

“A QUALITATIVE APPROACH TO THE SOFTWARE DEVELOPMENT ENGINEERING AND TESTING”

*Project Report submitted in partial fulfillment of the
requirements for the Degree of*

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

By

Dwaipayan Vidyanta(171211)



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,
WAKNAGHAT

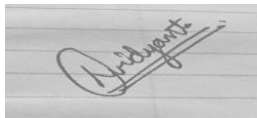
PROJECT UNDERTAKING BY CANDIDATE

I Mr. **Dwaipayan Vidyanta**, Roll No. **171211**, Branch: **CSE**, is doing my internship with **Cognizant** from **Feb, 2021** to **Aug, 2021**

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

A handwritten signature in black ink on a light-colored background. The signature is written in a cursive style and appears to read 'Dwaipayan Vidyanta'.

Name: Dwaipayan Vidyanta

Roll No. 171211

Date : 23/05/2021

TABLE OF CONTENTS

Topic	Page No
DECLARATION	
ACKNOWLEDGEMENT	
ABSTRACT	
CHAPTER-1: INTRODUCTION	1
1.1 General Introduction	1
1.2 Organization	3
CHAPTER-2: Introduction to Software testing process and Functional Testing	4
2.1 What is Testing	4
2.2 When to start testing?	4
2.3 When to stop testing?	5
2.4 Verification V validation	5
2.5 Stereotypes	6
2.6 QA vs QC vsTesting	8
2.7 Types of testing	9
2.8 Methods	11
2.9 Levels	13
2.10 Documentation	18

CHAPTER-3: Basics of JAVA Language **22**

3.1 Introduction	22
3.2 OOPS	23
3.3 Interfaces in JAVA	25
3.4 Exception Handling	26
3.5 Strings	28
3.5.1 String Methods	29
3.5.2 StringBuffer	30
3.6 Arrays in JAVA	32
3.7 Collections in JAVA	33
3.7.1 ArrayList	34
3.7.2 LinkedList	35
3.7.3 PriorityQueue	36
3.7.4 HashSet	37
3.7.5 TreeSet	38
3.8 Java Database Connectivity with MySQL	39

CHAPTER-4:SQL **42**

4.1 What is MySQL?	42
4.2 What is RDBMS?	43
4.3 Installation of MySQL	43
4.4 DDL	45
4.5 DML	47
4.6 About Joins	48
4.7 Subqueries	51
4.8 Normalization	52

CHAPTER-5: XPath and Selenium	55
5.1 What is XPath	55
5.2 Locating Elements by XPath	56
5.3 What is Selenium	59
5.4 Selenium Webdriver	60
5.5 Selenium basic commands	60

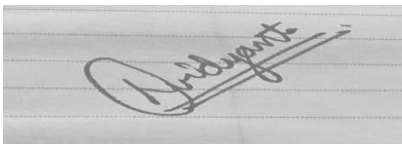
Conclusion and Future Work

REFERENCES

Candidate's Declaration

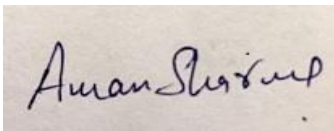
I hereby declare that the work presented in this report entitled "**Internship Report**" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from Feb 2021 to May 2021 under the supervision of Mr. **Abhijit Joshi** (Trainer at Cognizant).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

A handwritten signature in black ink on a light-colored background. The signature is written in a cursive style and appears to read 'Dwaipayan Vidyanta'.

Dwaipayan Vidyanta(171211)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

A handwritten signature in black ink on a light-colored background. The signature is written in a cursive style and appears to read 'Aman Sharma'.

Dr Aman Sharma

Supervisor

Acknowledgement

It gives me great pleasure to express my gratitude to Jaypee University and my college, Jaypee University of Information Technology, for providing me with the opportunity to test my abilities through this internship programme. For this chance, I would want to offer my heartfelt gratitude to our Tn P officer, Mr. Pankaj Kumar, and our faculty coordinator, Dr. Nafis UKhan. I'd also want to thank my internship supervisor for providing me with invaluable support and assistance in finishing this project.

I'd want to express my heartfelt gratitude to all of the officials and staff of cognizant, without whose help, my internship programme would not have been successful. Last but not least, I'd like to extend my heartfelt gratitude to everyone of my family, friends, and well-wishers for their unwavering support and well-wishes throughout the internship and the writing of this report.

This study, I feel, will be beneficial not just to academic institutions, but also to everyone interested in learning about internship experiences in auditing and consulting firms.

ABSTRACT

At Cognizant after I was onboarded, I was assigned to the SDET(Software Development Engineering with Testing) role. This report is prepared to cover the basic learning that I went through since joining th organization.

Withtheprogressioninitsfield,thereisacontinuousshiftfrommanualrobotizationtesting, we can see new positions coming up and one such job is of SDET. A Software Development/Design Engineer in Test (SDET) is an IT proficient who works for both, advancement and testing of the product. A SDET is likewise answerable for looking into the plan and the cycles associated with the productitem.

SDET ought to be equipped for performing test robotization, set up structures on application stages like web, portable, and work area. The SDET proficient is additionally answerable for exploring the issues looked by the client as guided by the specialized help group. SDET is neededtomakejustasoverseebugreports.Theyadditionallyneedtodealwiththespecialized correspondence with customers and work with organizationgroups.

CHAPTER 1

INTRODUCTION TO SDET

1.1 General Introduction

Who are SDETs and what do they do?

Testing has become more critical than programming as a result of our realization. For equipment and services, testing necessitates a larger budget. Traditional testers are falling short as Agile adoption grows and demands for faster consumer responsiveness grow. The traditional scenario of programmers writing code and testers evaluating code quality is evolving. Now is the time for testers to get involved. From the code creation stage Testers have progressed from becoming a liability to being an asset to every company. In reality, there is a significant increase in demand for people who can both write and test codes based on requirements. **Software Development Engineers in Test**, or SDETs, are people who work in this field.

SDETs are highly qualified individuals with experience in both Quality Engineering and software development. Though it's nice to have someone with in-demand skills and experience, it can also lead to some uncertainty about what SDETs' actual roles and responsibilities are. The distinction between an SDET and a QA Engineer is a frequently encountered ambiguity.

Difference between SDET and QA

Although the functions of SDETs and QA Automation Engineers can seem to be somewhat similar, with most of the tools and language skills needed (Selenium, Java, and Jenkins), there are some very noticeable and simple differences between the two.

In layman's terms, an SDET is a developer who serves as part of the test team rather than the product development team. In other words, SDETs are not only responsible for writing code, but also for testing it. SDETs are expected to write, test, and patch code on a regular basis. The Agile lifecycle model guides their tasks and responsibilities. SDETs are typically professionals with exceptional analytical, technical, and problem-solving abilities.

QA Engineers, on the other hand, are testers who do not need any programming expertise because they are rarely exposed to code. The functions and responsibilities of SDETs and QA Engineers are specifically defined as a result of this.

Need of SDET

With the acceptance of software testing as a necessity and value around the world, the questions of what to test, when to test, and how to test have never stopped evolving.

End-to-end Test Automation is required for the majority of modern products and apps, particularly in the areas of functional, performance, and security testing. SDETs, with their dual skill in code creation and testing, are ideal candidates. In this digital era, SDET are a perfect match. They aid in the improvement of code quality by doing thorough and comprehensive source code reviews as well as ensuring that the code is testable.

SDETs have become an important part of development ecosystems because they have advanced testing knowledge of various methods, strategies, best practises, and processes.

SDETs are expected to write code to test the code written by developers based on their development experience, knowledge of technical architecture and design, and programming skills. They'll also be expected to write unit tests and conduct white-box testing.

The following is a list of some of the activities that SDETs are usually in charge of:

- Building test automation systems that are reliable, scalable, and high-quality for practical, regression, and performancetesting.
- Developing technology for high-quality automation and ensuring that the code is thoroughlytested.
- Environment/test automation frameworks development, customization, deployment, andmanagement
- Product scalability, reliability, accuracy, and efficiency are allexamined.
- Participating in architecture and constructiondebates.
- Debugging of the highestcalibre.
- Reporting on testresults.

In a Nutshell.

SDETs are consumer advocates who use their knowledge of end-user preferences to influence product design. Although functional and automation testers will always be required, SDETs may be the all-arounder that most companies need. They're expected to dig deep into the code to see if it's in line with the overall goals and how it integrates with other parts of the application.

1.2 Organization

Cognizant Technology Solutions, based in New Jersey, is a global IT administration company (US).

In 1994, it started as a consultant to Dun and Bradstreet organisations all over the world. Dun and Bradstreet Satyam Software was the company's name back then, and it was based in India (DBSS).

As D&B split into three companies in 1998, it became self-contained, with one group of companies becoming Cognizant Corporation. By that time, DBSS had acquired Satyam Software's stake and was completely secure.

With a base camp in New Jersey, the company began providing IT Services and Solutions to clients worldwide as CTS or the brand Cognizant all inclusive. Mindful is listed on the NASDAQ stock exchange in the United States under the symbol CTSH.

Since then, the company has progressed significantly in the industry!

Cognizant Technology Solutions Corporation (Cognizant) is a provider of custom data innovation (IT) conference and innovation administrations, as well as redistributing services to Global 2000 companies in North America, Europe, and Asia. The Company's core administrations include innovation methodology consulting, complex framework creation, venture programming package execution and upkeep, data warehousing and business expertise, application testing, application support, foundation the board, and vertically-situated business process re-appropriating (V-BPO). Innovation methodology consultancy, complex system development, venture programming package execution and maintenance, data warehousing and market expertise, application testing, application support, foundation the board, and vertically-positioned business process re-appropriating are among the Company's core administrations (V-BPO).

CHAPTER 2

Introduction to Software Testing Process and Functional Testing

2.1 What is Testing?

Testing is the way toward surveying a framework or its component(s) fully intent on deciding if it meets the characterized necessities. Basically, testing is the way toward running a framework to discover any openings, bugs, or fragmented particulars that are not met by the real necessities.

Testing, as indicated by the ANSI/IEEE 1059 norm, is "a strategy for assessing a product thing to distinguish irregularities among current and required conditions (that is, abandons/blunders/bugs) and to test the product thing's highlights."

2.2 When do we start Testing?

One ambitious starting of testing decreases the time and cost to reiterate and design programs that are free of bugs and blunder, that may be delivered to the customer. According to the Software development Life Cycle (SDLC), We can start testing right from the gathering of requirements and continue till the release of the product.

Basically it depends upon the development model applied. For example, in the Waterfall model, there is a formal stage for testing and it is typically done only once; on the other hand, in the spiral model at the end of each development cycle the product is checked for possible bugs and the entire product is finally checked at the end.

- While gathering requirements, testing includes confirming and finding out additional prerequisites.
- Testing may also include reviewing of the development plan in the planning stage.
- Developers may also perform testing.

2.3 When do we stop the testing process?

When it comes to testing, it may be hard to decide when to stop your hand, Testing is an endless process. Everyone falls short to claim that they have tried every nook and cranny of the product. The following factors may help us to decide—

- Running out of deadlines
- Experiment execution completion
- Limited completion of code and practical part
- Low bug rate and higher %age of low priority bugs.
- Wish of the upper management

2.4 Verification Versus Validation

These two terms are exceptionally befuddling for the vast majority, who use them reciprocally. The accompanying table features the contrasts among confirmation and approval.

Sr.No.	Verification	Validation
1	"Are you building it right?" is a question that verification tackles.	"Are you constructing the right thing?" is a question that validation tackles.
2	Ensures that the software system has all of the necessary features.	Ensures that the functionalities perform as expected.
3	The first step is verification, which involves searching for paperwork, code, and so on.	Validation comes after verification and entails a thorough examination of the whole product.
4	Developers are in charge of this.	Testers are in charge of this.

5	It has static operations, such as gathering feedback, walkthroughs, and inspections in order to validate software.	It involves complex operations, such as executing applications in accordance with requirements.
6	It is an objective operation, and there should be no need for a subjective decision to validate software.	It's a subjective process that includes making subjective judgments on how well a piece of software performs.

2.5 Software Testing QA vs QC vs Testing

Testing, Quality Assurance, and Quality Control

A great many people get confounded with regard to nail down the distinctions among Quality Assurance, Quality Control, and Testing. In spite of the fact that they are interrelated and somewhat, they can be considered as same exercises, yet there exist recognizing focuses that put them aside. The accompanying table records the focuses that separate QA, QC, and Testing.

Quality Assurance	Quality Control	Testing
QA tasks include ensuring the implementation of processes, protocols, and standards in the sense of verifying developed software and meeting expected requirements.	It entails activities that ensure that established software meets known (or undocumented, in some cases) requirements.	It entails practises that ensure the detection of software glitches, errors, and defects.

Rather than performing actual system testing, this approach focuses on processes and procedures.	Focuses on actual testing by conducting applications with the aim of identifying bugs/defects using procedures and processes.	The focus is on actual research.
Activities focused on the method.	Activities that are focused on the product.	Activities that focus on the product.
Preventive measures are taken.	It is a remedial procedure.	It's a preventative measure.
It's a part of the Software Testing Life Cycle (STLC).	Quality Control is a branch of Quality Assurance.	Quality Control includes testing as a subset.

2.7 Testing Types

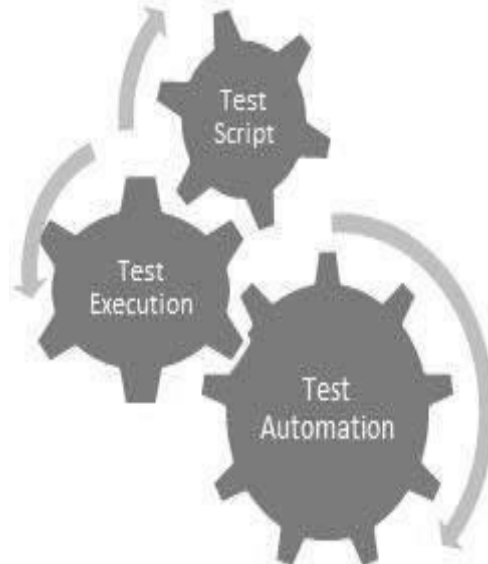
Manual Testing

Manual testing is when the tester tests the application as the end user and checks for any undesirable behavior in the product. All parts of the application are thoroughly explored by the tester as an end user would to check if everything is working as it should be. Whenever an abnormal behavior is found it is flagged as a defect. It may be critical or just a UI defect.

Automatic Testing

Automated testing Or Testing by automation is when the tester writes programs that are explicitly written to test other programs/ application. These programs mimic the way a real user may give input into the application and logs all the output that is yielded upon giving that input.

Apart from making it easier upon the tester automation testing is helpful for load testing, smoke testing and stress testing. They are also helpful for regression testing as a lot of test



2.1 Software Testing Process

cases are repeated for these methods.

What do we test using automation testing?

Mainly the areas in an application where a lot of users concurrently provide inputs and request output is where automated testing is used to check things out.

