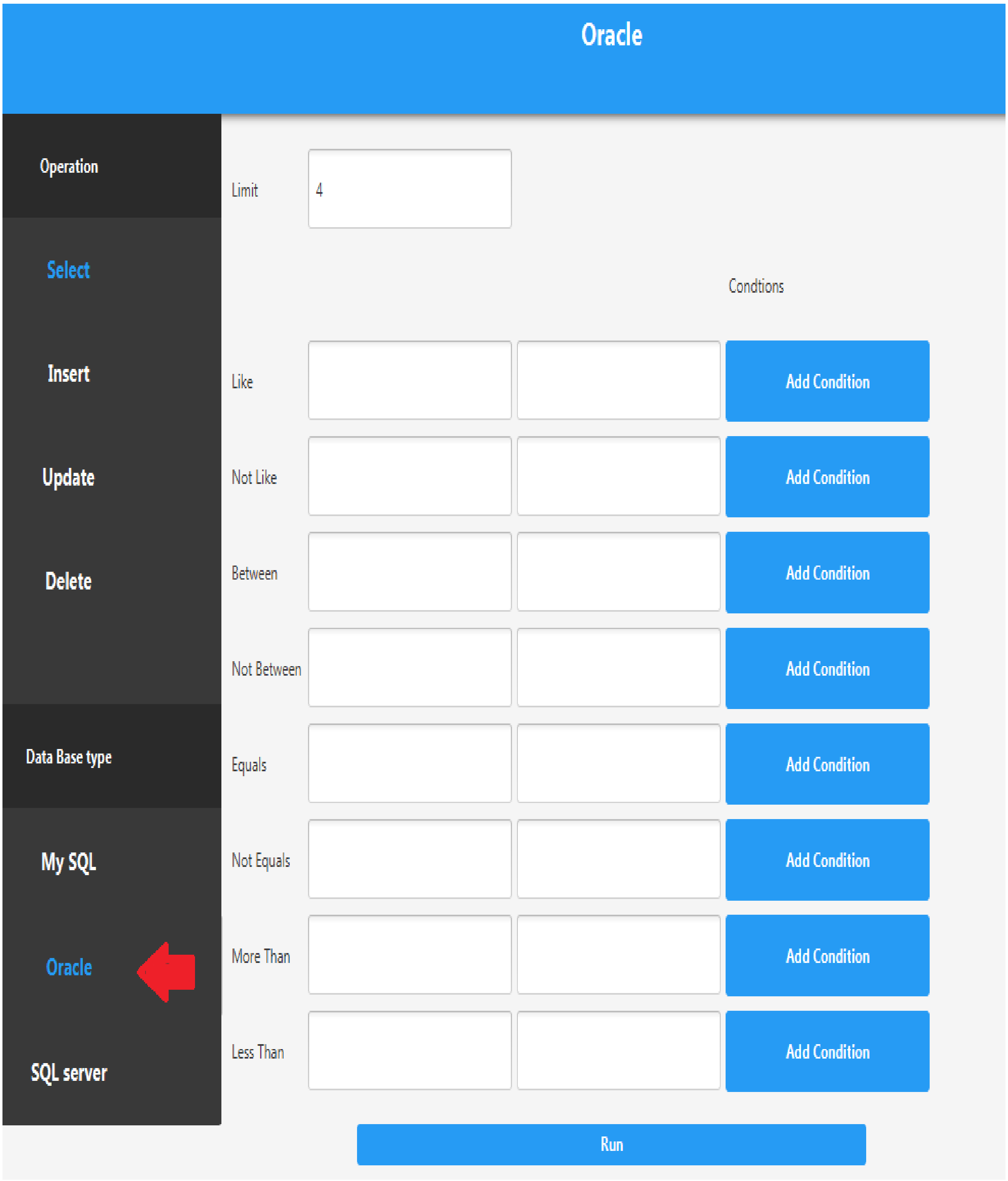


Name	Email	
ali	ali@yahoo.com	1
mohammed	mohammed@yahoo.com	2
hiba	ali@yahoo.com	3
sara	sara@yahoo.com	5