Moreover defects in the GUI, field validation etc. could also be checked by automated testing.

When do we use Automation testing?

Test automation should be used in:

- Large Scale applications
- Similar areas requiring repeated testing
- Stable requirements
- Where there is concurrent access by clients
- Where there is abundance of time

Programming Testing Tools

The accompanying devices can be utilized for computerization testing – HP Quick Test Professional, Selenium, IBM Rational Functional Tester etc.

2.8 Software Testing –Methods

There are various techniques that can be utilized for programming testing. This part momentarily depicts the techniques accessible.

Black Box Testing

Black Box Testing is also known as behavior based testing. These techniques test the system without any prior knowledge of the internal structure of the application. Meaning what goes on inside the application is a black box to the tester. The tester can only give input and wait for the output.

In this technique tester mainly focuses whether application is working around boundaries of possible inputs.

Some black box techniques are:

- Equivalence partitioning(E.P.)
- 2value or 3value BVA (Boundary Value Analysis)
- State Transition

White Box Testing

White box testing is also known as structural or architectural testing. Here tester looks at the internal structure of the system and hence they must be familiar with the language the application is coded in.

These techniques are mainly used to check for any logical error in the many branches and loops within the code.

Some useful techniques are:

- StatementCoverage(S.C.)
- DecisionCoverage(D.C)
- PathCoverage(P.C)

A Comparison of Testing Methods

The accompanying table records the focuses that separate discovery testing, dim box testing, and white box testing.

Black Box Testing	White Box Testing
It is not necessary to understand an application's internal workings.	The tester is fully aware of the application's internal workings.
Closed box testing, data-driven testing, and functional testing are other terms for the same thing.	Clear box testing, structural testing, and code-based testing are other terms for the same thing.
End users, as well as testers and developers, participate.	Normally, testers and developers are in charge of this.
External standards are used to guide testing. The application's internal behaviour is unknown.	Internal workings are well understood, allowing the tester to create test data that reflects this.
It is the most thorough and takes the least amount of time.	The most time-consuming and thorough method of research.
Algorithm testing is not suitable with this.	It is suitable for algorithm testing.

This can only be accomplished by trial and error.

Internal data domains and boundaries can be better checked.

2.9 Levels of Software Testing

The 2 main stages of software testing are:

- Functional
- Non-Functional

Functional Testing

This is a sort of discovery testing that depends on the determinations of the product that will be tried. The application is tried by giving info and afterward the outcomes are analyzed that need to adjust to the usefulness it was planned for. Utilitarian testing of a product is led on a total, coordinated framework to assess the framework's consistence with its predetermined necessities.

A compelling testing practice will see the above advances applied to the testing arrangements of each association and consequently it will ensure that the association keeps up the strictest of guidelines with regard to programming quality.

Unit Testing

This kind of testing is performed by engineers before the course of action is offered over to the testing gathering to authoritatively execute the examinations. Unit testing is performed by the specific architects on the individual units of source code consigned zones. The designers use test data that is extraordinary according to the test data of the quality certification bunch.

The target of unit testing is to isolate each piece of the program and show that individual parts are directly to the extent requirements and helpfulness.

Impediments of Unit Testing

Where Unit testing is immensely helpful to determine if the software component is working

correctly. It cannot be completely relied upon as it does not show that it would co-ordinate with the system as a whole.

Integration Testing

Integration in testing is characterized as the testing of consolidated pieces of an application to decide whether they work effectively. Incorporation testing should be possible twofold: Bottom up coordination testing and Top down reconciliation testing.

Integration Testing Method
Top down integration During this testing, the most elevated level modules are tried first, trailed by lower level modules.
Bottom up integration Unit testing is the main stage, trailed by trial of progressively complex blends of units known as modules or fabricates.

In an extensive programming advancement climate, base up testing is normally done first, trailed by top down testing. The cycle closes with numerous trial of the total application, ideally in situations intended to emulate genuine circumstances.

System Testing

System testing tests the system all things considered. At the point when all of the parts are fused, the application with everything taken into account is attempted altogether to see that it fulfills the predefined Quality Guidelines. This sort of testing is performed by a specific testing bunch.

- System testing is the initial phase in the Software Development Life Cycle, where the application is tried in general.
- The application is tried altogether to check that it meets the useful and specialized details.
- The application is tried in a climate that is near the creation climate where the application will be conveyed.

- System testing enables us to test, check, and support both the business requirements similarly as the application designing.

Regression Testing

Whenever a change of an item application is made, it is entirely possible that various districts inside the application have been affected by this change. Regression testing is performed to watch that a fixed bug hasn't achieved another handiness or business rule encroachment. The point of regression testing is to ensure that a change, for instance, a bug fix should not achieve another issue being revealed in the application.

Regression testing is huge because of the going with reasons –

- Minimize the openings in testing when an application with changes made should be attempted.
- Testing the new changes to affirm that the movements made didn't impact some other space of the application.
- Mitigates risks when regression testing is performed on the application.
- Test incorporation is extended without compromising schedules.
- Increase speed to grandstand the thing.

Acceptance Testing

This is apparently the principle kind of testing, as it is coordinated by the Quality Assurance Team who will check whether the application meets the proposed points of interest and satisfies the client's essential. The QA gathering will have a lot of pre formed circumstances and examinations that will be used to test the application.

More considerations will be shared about the application and more tests can be performed on it to check its accuracy and the reasons why the endeavor was begun. Acceptance tests are not simply proposed to raise fundamental spelling messes up, remedial botches, or interface openings, yet in expansion to raise any bugs in the application that will achieve system crashes or critical errors in the application.

By performing acceptance tests on an application, the testing gathering will diminish how the application will act in progress. There are moreover legal and legitimately restricting essentials for acceptance of the structure.

Non Functional Testing

The current segment centers around assessing an application dependent on its non-functional qualities. Nonfunctional testing involves assessing programming dependent on nonfunctional yet basic measures like proficiency, insurance, and UI.

Coming up next are the absolute generally critical and broadly utilized nonfunctional testing structures.

Execution Testing

Maybe than distinguishing glitches in programming, it is frequently used to identify bottlenecks or execution issues. There are an assortment of variables that lead to a product's yield being decreased.

As far as the accompanying elements, execution testing is viewed as perhaps the most basic and mandatory testing types:

- Effortlessness (for example Reaction Time, information delivering and getting to)
- Amount of capacity
- Continuity
- Adaptability

Execution evaluation can be subjective or quantitative, and it's separated into subtypes like burden testing and stress testing.

Burden Testing

It is a cooperation of testing the direct of an item by applying most limit trouble the extent th at programming getting to and controlling gigantic data. It might be done at both run of the m ill and apex load conditions. This kind of testing perceives the best furthest reaches of progra mming and its lead at active time.

Usually, load testing is performed with the help of electronic gadgets like Load Runner, App Loader, Apache JMeter, Silk Performer, Visual Studio Load Test, etc

Virtual customers (VUsers) are portrayed in the electronic testing contraption and the substance is executed to check the pile testing for the item. The amount of customers can be extended or decreased all the while or progressively subject to the necessities.

Stress Testing

Stress testing joins testing the direct of an item under odd conditions. For example, it may consolidate eliminating a couple of resources or applying a store past the genuine weight li mit.

The place of pressing factor testing is to test the item by applying the pile to the system an d expecting authority over the resources used by the item to recognize the breaking point.

This testing can be performed by testing different circumstances, for instance, –

- Closure or restart of association ports discretionarily
- Killing the informational index on or
- Running different cycles that consume resources like CPU, memory, laborer, etc

Usability Testing

Usability testing is a behavioral testing methodology and is used to separate any error(s) and upgrades in the item by seeing the customer through their

utilization and action.

According to Nielsen, comfort can be described similar to five factors, for instance viability of usage, learn limit, memory limit, botches/security, and satisfaction. As demonstrated by him, the comfort of a thing will be worthy and the structure is usable if it has the above factors.

Nigel Bevan and Macleod contemplated that accommodation is the quality need that can be assessed as the aftereffect of relationship with a PC structure. This need can be fulfilled and the end customer will be satisfied if the normal goals are refined effectively with the use of suitable resources

UI versus Usability Testing

The Graphical User Interface (GUI) of the product is tried during UI testing. UI testing guarantees that the GUI meets the particulars and that tone, direction, scale, and different properties are completely checked.

Usability testing, then again, guarantees a decent, easy to use interface that is easy to utilize. Usability testing incorporates UI testing, which is a subset of usability testing.

2.10 Software Testing – Documentation

Testing documentation incorporates the documentation of relics that should be made beforehand or during the testing of Software.

Documentation for programming testing helps in surveying the testing effort required, test, consideration, essential after/following, etc This part portrays a bit of the by and large used recorded relics related to programming testing, for instance,

- Test Planning
- Test Suite

- Testcase
- RTM

Test Planning

A test plan follows the method that will be used to test an application, the resources that will be used, the test environment where testing will be performed, and the hindrances of the testing and the schedule of testing works out. Customarily the Quality Assurance Team Lead will be liable for forming a Test Plan.

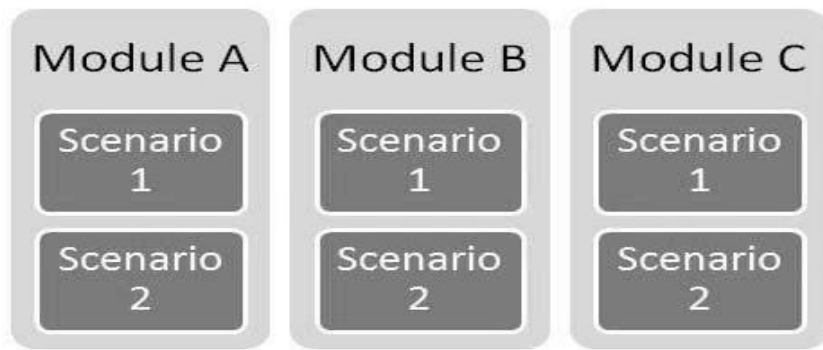
A test plan fuses the going with –

- Prologue to the Test Plan report
- Suspicions while testing the application
- Rundown of trials associated with testing the application
- Rundown of features to be attempted
- What sort of approach to manage use while testing the item
- Rundown of assumptions that ought to be attempted
- The resources appropriated for testing the application
- Any perils inferred during the testing communication
- A schedule of endeavors and accomplishments to be cultivated

Test Suites

It is a one line clarification that determines which area in the application will be attempted. Test suites are used to ensure that all cycle streams are attempted starting with one completion then onto the next. A particular space of an application can have only one test circumstance to a few hundred circumstances depending upon the degree and unpredictability of the application.

The terms 'test circumstance' and 'trials' are used then again, in any case a test circumstance as a couple of stages, while an analysis has a singular development. Seen from this perspective, test circumstances are tests, yet they fuse a couple of investigations and the progression that they should be executed. Beside this, each test is dependent upon the yield from the past test.



Example of Test Module having multiple test scenarios/suite

Testcase

Testcases incorporate a lot of steps, conditions, and information sources that can be used while performing testing tasks. The major assumption for this development is to ensure whether an item leaves or comes behind short similar to its value and various perspectives.

There are various kinds of examinations like commonsense, negative, botch, sound investigations, real trials, UI tests, etc

Additionally, tests are formed to screen the testing incorporation of an item. Overall, there are no regular designs that can be used during test making.

In any case, the going with sections are reliably available and associated with each trial

- Test caseID
- Assumptions
- Preconditions
- Intent
- Steps
- Expectedresult
- Actualresult

A solitary test situation can yield countless experiments. Moreover, various experiments are now and again composed for a solitary piece of programming, and these are alluded to as test suites.

Requirement Traceability Matrix

Perceptibility Matrix (OR Requirement Traceability Matrix RTM) is a table used to follow the essentials during the Software Development Life Cycle. It will in general be used for forward after (for instance from Requirements to Design or Coding) or in invert (for instance from Coding to Requirements).

There are various customer described configurations for RTM.

Each essential in the RTM report is associated with its connected test so that testing ought to be conceivable as indicated by the referred to necessities. Plus, Bug ID is also included and associated with its connected requirements and test. The essential destinations for this structure are –

- Ensure the item is made by the referred to necessities.
- Helps in finding the fundamental driver of any bug.
- Helps in after the made records during different times of SDLC.

CHAPTER 3

Tools of Automated Testing: JAVA

3.1 Introduction

Java is a broadly useful programming language that is class-based, object-situated, and intended to have as not many execution conditions as could reasonably be expected. It is expected to let application engineers compose once, run anyplace (WORA), implying that ordered Java code can run on all stages that help Java without the requirement for recompilation. Starting at 2019, Java was one of the most well known programming dialects being used by GitHub, especially for customer server web applications.

Real world Applications

There are numerous spots where Java is utilized in reality, beginning from a business online business site to android applications, from logical application to money related applications like electronic exchanging frameworks, from games like Minecraft to work area applications like Eclipse, Netbeans, and IntelliJ, from an open-source library to J2ME applications, and so forth.

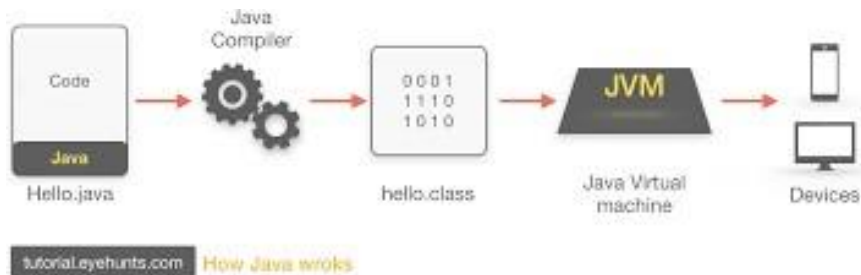


Figure 3.1 How it works

3.1.1 Java virtualMachine

Java Virtual Machine (JVM) is a motor that gives runtime condition to drive the Java Code or applications. It changes over Java byte code into machines language. JVM is a piece of Java Run Environment (JRE). In other programming dialects, the compiler produces machine code for a specific framework. In any case, Java compiler produces code for a Virtual Machine

3.1.2 Data types injava:

Primitive data types: Primary data forms include integer, character, short, large, float and double.

Non-primitive data types: They comprise classes, interfaces, and arrays.

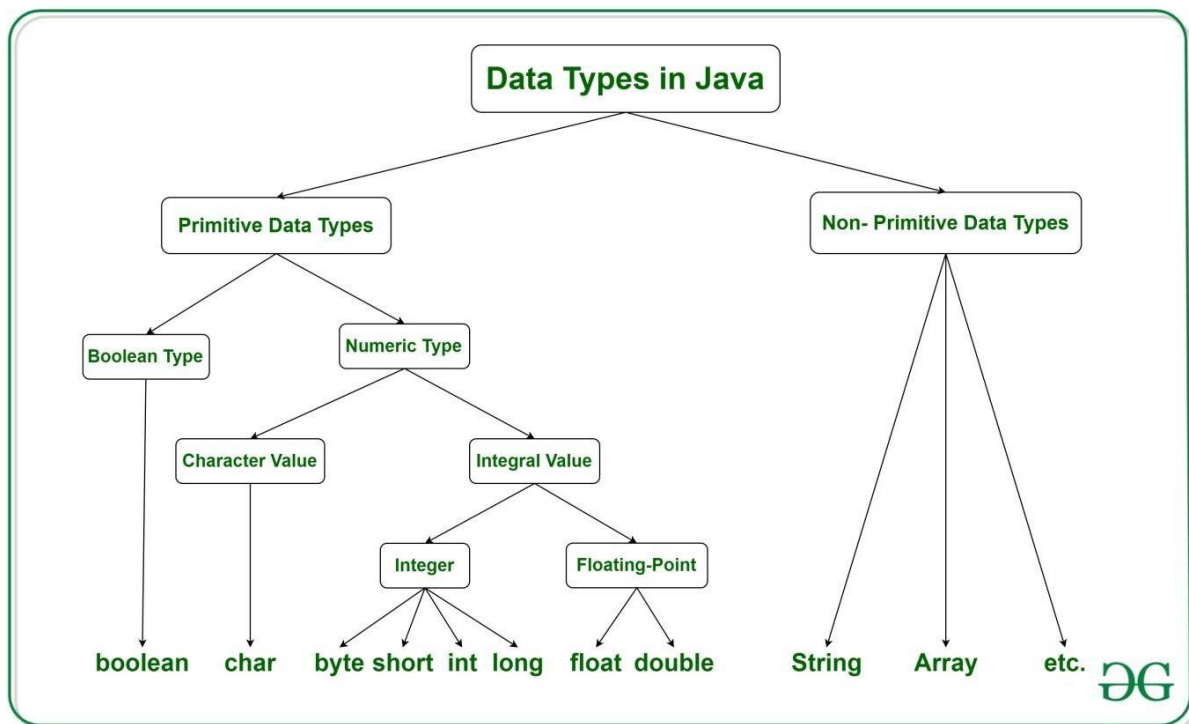


Figure 3.2 Data Types in Java

3.2 What isOOPS?

Object-oriented programming System (OOPS) is a programming worldview dependent on the idea of "objects" that contain information and strategies. The basic role of it is arranged writing computer programs to build the adaptability and practicality of projects. Articles situated

programming unites information and its behaviour (methods) in a solitary location (object) makes it more clear how a program functions.

Concepts of OOPS:

- **Class:** It is a conclave of comparative elements. For instance, on the off chance that you had a class called "Costly Cars". Its properties (data) can be cost or speed of these vehicles. While the strategies might be performed with these vehicles are driving, slowing down and soforth.
- **Objects:** An object can be designated as a development of a class, and there can be various versions of a class in a program. For instance- seat, bicycle, marker, pen, table, vehicle, and soforth.
- **Inheritance:** Classes can share, acquire or "acquire" properties and techniques that have a place with existing classes. This lets you reuse existing code and lessens the time you spend coding. A class that acquires from another is known as an inferred class or a youngster class. A class that shares its properties and strategies is known as a base class or parent class.

The different types of inheritance in java are:

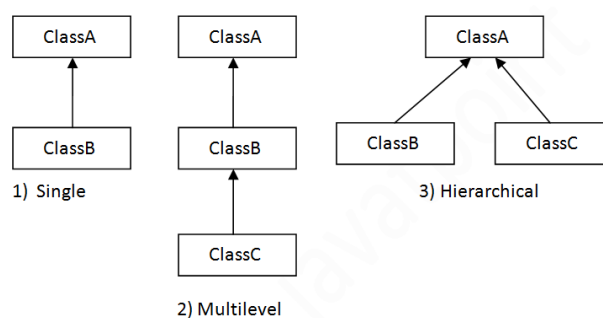


Figure 3.3 Types of Interference in Java

Note: Java does not support multiple inheritance.

- **Polymorphism:** alludes to the capacity of a strategy to be utilized in various manners, that is, it can take various structures at various occasions (poly + morphs). There are two sorts of polymorphism: compile time polymorphism and run time polymorphism.

Compile time (static) polymorphism happens when a technique is over-burden; that is, the point at which the contention utilized with the strategy is changed. This is done to get various outcomes. A case of this would propose various names for being the President of a nation, which would get you various outcomes each time – yet they would in any case be known as the President.

Run time (dynamic) polymorphism happens when the techniques itself are changed. At the point when you needn't bother with a President yet a Prime Minister, you would need to supersede the current strategy.

- **Abstraction:** Information deliberation alludes to the procedure of just showing applicable properties and techniques to deal with an item, while concealing the rest. Information deliberation lets you lessen the multifaceted nature (clearly) of a program and is an enormous bit of leeway offered by classes in OOP dialects.
- **Encapsulation:** Embodiment alludes to keeping objects with their strategies in a single spot. It likewise ensures the trustworthiness of the information – keeps it from being unnecessarily modified by confining access to the information, ideally by concealing it from outside components.

3.3 Interfaces in Java

Like a class, an interface can have techniques and factors, however the strategies pronounced in an interface are naturally unique (just strategy signature, nobody).

- If a class executes an interface and doesn't give strategy bodies to all capacities indicated in the interface, at that point the class must be announced theoretical.

- A Java library model is, Comparator Interface. On the off chance that a class executes this interface, at that point it tends to be utilized to sort an assortment.

Syntax:

```
interface <interface_name> {
/proclaim steady fields
/proclaim techniques that theoretical
/of course. }
```

3.4 Exception Handling

The Exception Handling in Java is one of the incredible instrument to deal with the runtime blunders so typical progression of the application can be kept up.

JAVA exception Hierarchy

The java.lang.Throwable class is the root class of Java Exception order which is acquired by two subclasses: Exception and Error. An order of Java Exception classes is given beneath:

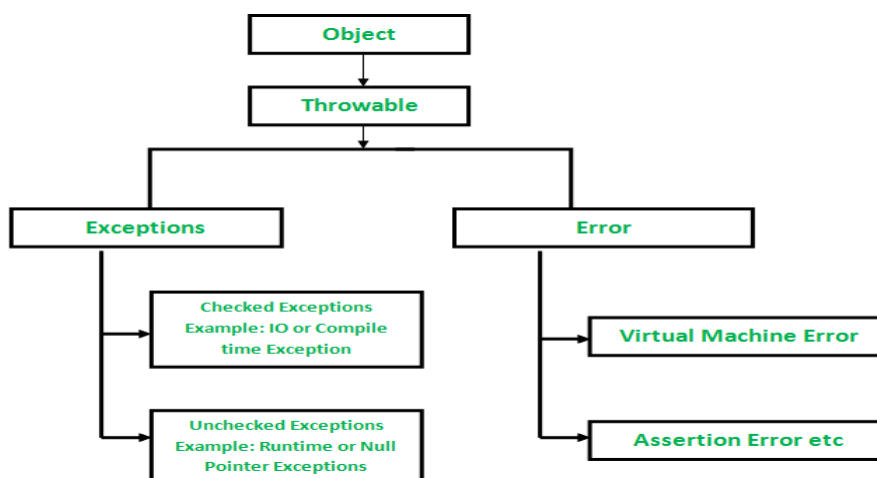


Figure 3.4 Exception Hierarchy

Types of Exceptions:

- Built-inExceptions
- User DefinedExceptions

3.4.1 Java ExceptionKeywords:

There are 5 watchwords which are utilized in dealing with special cases in Java.

1.	Try	The "try" watchword is utilized to determine a square where we should put exemption code. The attempt square should be trailed by either get or at last. That is to say, we can't utilize try square alone.
2.	catch	The "catch" square is utilized to deal with the exemption. It must be gone before by attempt square which implies we can't utilize catch square alone. It tends to be trailed by at long last square later.
3.	finally	The "finally" square is utilized to execute the significant code of the program. It is executed whether a special case is dealt with or not.
4.	throw	The "throw" catchphrase is utilized to toss a special case.
5.	throws	The "throws" catchphrase is utilized to announce exemptions. It doesn't toss a special case. It determines that there may happen a specialcaseinthetechnique.Itisconstantlyutilizedwithstrategy signature.

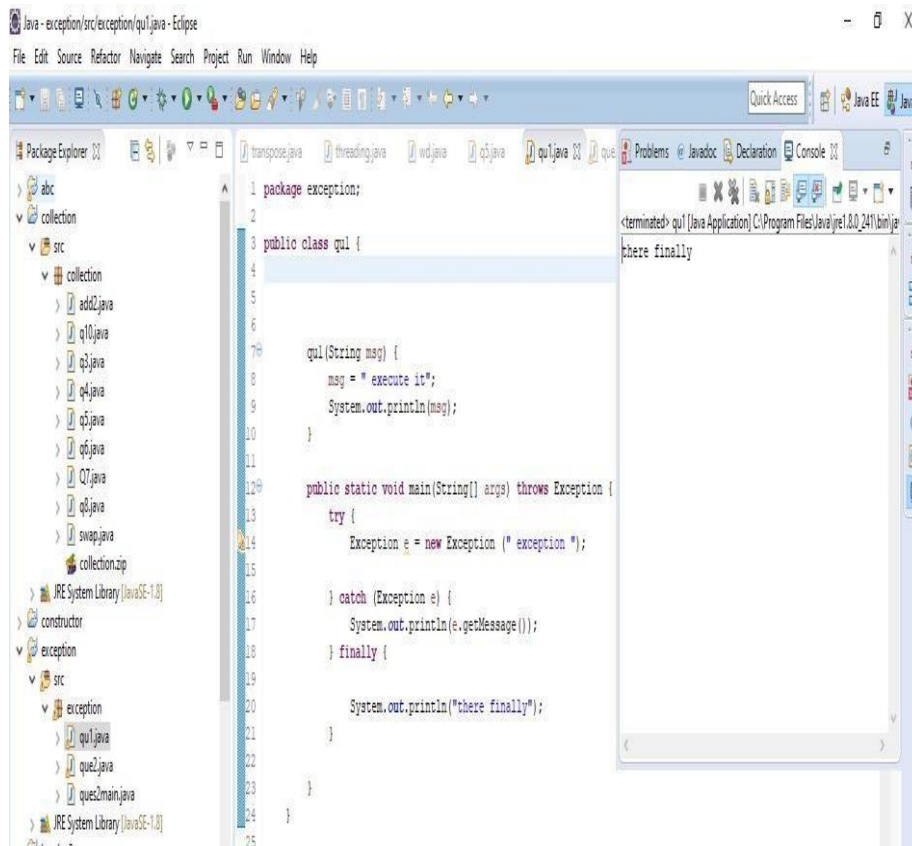


Figure 3.5 Example of exception handling.

3.5 STRINGS:

String is an arrangement of characters. In java, objects of String are permanent which implies a steady and can't be changed once made.

Making a String:

There are two different ways to make string in Java:

String literal

String s = "PAYOOSNI "

Using *new* keyword

String s = new String ("PAYOOSNI");

3.5.1 StringMethods:

1. char charAt(int list) - Restores the character at the predefinedlist.
2. int compareTo(Object o) - Examines this String to anotherObject.
3. int compareTo(String anotherString) -Examines two stringslexicographically.
4. int compareToIgnoreCase(String str) - Examines two strings lexicographically, disregarding casecontrasts.
5. Stringconcat(Stringstr)-Connectsthepredefinedstringasfaraspossibleofthisstring.
6. boolean equals(Object anObject) -Looks at this string to the predetermineditem.
7. int indexOf(int ch) -Returns the list inside this string of the principal event of the predeterminedcharacter.
8. int indexOf(String str) -Restores the file inside this string of the primary event of the predefinedsubstring.
9. int lastIndexOf(int ch, int fromIndex) -Restores the file inside this string of the last event of the predetermined character, looking through in reverse beginning at the predefinedrecord.
10. intlastIndexOf(Stringstr)-Restoresthefileinsidethisstringofthefurthestrightevent of the predeterminedsubstring.

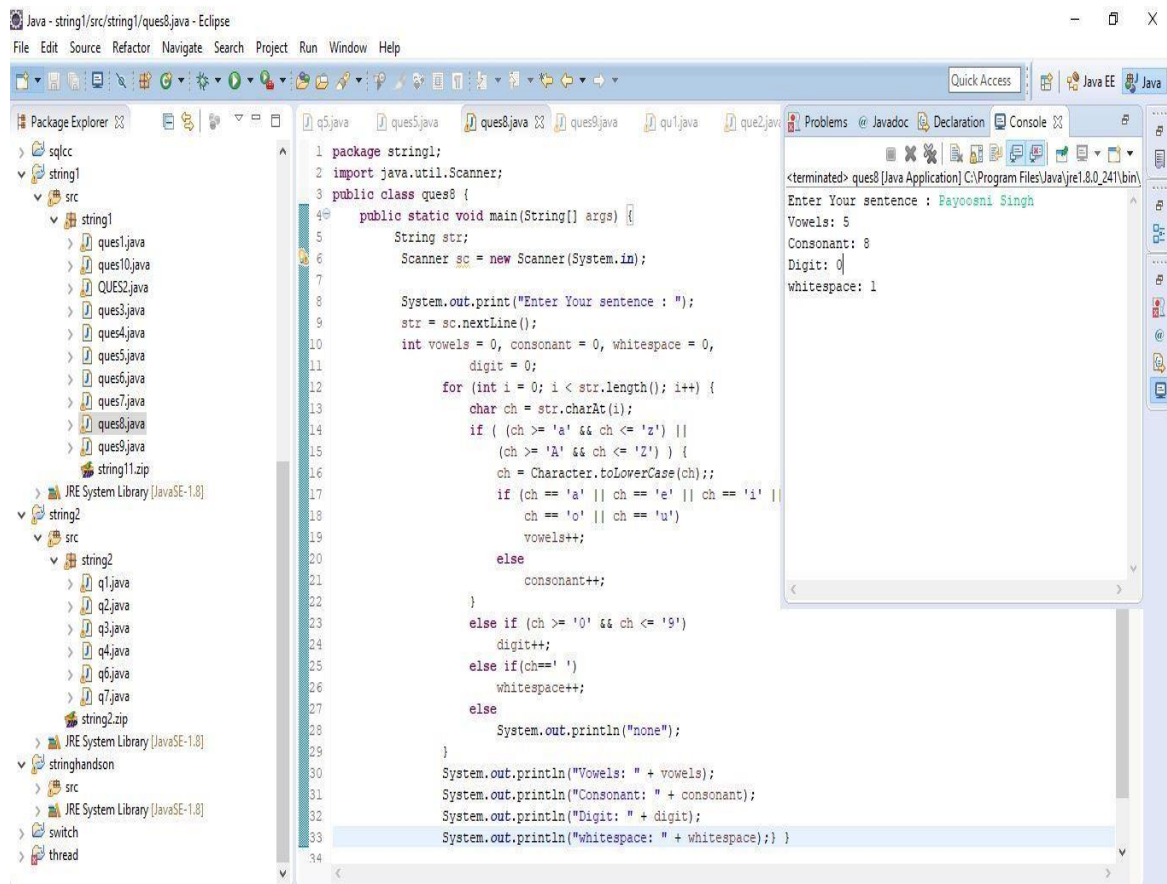


Figure 3.6 Example with String methods

3.5.2 StringBuffer

Java StringBuffer class is alterable (modifiable) string. The StringBuffer class in java is alike to String class with the exception of it is impermanent for example it very well may be changed.