SELECT name , email , id FROM programer LIMIT 4

Figure : Illustrates **RunMySql**

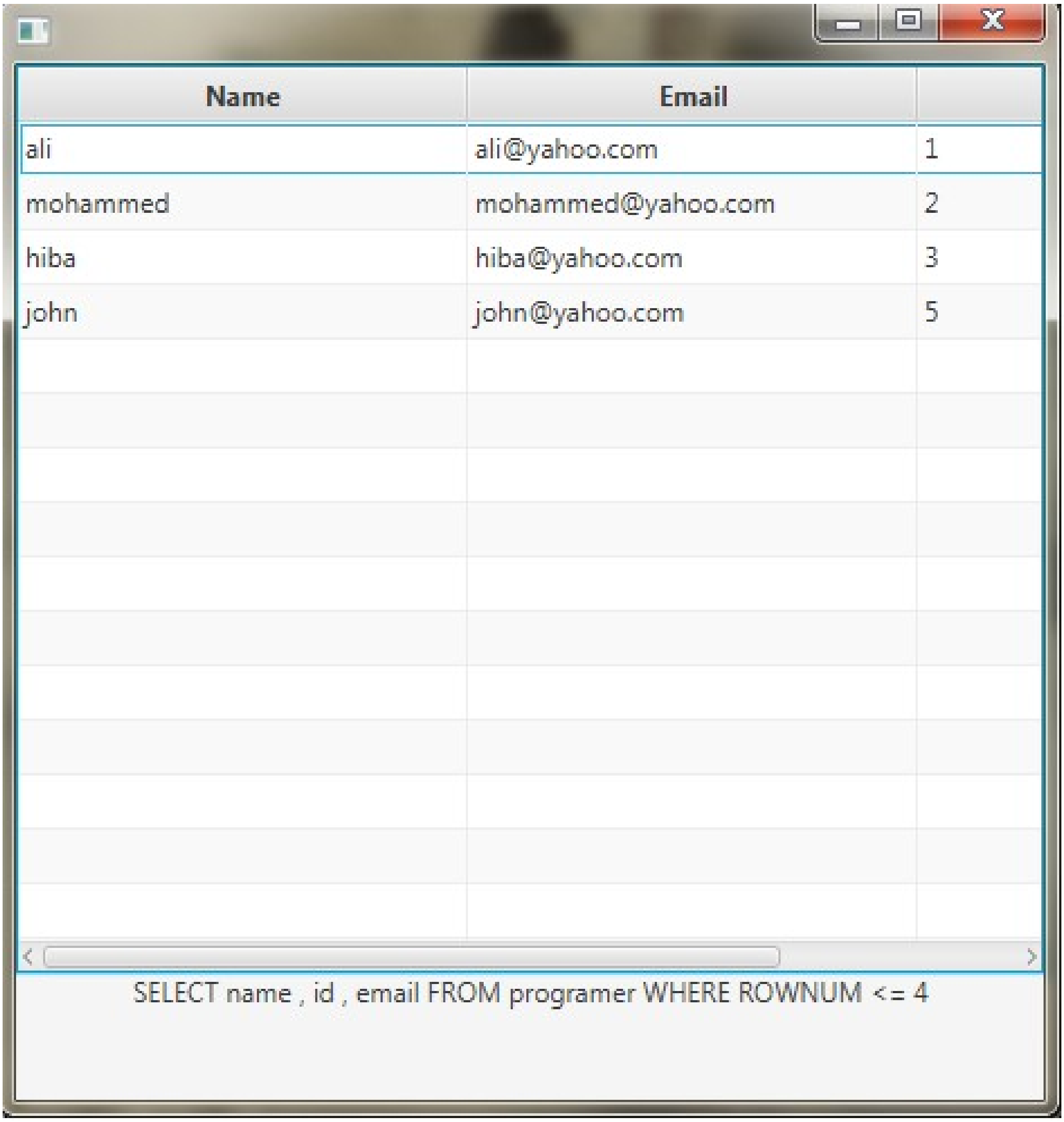
4-change the RDBMS to Oracle



Limit 4

Figure : Illustrates Oracle

5- **Run in Oracle**



Name	Email	
ali	ali@yahoo.com	1
mohammed	mohammed@yahoo.com	2
hiba	hiba@yahoo.com	3
john	john@yahoo.com	5

```
SELECT name , id , email FROM programer WHERE ROWNUM <= 4
```