Important Constructors of StringBuffer class:

	Constructor	Description
1.	StringBuffer()	makes an unfilled string support with the underlying limit of 16.
2.	StringBuffer(String str)	creates a string support with the predetermined string.
3.	StringBuffer(int capacity)	creates an unfilled string support with the predefined limit as length.

Method	Syntax	Description
append(char Sw)	public StringWriter append(char Sw)	appends a single character to the String Buffer.
append(CharSequence char_sq)	public StringWriter append(CharSequence char_sq)	appends specified character sequence to the String Buffer..
append(CharSequence char_sq, int start, int end)	public StringWriter append(CharSequence char_sq, int start, int end)	appends specified part of a character sequence to the String Buffer..
flush()	public void flush()	flushes the StringWriter stream
getBuffer()	public StringBuffer getBuffer()	returns the String Buffer of StringWriter.
write(int char)	public void write(int char)	writes a single character to the String Buffer.
write(String str)	public void write(String str)	writes string to the String Buffer.
write(String str, int offset, int maxlen)	public void write(String str, int offset, int maxlen)	writes some part of the string to the String Buffer.
write(char[] carray, int offset, int maxlen)	public void write(char[] carray, int offset, int maxlen)	writes some part of the character array to the String Buffer.
toString()	public String toString()	returns value of the String Buffer as string
close()	public void close()	closes String Buffer. Since, method can also be called after closing the String Buffer so, it play no role.

Figure 3.7 Important methods of StringBuffer class.

```

1 package string2;
2 import java.util.*;
3 public class q6 {
4
5
6
7
8
9
10
11     public static void main(String args[])
12     {
13         Scanner sc = new Scanner(System.in);
14         System.out.println("Enter the string: ");
15         String str = sc.nextLine();
16         System.out.println("Enter the string to be replaced: ");
17         String old_str = sc.nextLine();
18         System.out.println("Enter the new string: ");
19         String new_str = sc.nextLine();
20         String replaced = str.replace(old_str, new_str);
21         System.out.println("replaced string: " + replaced);
22     }
23 }

```

Console Output:

```

<terminated> q6 (!) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\
Enter the string:
college report by payoooni
Enter the string to be replaced:
college
Enter the new string:
internship
replaced string: internship report by payoooni

```

Figure 3.8: Example of StringBuffer.

3.6 Arrays inJava

An array is a holder object that holds a fixed number of estimations of a solitary kind. The length of a cluster is set up when the exhibit is made. After creation, its length is fixed.

- In Java all exhibits are powerfully allocated.

Exhibit can contain native’s information types just as objects of a class contingent upon the meaning of cluster. If there should arise an occurrence of natives information types, the real qualities are put away in bordering memory areas. If there should be an occurrence of objects of a class, the genuine items are put away in load section.

Creating, Initializing, and Accessing an Array

```
// create an array of integers
```

```
anArray = new int[10];
```

```
//Initialization
```

```
anArray[0] = 100;/instate first component
```

```
anArray[1] = 200;/introduce second component
```

```
anArray[2] = 300;/, etc
```

```
//Accessing an array
```

```
System.out.println("Element 1 at file 0: " + anArray[0]);
```

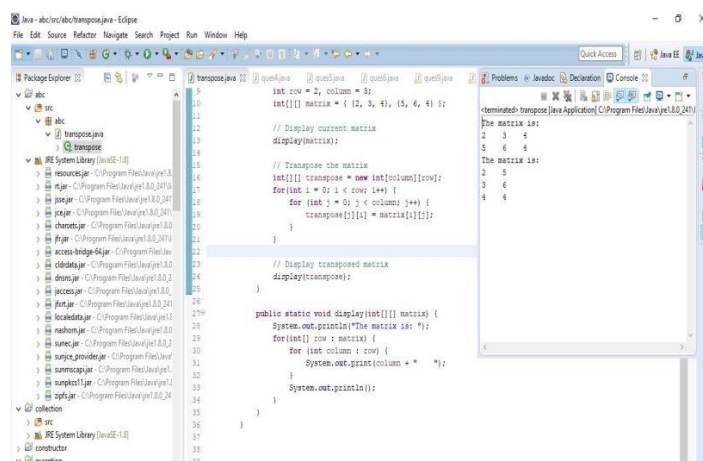


Figure 3.9 Array Example in Eclipse

3.7 Collections in Java

The Collection in Java is a structure that gives a design to store and control the social affairs of articles.

Java Collections can achieve all the exercises that you perform on data, for instance, looking, organizing, consideration, control, and retraction.

Java Collection infers a lone unit of things. Java Collection framework gives various interfaces (Set, List, Queue, Deque) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

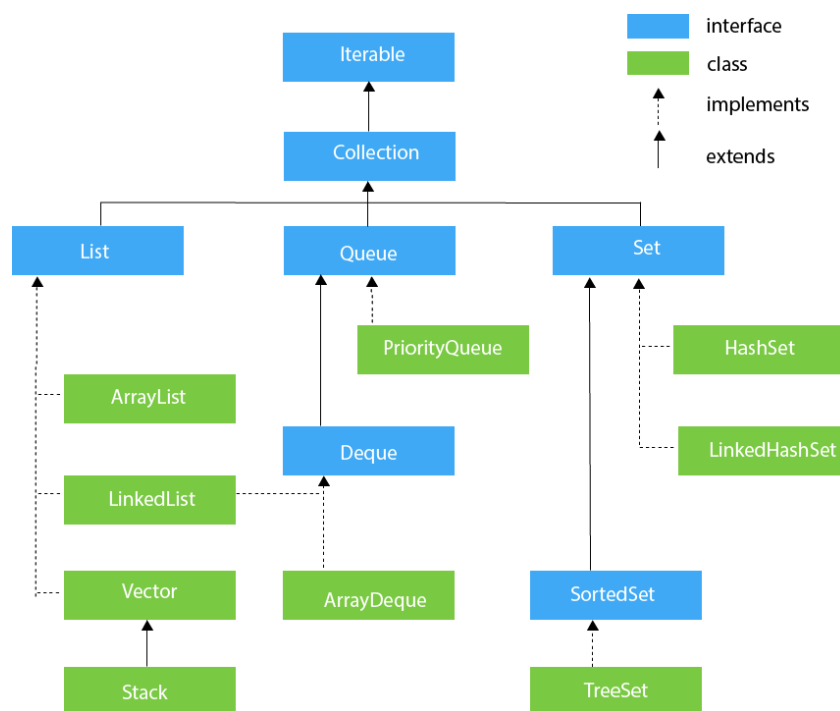


Figure 3.10 : Chain of command of Collection Framework.

3.7.1 ArrayList

ArrayList class actualizes List interface and it depends on an Array information structure. It is generally utilized as a result of the usefulness and adaptability it offers. A large portion of the engineers pick ArrayList over Array as it's a generally excellent option of conventional java exhibits.

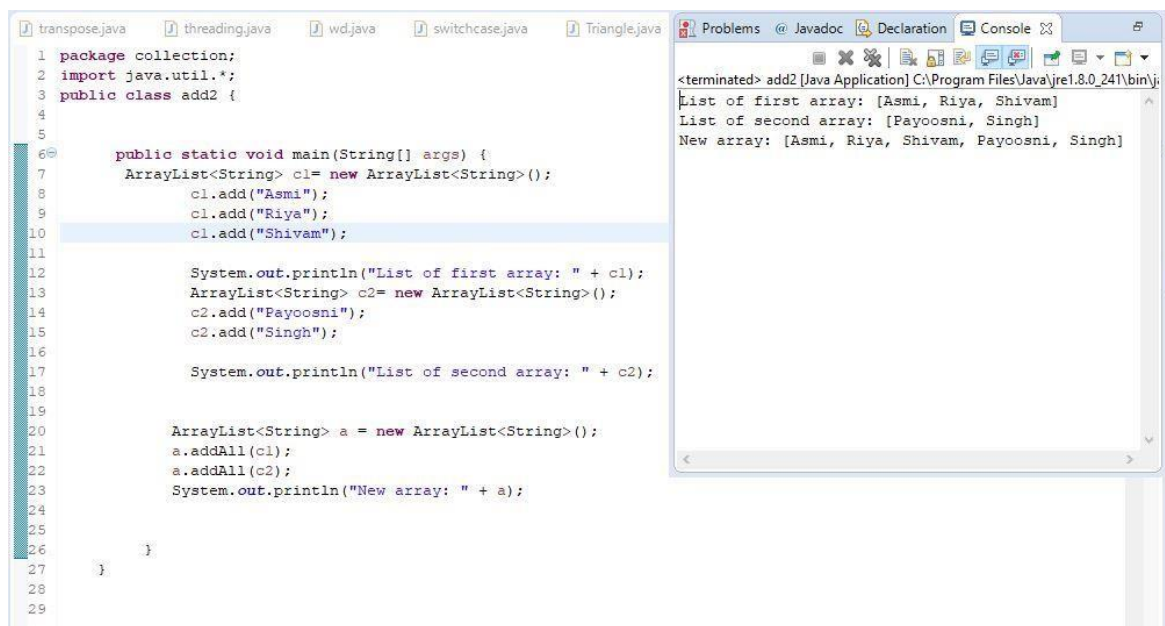


Figure 3.11 – Example of ArrayList

3.7.2 LinkedList

LinkedList is a straight information structure. Its components are not put away in touching areas like clusters; they are connected with one another utilizing pointers. Every component of the LinkedList has the reference (address/pointer) to the following component of the LinkedList.

LinkedList Class Features

Underneath you can locate the most significant properties of the class LinkedList:

- Actualizes Queue and Deque interfaces. Along these lines, It can likewise be utilized as a Queue, Deque or Stack
- It can contain all the components, including copies and invalid
- LinkedList keeps up inclusion request of the components
- Java LinkedList Class isn't synchronized, that implies in a multi-strung condition you should synchronize simultaneous adjustments to the connected rundown remotely.

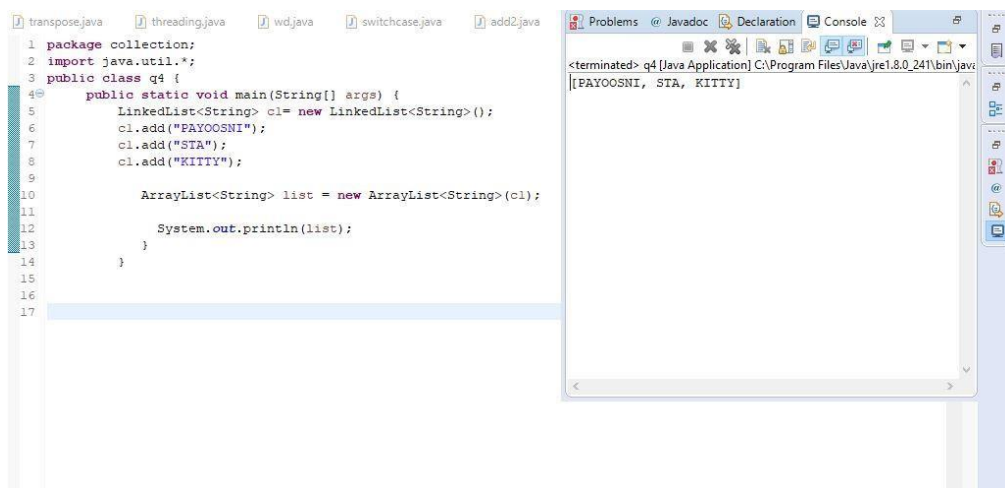


Figure 3.12 – Example of LinkedList

3.7.3 PriorityQueue

A PriorityQueue is utilized when the articles should be handled dependent on the need. It is realized that a line follows First-In-First-Out calculation; however now and then the components of the line should have been prepared by the need, that is the point at which the PriorityQueue becomes an integral factor. The PriorityQueue depends on the need pile. The components of the need line are requested by the regular requesting, or by a Comparator given at line development time, contingent upon which constructor is used.

Few significant focuses on PriorityQueue are as per the following:

- We can't make PriorityQueue of Objects that arenon-equivalent
- PriorityQueue are unboundlines.
- The leader of this line is minimal component as for the predefined requesting. On the off chance that different components are tied for least worth, the head is one of those components — ties are brokensubjectively.

```

1 package collection;
2 import java.util.*;
3 public class priorityqueue {
4
5
6
7     public static void main(String args[]){
8         PriorityQueue<String> queue=new PriorityQueue<String>();
9         queue.add("Amit Sharma");
10        queue.add("Vijay Raj");
11        queue.add("JaiShankar");
12        queue.add("Raj");
13        System.out.println("head:"+queue.element());
14        System.out.println("head:"+queue.peek());
15        System.out.println("iterating the queue elements:");
16        Iterator itr=queue.iterator();
17        while(itr.hasNext()){
18            System.out.println(itr.next());
19        }
20        queue.remove();
21        queue.poll();
22        System.out.println("after removing two elements:");
23        Iterator<String> itr2=queue.iterator();
24        while(itr2.hasNext()){
25            System.out.println(itr2.next());
26        }
27    }
28 }

```

```

<terminated> priorityqueue [Java Application] C:\Program Files\Java\jre1.8.0_2
head:Amit Sharma
head:Amit Sharma
iterating the queue elements:
Amit Sharma
Raj
JaiShankar
Vijay Raj
after removing two elements:
Raj
Vijay Raj

```

Figure 3.13 Example of PriorityQueue.

3.7.4 HashSet

Java HashSet class is an individual from Java assortments system. It actualizes the Set interface. HashSets are utilized to store an assortment of one of kind components

Following are not many key focuses to note about HashSet in Java -

- HashSet can't contain copyesteems.
- HashSet permits invalid worth.

- HashSet is an unordered assortment. It doesn't keep up the request in which the components are embedded.
- HashSet inside utilizes a HashMap to store its components.
- HashSet isn't string safe. On the off chance that various strings attempt to change a HashSetsimultaneously,atthatpointtheultimateresultisn'tdeterministic. You should unequivocally synchronize simultaneous access to a HashSet in a multi-strung condition.

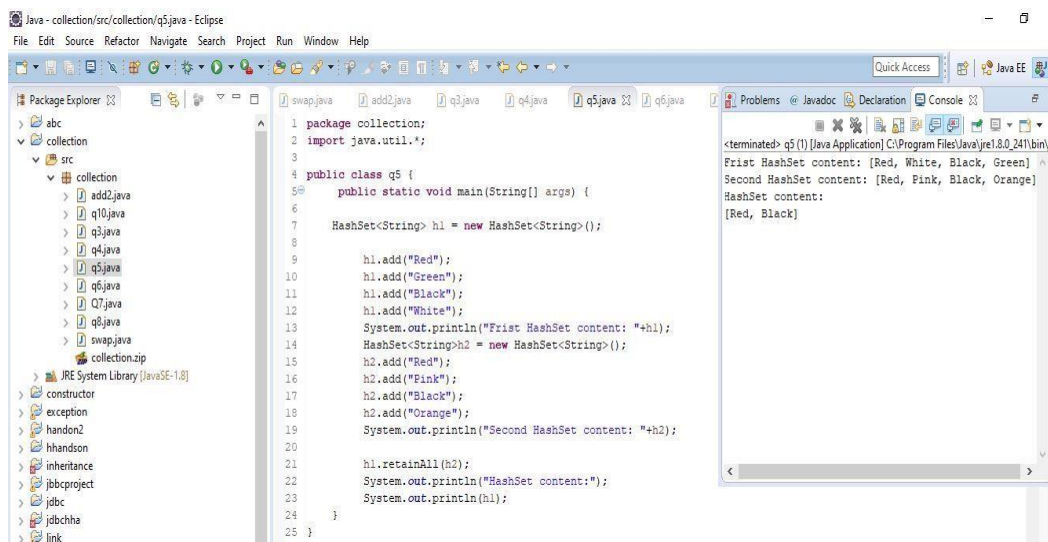


Figure 3.14 Example of HashSet

3.7.5 TreeSet

TreeSet is one of the most significant executions of the SortedSet interface in Java that utilizes a Tree for capacity. TreeSet actualizes the SortedSet interface so copy esteems are not permitted. Articles in a TreeSet are put away in an arranged and rising request. Posting down a portion of the top highlights of TreeSet:

- They store one of kind information. Henceforth they decay copy additions.

- TreeSet sorts the components dependent on their natural request.
- Be that as it may, you can give a Comparator occurrence so as to modify arranging.
- TreeSet permits just a single invalid component.
- The activities like include, contains, and evacuate are steady intime.

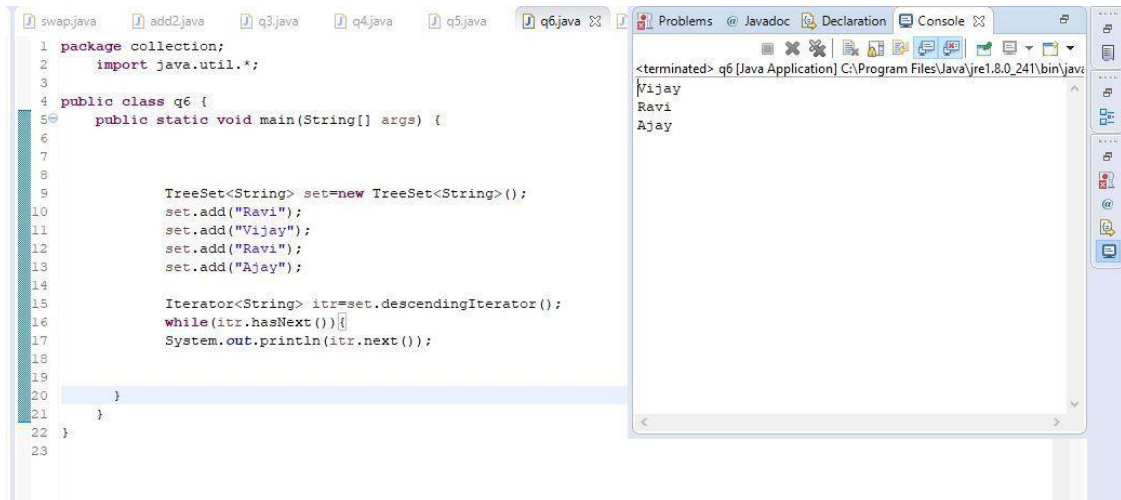


Figure 3.15 Example of TreeSet

3.8 Java Database Connectivity with MySQL

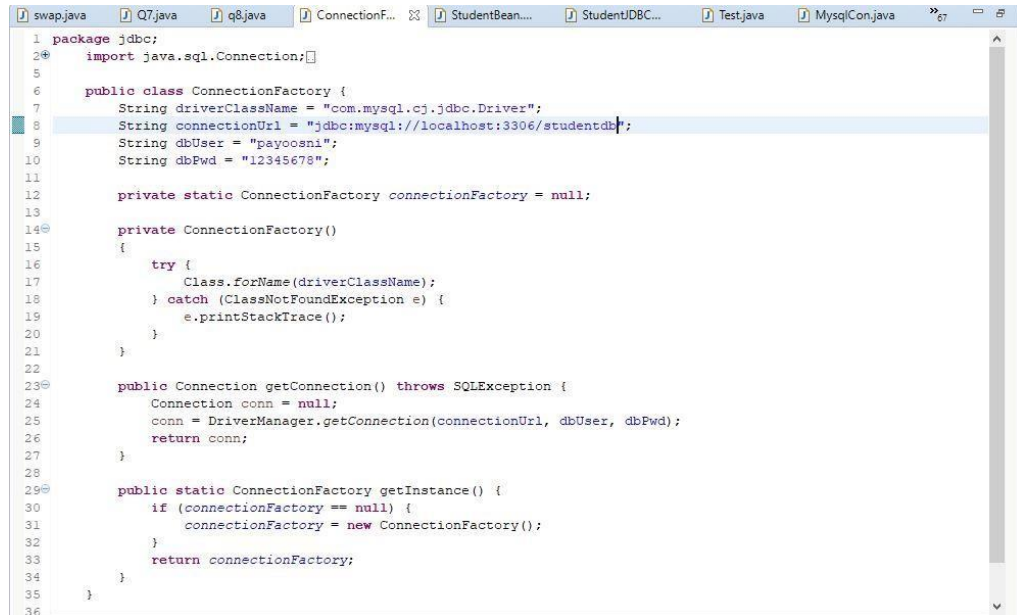
To interface Java application with the MySQL database, we have to follow 5 after advances.

In this model we are utilizing MySQL as the database. So we have to know the following information for the MySQL database:

1. Driver class: The driver class for the MySQL database is `com.mysql.jdbc.Driver`.
2. Connection URL: The association URL for the MySQL database is `jdbc:mysql://localhost:3306/sonoo` where `jdbc` is the API is running, we may likewise utilize IP address. We may utilize any database, in such case, we have to supplant the `sonoo` with our database name.

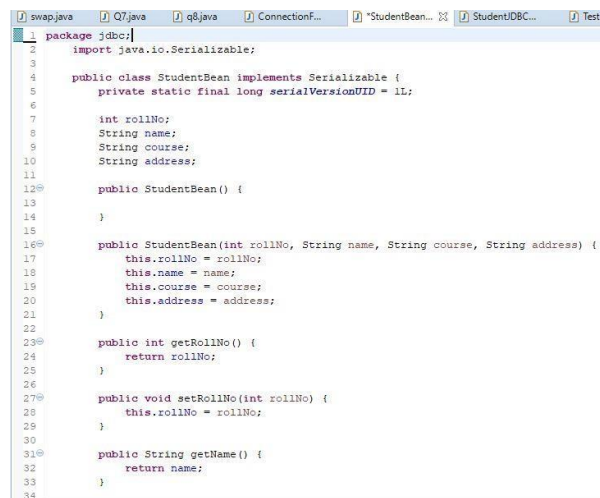
3. Username: The default username for the mysql database isroot.

4. Password: It is the secret phrase given by the client at the hour of introducing the MySQL database. In this model, we are going to utilize root as the secretkey.



```
1 package jdbc;
2 import java.sql.Connection;
3
4
5
6 public class ConnectionFactory {
7     String driverClassName = "com.mysql.cj.jdbc.Driver";
8     String connectionUrl = "jdbc:mysql://localhost:3306/studentdb";
9     String dbUser = "payoosni";
10    String dbPwd = "12345678";
11
12
13
14    private static ConnectionFactory connectionFactory = null;
15
16    private ConnectionFactory()
17    {
18        try {
19            Class.forName(driverClassName);
20        } catch (ClassNotFoundException e) {
21            e.printStackTrace();
22        }
23    }
24
25    public Connection getConnection() throws SQLException {
26        Connection conn = null;
27        conn = DriverManager.getConnection(connectionUrl, dbUser, dbPwd);
28        return conn;
29    }
30
31    public static ConnectionFactory getInstance() {
32        if (connectionFactory == null) {
33            connectionFactory = new ConnectionFactory();
34        }
35        return connectionFactory;
36    }
37 }
```

Figure 3.16 Java Database Connectivity with MySql



```
1 package jdbc;
2 import java.io.Serializable;
3
4 public class StudentBean implements Serializable {
5     private static final long serialVersionUID = 1L;
6
7     int rollNo;
8     String name;
9     String course;
10    String address;
11
12    public StudentBean() {
13    }
14
15
16    public StudentBean(int rollNo, String name, String course, String address) {
17        this.rollNo = rollNo;
18        this.name = name;
19        this.course = course;
20        this.address = address;
21    }
22
23    public int getRollNo() {
24        return rollNo;
25    }
26
27    public void setRollNo(int rollNo) {
28        this.rollNo = rollNo;
29    }
30
31    public String getName() {
32        return name;
33    }
34 }
```

Figure 3.17 Code in Eclipse

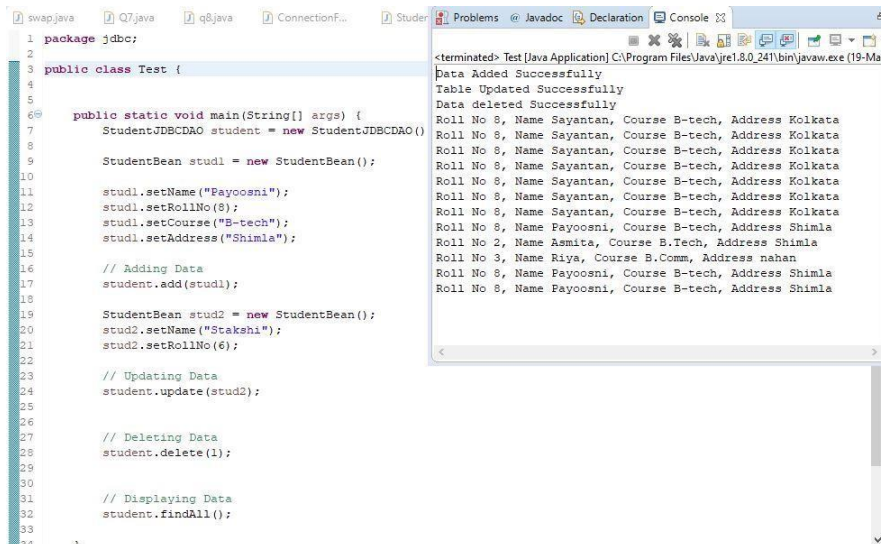


Figure 3.18 Main method in eclipse

The changes can be seen in the MySQL file now.

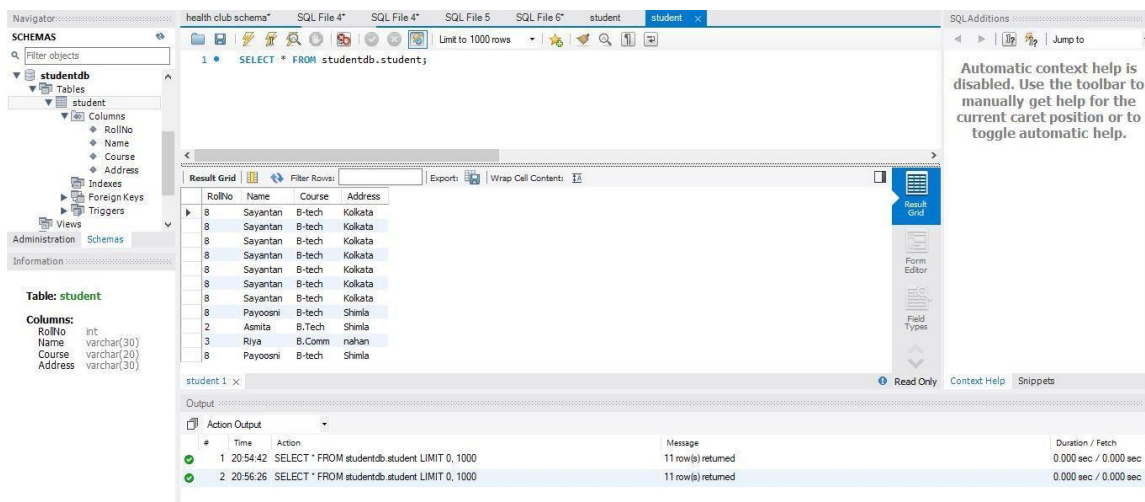


Figure 3.19 Workbench

CHAPTER 4

Tools of Automated Testing: Introduction to MySQL

SQL represents Structured Query Language. It is a standard language used to speak with the databases.

SQL is utilized to perform assignments on database. It permits to connect with the information.

4.1 What is MySQL?

MySQL is a social database the board framework. It gives a UI to us to connect with the database.

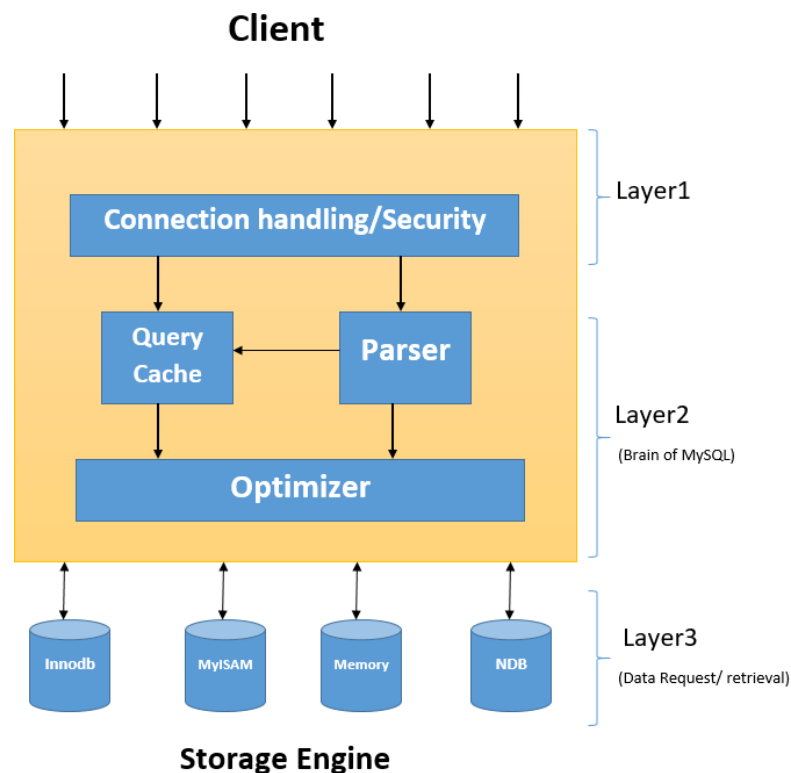


Figure 4.1 MySQL Database Administrators

4.2 What is RDBMS?

A social database is an assortment of information sorted out into tables. Tables contain sections of information classes and columns with specific cases of that information classification. For instance,

Tables in a social database can be connected together.