Figure : Illustrates Run in Oracle

6-change the RDBMS to SQL Server

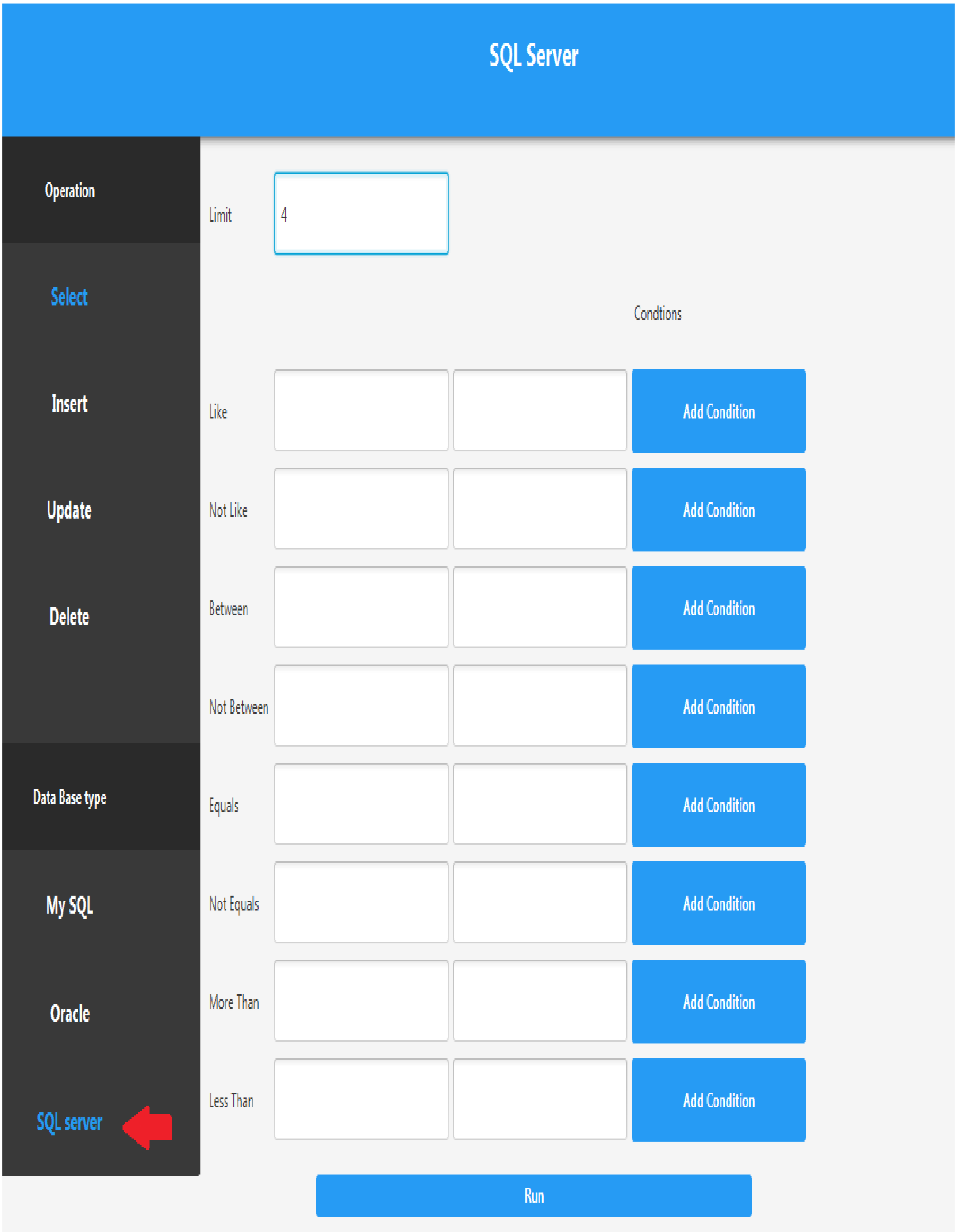
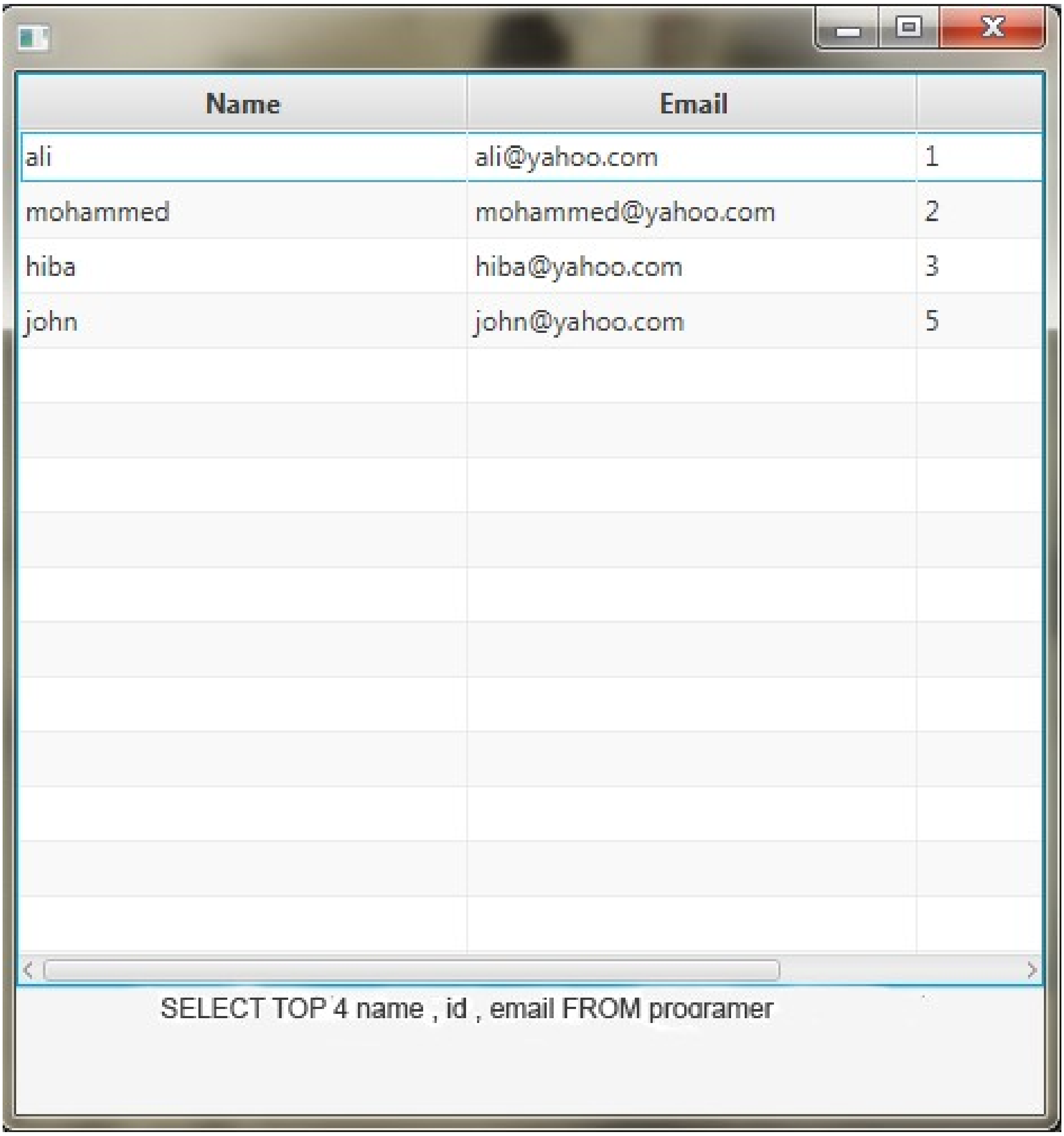