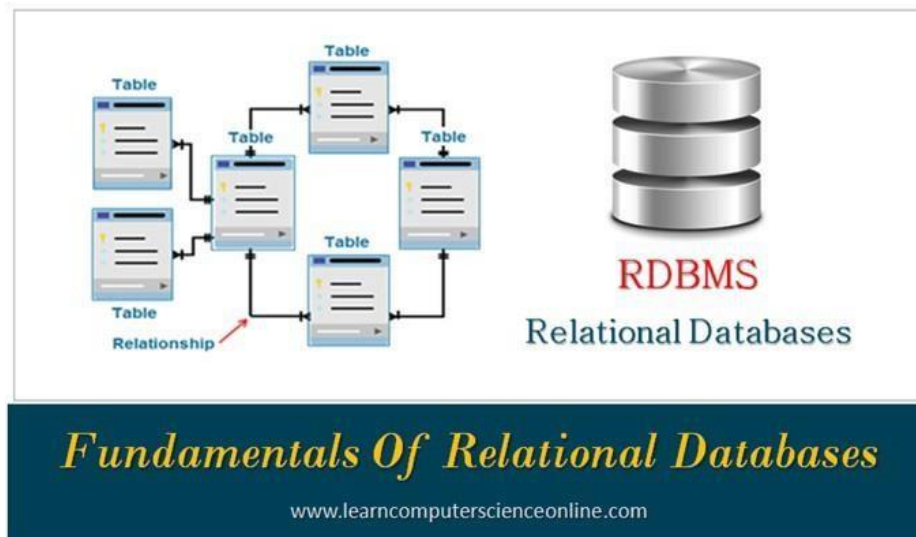


Figure 4.2 Fundamentals of Relational Databases

4.3 Installation of MySQL

Step 1:

The initial step is to go to the site www.dev.sql.com and introduce MySQL for the necessary working framework.

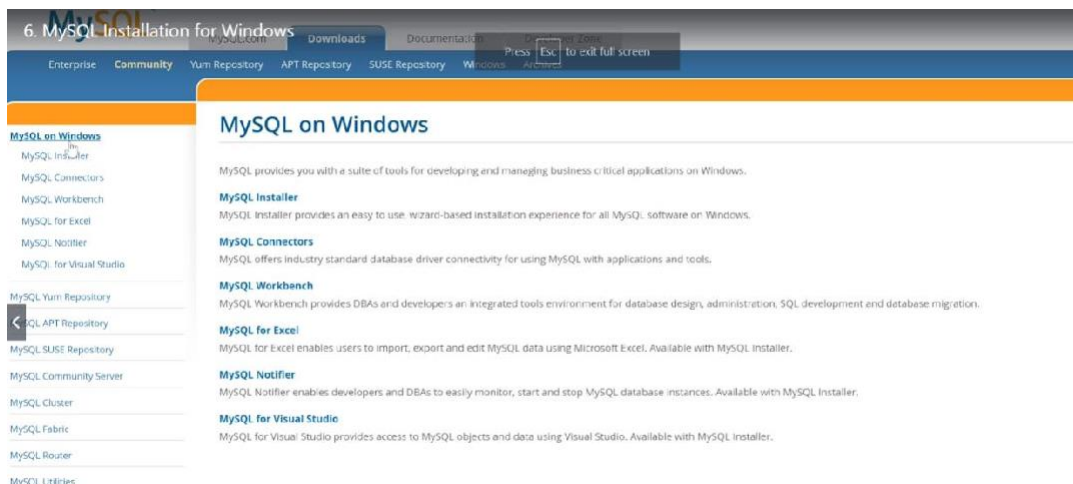


Figure 4.3 (a) Step 1

Step2:

Subsequent to introducing the MySQL to your framework, fire setting up the workbench stage where you can compose the questions.

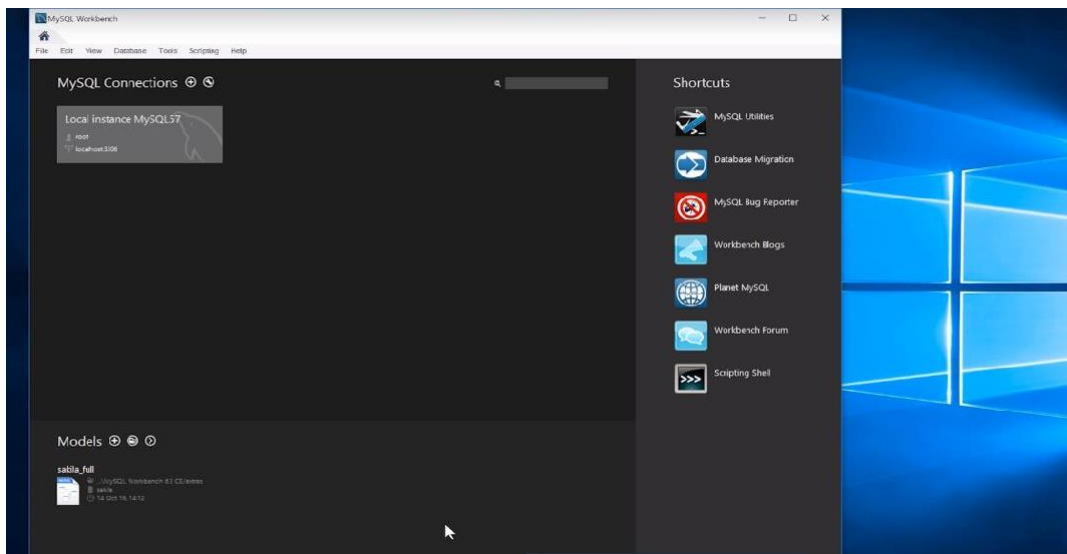


Figure 4.4 - Step 2 of Installing MySQL

Step3:

At the point when you open the workbench, a secret phrase brief will spring up and you have to enter the root secret word that you made during the establishment procedure.

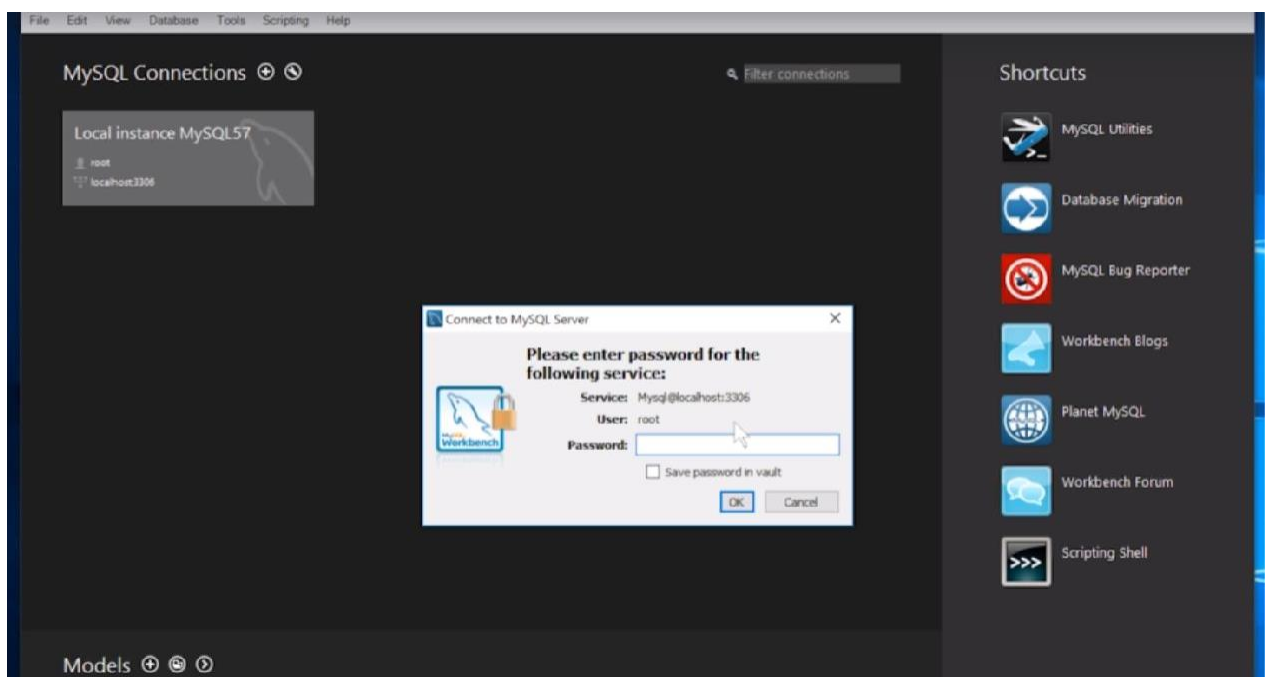


Figure 4.5 - Step 3 of Installing MySQL

Step4:

Presently you can make your first database inside the workbench.

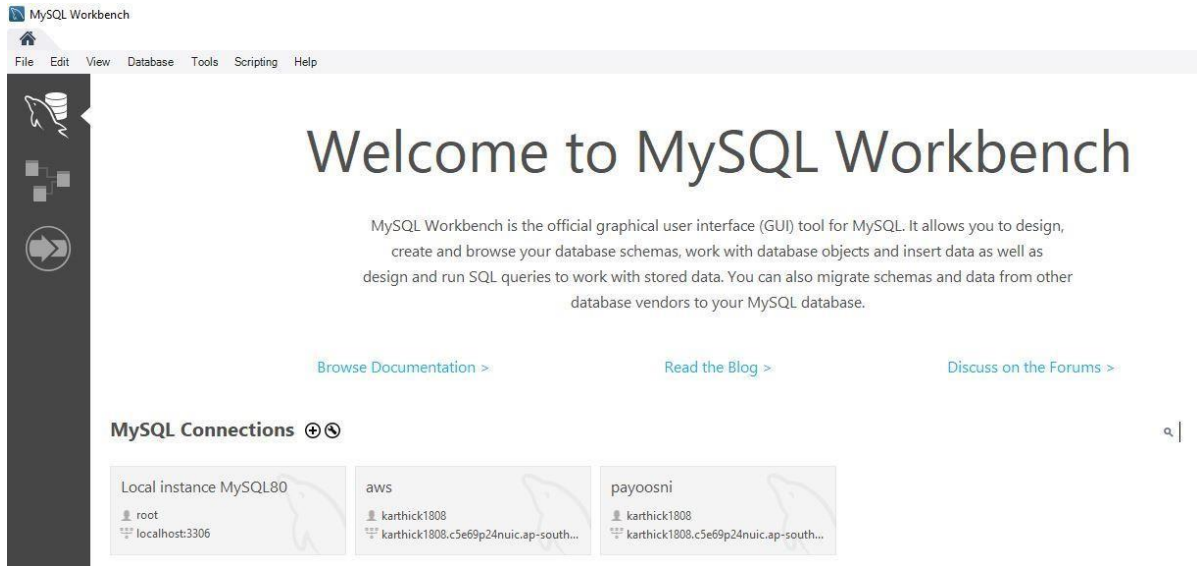


Figure 4.6 - Step 4 of Installing MySQL

4.4 Data Definition Language

Information definition language essentially comprises of the SQL orders which can be use to characterize the composition of the database. The orders remembering for this are: CREATE, ALTER, DROP, TRUNCATE.

2.4.1 Data Types

- 1) Numeric Datatype:
 - INT: Wholenumbers
 - FLOAT(M,D): Decimalnumbers(approx.)
 - DECIMAL(M,D): Decimalnumbers(precise)
- 2) Non-Numeric Data type:
 - CHAR(N): Fixed lengthcharacter
 - VARCHAR(N): Varying length character
 - ENUM('M','F'): Value from a characterizedlist
 - BOOLEAN: True or Falsequalities

- 3) Date and Timetype:
 - DATE:Date(yyyy-mm-dd)
 - DATETIME: Date and time (yyyy-mm-ddhh-mm-ss)
 - TIME:Time(hh-mm-ss)
 - YEAR:Year(yyyy)

4.4.2 Primary and ForeignKeys

Essential Key: It is a section or the arrangements of segments which exceptionally perceive a record inside the given table. Following are the conditions that should be fulfilled for a segment to be an essential key:

- It can't be invalid
- It ought to be one of a kind
- A table ought to have one essential key

Outside Key: It is utilized to interface two tables together. A remote key is a segment whose qualities coordinate the estimations of another table's essential key segment.

The screenshot shows a SQL query in a text editor window titled 'Statement 1'. The query is:


```

1 CREATE TABLE test_results
2 (
3   name          TEXT,
4   student_id   INTEGER PRIMARY KEY,
5   birth_date   DATE,
6   test_result  DECIMAL NOT NULL,
7   grade        TEXT NOT NULL,
8   passed       BOOLEAN NOT NULL
9 );
  
```

 Below the query, a 'Messages' pane displays the output:


```

Table test_results created
Execution time: 0.11s
  
```

Figure 4.7 Creating a Table with primary key

4.4.3 Modifying Tables: Adding and Removing Tables

To include another segment in the current made table the language structure is: "Include COLUMN <COL NAME><DATATYPE>".

To expel the section from a table the grammar is: "DROP COLUMN <COL NAME>".

What's more, so as to erase all the information from inside the tables shorten order is utilized.

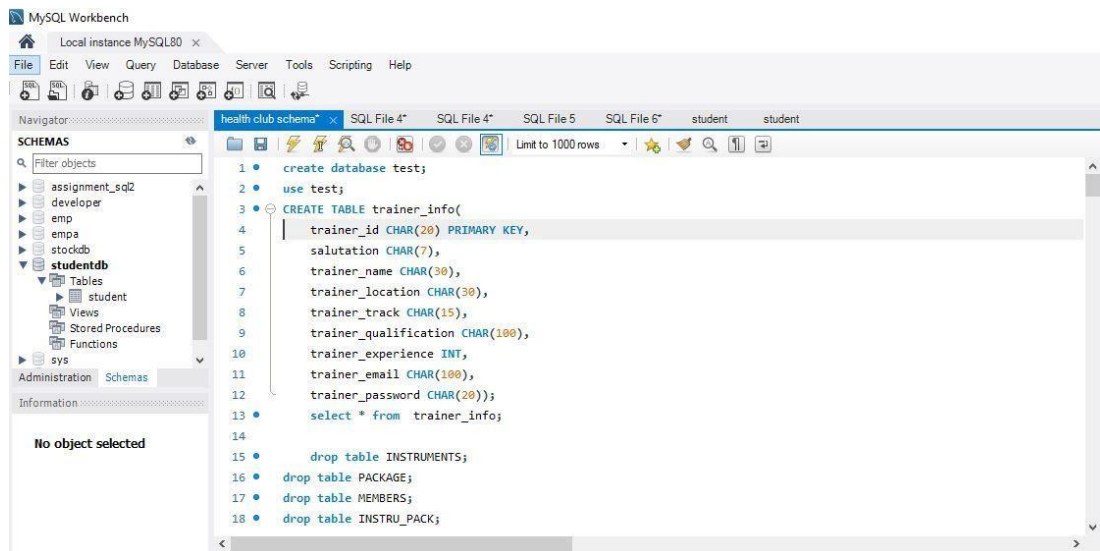


Figure 2.8 Creating a Table

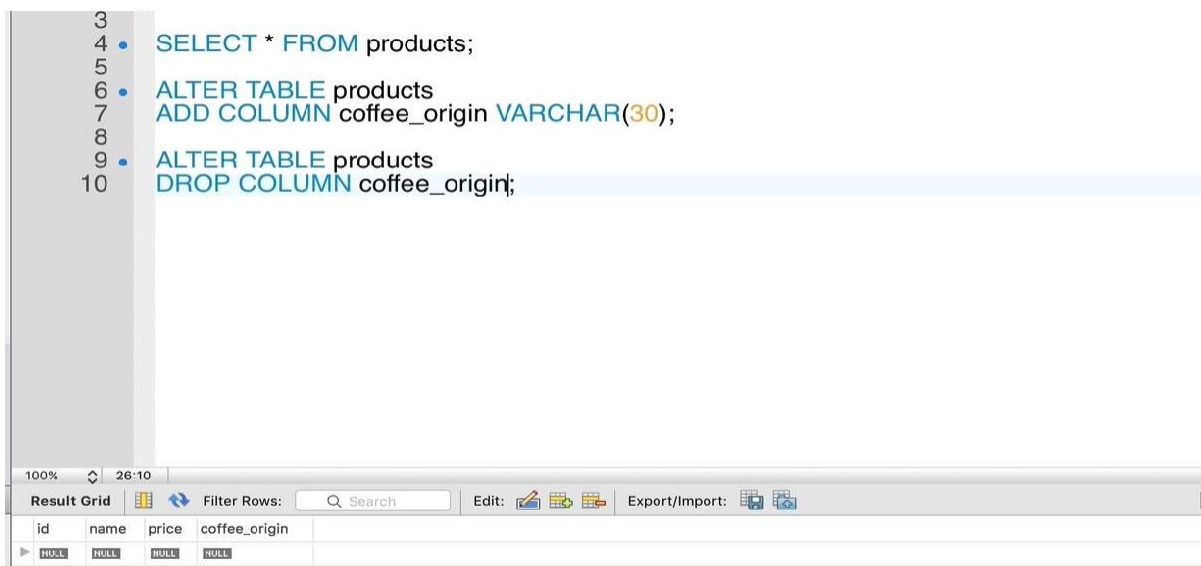


Figure 2.9 Adding and removing column.

4.5 Data Manipulation Language

The information control language is utilized to get to, adjust or to recover the given measure of information from the database. It is essentially used to populate and control the information. The orders utilized under DML are: INSERT, SELECT, UPDATE, DELETE.

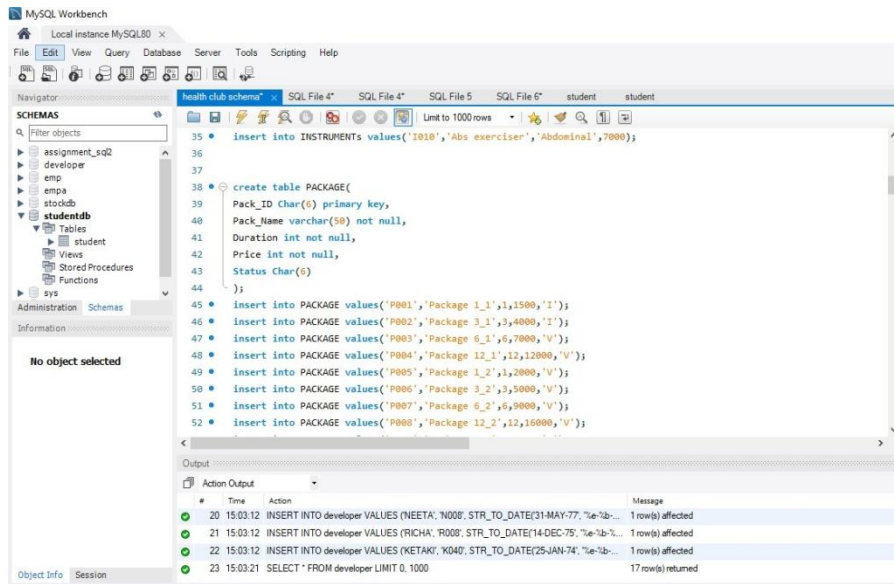


Figure 2.10 Inserting values.

4.6 About Joins

Joins permit to recover information from different tables in a solitary select articulation. To join two tables there should be a related segment between them. There are various sorts of joins, for example, inward join, left join, right join.

4.6.1 Inner Join

Inward join will recover information just when there is coordinating qualities in both the tables.

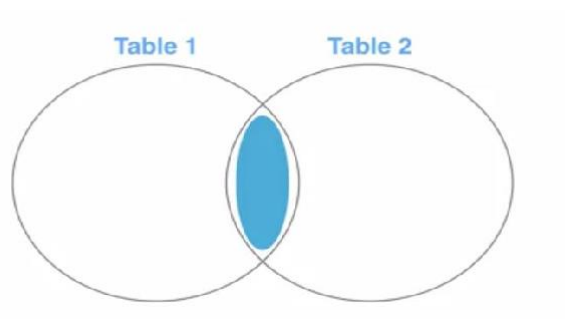


Figure 4.11 Inner Join

```
1 • USE coffee_store;
2
3
4 • SELECT * FROM products;
5 • SELECT * FROM orders;
6
7 • SELECT products.name, orders.order_time FROM orders
8 INNER JOIN products ON orders.product_id = products.id;
9
10 • SELECT p.name, p.price, o.order_time FROM orders o
11 JOIN products p ON o.product_id = p.id
12 ORDER BY;
13
14
15
16
17
18
```

Figure 4.12 Query of Inner Join

4.6.2 LeftJoin

Left join will recover all the information from the left table and the coordinating lines from the correct side of the table.

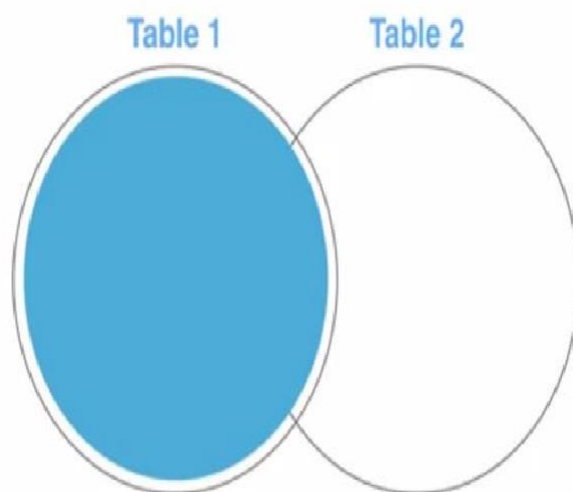


Figure 4.13 Left join

```
1 • USE coffee_store;
2
3
4 /*
5 UPDATE orders
6 SET customer_id = 1
7 WHERE id = 1;
8 */
9 • SELECT * FROM customers;
10 • SELECT * FROM orders;
11
12 • UPDATE orders
13 SET customer_id = NULL
14 WHERE id = 1;
15
16 • SELECT o.id, c.phone_number, c.last_name, o.order_time FROM orders o
17 LEFT JOIN customers c ON o.customer_id = c.id
18 ORDER BY o.order_time
19 LIMIT 10;
20
```

Figure 4.14 Query of Left Join

4.6.3 RightJoin

The correct join will recover the information from the correct table and coordinating columns of the left table.

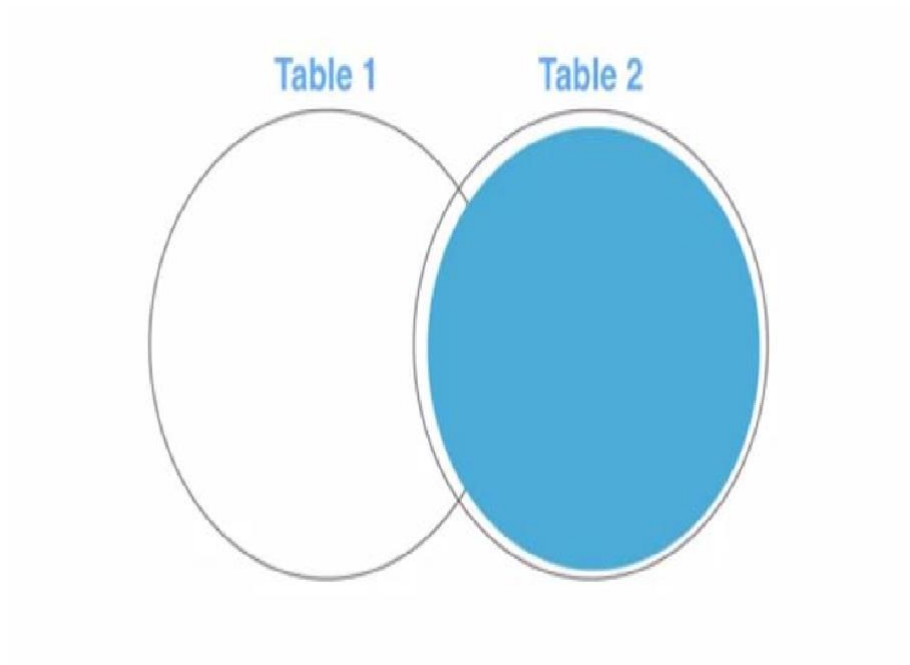


Figure 4.15 Right Join

```

1 • USE coffee_store;
2
3
4 UPDATE orders
5 SET customer_id = 1
6 WHERE id = 1;
7 */
8
9 • SELECT * FROM customers;
10 • SELECT * FROM orders;
11
12 • UPDATE orders
13 SET customer_id = NULL
14 WHERE id = 1;
15
16 • SELECT o.id, c.phone_number, c.last_name, o.order_time FROM customers c
17 RIGHT JOIN orders o ON c.id = o.customer_id
18 ORDER BY o.order_time
19 LIMIT 10;
20

```

Figure 4.16 Query of Right Join.

4.7 Subqueries

Subqueries are the inquiries that are settled inside different questions. The subqueries can be utilized in a SELECT, INSERT, UPDATE, or DELETE question. The settled question could be in the WHERE provision or in the FROM. There are two kinds of subqueries to be specific, non-corresponded subquery and related subquery.

4.7.1 Non-Correlated Subquery

In this sort of subquery, the inward question can run freely of the external inquiry. Inward questions run first and produce an outcome set and which is then utilized by the external inquiry.

4.7.2 Correlated Subquery

In this sort of subquery, the internal question runs for each line in the external inquiry. Here the internal question can't run freely of the external inquiry.

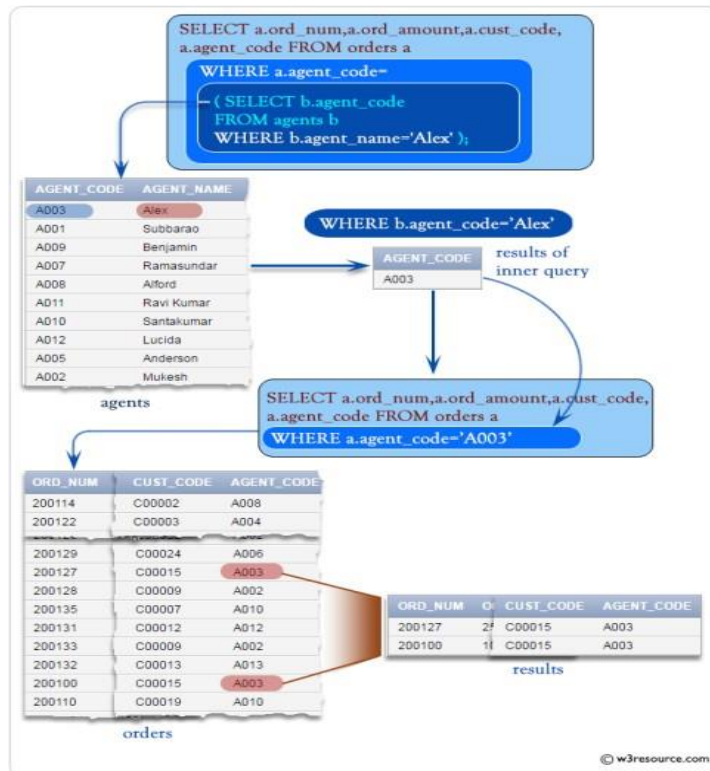


Figure 4.17 Correlated Subquery

4.8 Normalization

Standardization is the procedure of effectively sorting out the information into the database. It is fundamentally done so as to expel the excess information and to just store the related information in a table. The advantages of standardization are:

- Reduce the extraroom
- Reduce addition, update, and cancellationinconsistencies
- Improve inquiryexecution
- Reduce the extraroom
- Reduce addition, update, and cancellationinconsistencies
- Improve inquiryexecution

Following are the degrees of standardization:

- 1st typical structure(1NF)
- 2nd typical structure(2NF)
- 3rd typical structure (3NF)
- Boyce and Codd typical structure(BCNF)

4.8.1 First Normal Form(1NF)

The standards in the main typical structure are that a table is in the 1NF if:

- There are no rehashed lines of information
- The sections just contain a solitaryworth
- The table has an essential key

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721, 9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 1

Conversion to first normal form

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY
1	RAM	9716271721	HARYANA	
1	RAM	9871717178	HARYANA	INDIA
2	RAM	9898297281	PUNJAB	INDIA
3	SURESH		PUNJAB	INDIA

Table 2

Figure 4.18 1NF

4.8.2 Second Normal Form(2NF)

The standards in the subsequent ordinary structure are that a table is in the 2NF if:

- They fit in with 1NF
- Every segment that is anything but an essential key of the table is subject to the entire of the essential key

1NF to 2NF – Example

1NF				2NFa			2NFb	
Module	Dept	Lecturer	Text	Module	Dept	Lecturer	Module	Text
M1	D1	L1	T1	M1	D1	L1	M1	T1
M1	D1	L1	T2	M2	D1	L1	M1	T2
M2	D1	L1	T1	M3	D1	L2	M2	T1
M2	D1	L1	T3	M4	D2	L3	M2	T3
M3	D1	L2	T4	M5	D2	L4	M3	T4
M4	D2	L3	T1				M4	T1
M4	D2	L3	T5				M4	T5
M5	D2	L4	T6				M1	T6

Figure 4.19 2NF

4.8.3 Third Normal Form(3NF)

The guidelines in the third typical structure are that a table is in the 3NF if:

- They fit in with 2NF.
- Every segment that isn't the essential key is just subject to the entire of the essential key.