Figure : Illustrates in Sql Server

7-Run in Sql Server



Name	Email	
ali	ali@yahoo.com	1
mohammed	mohammed@yahoo.com	2
hiba	hiba@yahoo.com	3
john	john@yahoo.com	5

SELECT TOP 4 name , id , email FROM programer

Figure : Illustrates Run Sql Server.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

CHAPTER 6CONCLUSION AND RECOMMENDATIONS

6. 1 INTRODUCTION

This chapter shows the results that have been accessed from the build of the library, and also includes recommendations and conclusion.

6. 2 RESULTS

Through this research we reached the following result:

1- Has been created DBAL library that works as a layer between the program and the database.

2- Library supports three important types of databases.

3- Providing a lot of DML instructions.

4- Prevent SQL Injection.

5- No need to know to SQL syntax to building programs dealing with RDBMS.

6- Make the program run with three type of database.

6. 3 RECOMMENDATIONS

We recommend the following:

1- Add other types of databases.

2- Providing DDL and DCL instructions.

3- Improve the performance of algorithms.

4- Improve the validation.

5- Reduce the load on the database.

6. 4 CONCLUSION

This project is solving the problem of the difference in the SQL syntax of RDBMS, by building a Java library to develop software that can deal with three types of the most important databases, and achieved satisfactory results and has been confirmed to achieve all the goals in implementation.

BIBLIOGRAPHY

1. Oracle Help Center. *Oracle.* [Online] 2016. [Cited: June 12, 2016.] https://docs.oracle.com/cd/E11882_01/server.112/e40540/sqllangu.htm#CNCPT015.

2. **tutorialspoint.com.***SQL Tutorial.*

3. **Haes, Tom De.** SQL. *anubex.* [Online] april 1, 2016. [Cited: June 18, 2016.] http://www.anubex.com/blog/SQL-Standard-or-Structured-Query-Language.

4. IBM Knowledge Center. *IBM.* [Online] 12.1. [Cited: June 22, 2016.] http://www.ibm.com/support/knowledgecenter/SSGU8G_12.1.0/com.ibm.sqls.doc/ids_sqs_0212.htm. SSGU.

5. DATABASE TECHNOLOGY. *studytonight.* [Online] 2016. [Cited: June 22, 2016.] http://www.studytonight.com/dbms/sql-constraints.php.

6. **Bodnar, Jan.** MySQL tutorial. *zetcode.* [Online] 2007 . [Cited: June 23, 2016.] http://zetcode.com/databases/mysqltutorial/introduction/.

7. *MYSQL TUTORIAL.* **Singh, Mohit.** June 27, 2013, mohitmysql.blogspot.

8. MySQL Administration. *tutorialspoint.* [Online] 2016. [Cited: June 23, 2016.] https://www.tutorialspoint.com/mysql/mysql-administration.htm.

9. MySQL. *MySql.com.* [Online] Oracle, 2016. [Cited: June 23, 2016.] https://www.mysql.com/why-mysql/topreasons.html.

10. **Hill, Jordan.***Oracle SQL.* 2005. ISBN: 1-55558-323-7.

11. **Gertz, Michael.***Oracle/SQL Tutorial1.* Davis : s.n., 2000. pp. 20-27.

12. Introduction to the Oracle Database. *Oracle Help Center.* [Online] 2016. [Cited: June 29, 2016.] https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm.

13. SQL Server Data Tools. *microsoft.* [Online] microsoft, 2016. [Cited: June 30, 2016.] https://msdn.microsoft.com/en-us/library/hh272686(v=vs.103).aspx.
14. **John, Lim.** Database Abstraction Layer for PHP. *ADOdb.* [Online] 5.20.7, 2014. [Cited: July 6, 2016.] http://adodb.org/dokuwiki/doku.php.
15. **Anhofer, Johann.***Relational Database access layer.* 2014. N3886".
16. **Pierro, Massimo Di.** database abstraction layer. *web2py.* [Online] 3.0, 2016. [Cited: July 12, 2016.] http://www.web2py.com/books/default/chapter/29/06/the-database-abstraction-layer#The-database-abstraction-layer.
17. Java. *rajkumarreddysadi.wordpress.* [Online] [Cited: July 21, 2016.] https://rajkumarreddysadi.wordpress.com/welcome/java/.
18. Java Development. *theserverside.* [Online] 2016. [Cited: July 21, 2016.] http://www.theserverside.com/resources.
19. **Kumar, Krishan.** JDBC & MySQL Database Connection. *cs-fundamentals.* [Online] 2016. [Cited: July 23, 2016.] http://cs-fundamentals.com/java-programming/java-jdbc-connection-tutorial.php.
20. JDBC Architecture. *Oracle.* [Online] Oracle, 2015. [Cited: July 23, 2016.] https://docs.oracle.com/javase/tutorial/jdbc/overview/architecture.html.
21. NetBeans IDE. *netbeans.* [Online] Oracle, 2016. [Cited: Augst 3, 2016.] https://netbeans.org/community/releases/81/.
22. **Williams, Laurie.***Unified Modeling Language.* 2004.
23. SQL Tutorial. *w3schools.* [Online] [Cited: Septemper 18, 2016.] http://www.w3schools.com/sql/.