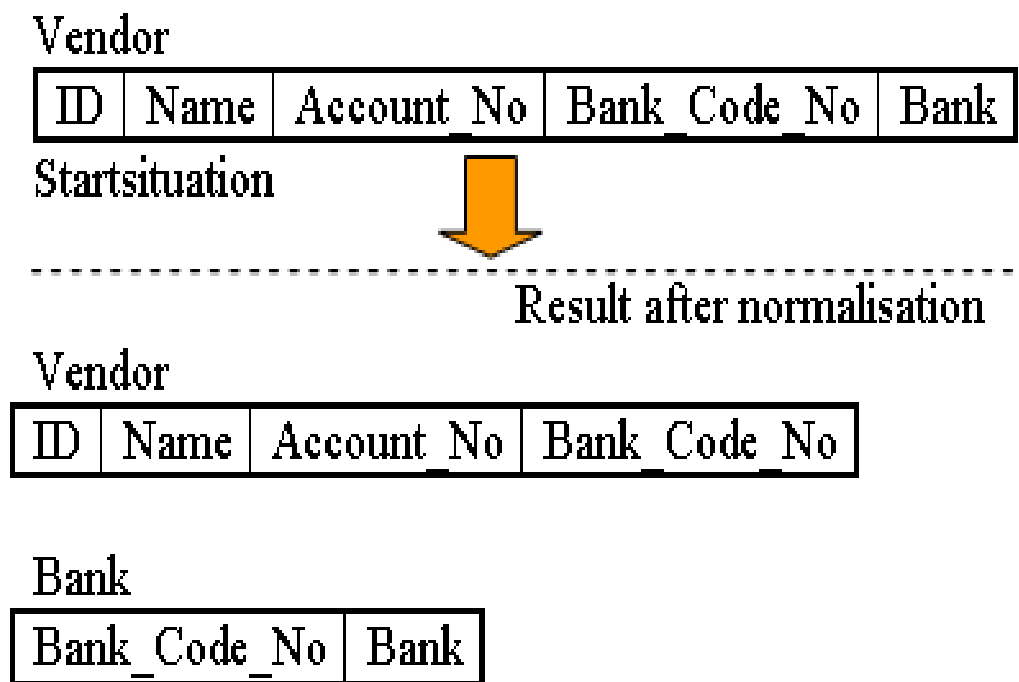


Figure 4.20 3NF example.

CHAPTER 5

Basics of Automated Testing: XPath and Selenium

5.1 What is XPath

In automated testing, if the components are not found by the overall finders like `id`, `class`, `name`, and so on then XPath is utilized to discover a component on the site page.

XPath in Selenium is a XML way utilized for route through the HTML design of the page. It is a grammar or language for discovering any component on a website page utilizing XML way articulation. XPath can be utilized for both HTML and XML archives to discover the area of any component on a site page utilizing HTML DOM structure.

Syntax for XPath selenium:

XPath contains the way of the component arranged at the website page. Standard XPath sentence structure for making XPath is.

```
Xpath=//tagname[@attribute='value']
```

/: Select current hub.

TagName: Tagname of the specific hub.

@: Select property.

Characteristic: Attribute name of the hub.

Worth: Value of the property.

There are two sorts of XPath:

1) AbsoluteXPath

2) RelativeXPath

Absolute XPath:

It is the immediate method to discover the component, however the burden of the total XPath is that assuming there are any progressions made in the way of the component, that XPath gets fizzled.

The vital trait of XPath is that it starts with the single forward cut(/), which implies you can choose the component from the root hub.

Eg.

```
/html/body/div[2]/div[1]/div/h4[1]/b/html[1]/body[1]/div[2]/div[1]/div[1]/h4[1]/b[1]
```

5.2 Locating elements by XPath

1) Basic XPath:

XPath articulation select hubs or rundown of hubs based on ascribes like ID , Name, Classname, and so on from the XML record as represented beneath.

```
Xpath=//input[@name='uid']
```

Some more essential xpath articulations:

```
Xpath=//input[@type='text']
```

```
Xpath= //label[@id='message23']
```

```
Xpath= //input[@value='RESET']
```

```
Xpath=//*[ @class='barone']
```

```
Xpath=//a[@href='http://demo.com/']
```

```
Xpath=/img[@src='//cdn.guru99.com/pictures/home/java.png']
```

2) Contains():

Contains() is a strategy utilized in XPath articulation. It is utilized when the worth of any quality changes powerfully, for instance, login data.

The contain highlight has a capacity to discover the component with fractional content as demonstrated in beneath XPath model.

In this model, we attempted to distinguish the component simply by utilizing fractional content worth of the property. In the beneath XPath articulation halfway worth 'sub' is utilized instead of submit button. It tends to be seen that the component is found effectively.

Complete worth of 'Type' is 'submit' however utilizing just halfway worth 'sub'.

```
Xpath=//*[contains(@type,'sub')]
```

Complete worth of 'name' is 'btnLogin' yet utilizing just incomplete worth 'btn'.

```
Xpath=//*[contains(@name,'btn')]
```

3) Using ORAND:

In OR articulation, two conditions are utilized, regardless of whether first condition OR second condition ought to be valid. It is additionally material if any one condition is valid or possibly both. Means any one condition ought to be consistent with discover the component.

In the beneath XPath articulation, it distinguishes the components whose single or the two conditions are valid.

```
Xpath=//*[@type='submit' or @name='btnReset']
```

In AND articulation, two conditions are utilized, the two conditions ought to be consistent with discover the component. It neglects to discover component if any one condition is bogus.

```
Xpath=//input[@type='submit' and @id='bttLogin']
```

4) XPathStarts-with

XPath begins with() is a capacity utilized for discovering the web component whose quality worth gets changed on revive or by other powerful procedure on the site page. In this technique, the beginning content of the property is coordinated to discover the component whose trait esteem changes powerfully. You can likewise discover components whose property estimation is static (not changes).

For instance - : Suppose the ID of specific component changes powerfully like:

```
Id="message152"
```

```
Id=" message34523"
```

```
Id="message879"
```

etc.. be that as it may, the underlying content is same. For this situation, we use Start-with articulation.

In the underneath articulation, there are two components with an id beginning "message"(i.e., 'Client ID should not be clear' and 'Secret phrase should not be clear'). In beneath model, XPath discovers those component whose 'ID' beginning with 'message'.

Xpath=//label[starts-with(@id,'message')]

5.3 What is Selenium

Selenium is perhaps the most generally utilized open source Web UI (User Interface) mechanization testing suite. It was initially evolved by Jason Huggins in 2004 as an inner instrument at Thought Works. Selenium upholds robotization across various programs, stages and programming dialects.

Selenium can be handily conveyed on stages like Windows, Linux, Solaris and Macintosh. Besides, it upholds OS (Operating System) for portable applications like iOS, windows versatile and android.

Selenium upholds an assortment of programming dialects using drivers explicit to each language. Languages upheld by Selenium incorporate C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is generally well known with Java and C#. Selenium test contents can be coded in any of the upheld programming dialects and can be run straightforwardly in most present day internet browsers. Programs upheld by Selenium incorporate Internet Explorer, Mozilla Firefox, Google Chrome and Safari.

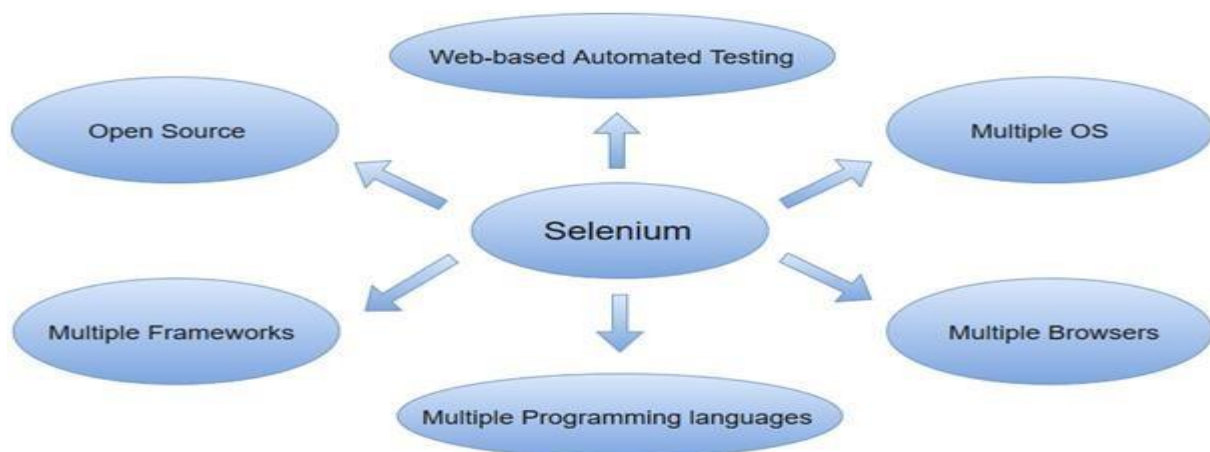


Figure 5.1 Selenium and its features

Selenium can be utilized to robotize useful tests and can be incorporated with computerization test devices like Maven, Jenkins, and Dockert to accomplish nonstop testing. It can likewise be coordinated with apparatuses like TestNG, and JUnit for overseeing experiments and creating

reports.

5.4 Selenium basiccommands

Given underneath are probably the most ordinarily utilized Selenium orders in WebDriver:

Fetching a Webpage

- *Using getmethod*

```
driver.get("www.facebook.com")
```

- *Using navigatemethod*

```
driver.navigate().to("https://facebook.com");
```

Locating forms and sending user inputs

```
driver.findElement(By.id("lst-ib")).sendKeys("MyName");
```

Clearing User inputs

```
driver.findElement(By.name("q")).clear();
```

Fetching data over any web element

```
driver.findElement(By.id("element567")).getText();
```

Performing Click event

```
driver.findElement(By.id("btnK")).click();
```

Navigating backward in browser history

```
driver.navigate().back();
```

Navigating forward in browser history

```
driver.navigate().forward();
```

Refresh/ Reload a web page

```
driver.navigate().refresh();
```

Closing Browser

```
driver.close();
```

Closing Browser and other all other windows active with the driver

```
driver.quit();
```

Conclusion

Throughout this report, I have gone through my learnings and experience acquired since day one in Cognizant as a Software Development Engineering and Testing Trainee. We were trained first on how to write and organize test cases according to the given software requirements. From there on our training shifted focus towards the software development part of our role and we were retrained on JAVA and SQL along with the basics of Web UI. Currently, we are learning about Selenium WebDriver, which is a tool to enable automated testing of web applications. Other than that we were regularly given Corporate Ethics classes so that we may orient our behavior towards the professionalism that Cognizant as a Service-based IT firm demands. I have learned and implemented a lot of new things since my onboarding at Cognizant and I hope I was able to portray what I have done in the organization since my condensed semester through my report.

Future Scope

In the future, I will be learning about the remaining concepts of Selenium web Driver as well as about the famous Java Framework Spring. I would also be learning more about DevOps tools like Maven and Cucumber.

According to our training schedule, we would also be assigned a business project where we would have to achieve the necessary milestones and complete the deliverables within given deadlines.

References

- [1] Cognizant Handbook
- [2] Internshipexperience
- [3] Assessment
- [4] Cognizant internshipcurriculum

PLAG REPORT ON THIS PROJECT

khkhkjkhkhkj

ORIGINALITY REPORT

29% SIMILARITY INDEX **22%** INTERNET SOURCES **2%** PUBLICATIONS **18%** STUDENT PAPERS

PRIMARY SOURCES

1	newlightit.com Internet Source	3%
2	www.guru99.com Internet Source	3%
3	www.javatpoint.com Internet Source	2%
4	Submitted to University of Hertfordshire Student Paper	2%
5	blog.udemy.com Internet Source	1%
6	www.tutorialspoint.com Internet Source	1%
7	Submitted to Sogang University Student Paper	1%
8	Submitted to Westcliff University Student Paper	1%
9	developerintest.wordpress.com Internet Source	1%

10	www.coursehero.com Internet Source	1%
11	www.ir.juit.ac.in:8080 Internet Source	1%
12	Submitted to Jaypee University of Information Technology Student Paper	1%
13	sonisoft.simplesite.com Internet Source	1%
14	Submitted to Harrisburg University of Science and Technology Student Paper	1%
15	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
16	Submitted to Higher Education Commission Pakistan Student Paper	1%
17	Submitted to Birla Institute of Technology Student Paper	1%
18	Submitted to University of Bolton Student Paper	1%
19	Submitted to Lithan Academy Pte Ltd Student Paper	<1%
20	docplayer.net	

	Internet Source	<1%
21	Submitted to Chicago State University Student Paper	<1%
22	javarevisited.blogspot.lu Internet Source	<1%
23	Submitted to American Intercontinental University Online Student Paper	<1%
24	Submitted to Middle East College of Information Technology Student Paper	<1%
25	www.bipinwebacademy.com Internet Source	<1%
26	www.cigniti.com Internet Source	<1%
27	www.slideshare.net Internet Source	<1%
28	Submitted to Campbellville University Student Paper	<1%
29	Submitted to KDU College Sdn Bhd Student Paper	<1%
30	intellipaas.com Internet Source	<1%

31	www.scribd.com Internet Source	<1%
32	testingtechnology.blogspot.com Internet Source	<1%
33	Adam Roman. "A Study Guide to the ISTQB® Foundation Level 2018 Syllabus", Springer Science and Business Media LLC, 2018 Publication	<1%
34	www.authorstream.com Internet Source	<1%
35	www.hotmit.com Internet Source	<1%
36	Submitted to Arab Open University Student Paper	<1%
37	fedetd.mis.nsysu.edu.tw Internet Source	<1%
38	www.notesale.co.uk Internet Source	<1%
39	Submitted to South University Student Paper	<1%
40	Submitted to Modi Apollo International Institute Student Paper	<1%
41	qualitytestified.blogspot.com Internet Source	<1%

42 Submitted to Roehampton University <1%
Student Paper

43 Zhimin Zhan. "Selenium WebDriver Recipes in C#", Springer Science and Business Media LLC, 2015 <1%
Publication

44 ebooks.lpude.in <1%
Internet Source

45 software-testing-imp.blogspot.com <1%
Internet Source

46 iugspace.iugaza.edu.ps <1%
Internet Source

47 Submitted to Harare Institute of Technology <1%
Student Paper

48 practice.geeksforgeeks.org <1%
Internet Source

49 theses.whiterose.ac.uk <1%
Internet Source

Exclude quizzes On

Exclude matches Off

Exclude bibliography On