

**“A QUALITATIVE APPROACH TO THE QUALITY
ENGINEERING ASSESSMENT BASICS”**

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

BACHELOR OF TECHNOLOGY

IN

**COMPUTER SCIENCE AND ENGINEERING
&
INFORMATION TECHNOLOGY**

By

ANANYA BAJPAI (171221)

Under the Guidance of

(Cognizant)



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TABLE OF CONTENTS

CAPTION NO.	PAGE
CHAPTER-1: INTRODUCTION	
1.1 General Introduction	1
1.2 Organization	2
CHAPTER-2: SOFTWARE TESTING	
2.1 Software Development Lifecycle	3
2.2 Software Testing	8
2.3 Test Process	9
2.4 Test Levels	9
2.5 Test Case Writing	10
CHAPTER-3: INTRODUCTION TO MYSQL	
3.1 What is MySQL?	11
3.2 What is RDBMS?	12
3.3 Installation of MySQL	12
3.4 Data Definition Language	14
3.5 Data Manipulation Language	17
3.6 About Joins	17
3.7 Sub queries	20
3.8 Normalization	21
CHAPTER-4: BASICS OF JAVA LANGUAGE	
4.1 Introduction	24
4.2 OOPS	25
4.3 Interfaces in JAVA	27
4.4 Exception Handling	28

4.5 Strings	30
4.6 Arrays in JAVA	33
4.7 Collections in JAVA	35
4.8 JAVA Database Connectivity with MySQL	36
CHAPTER-5: SELENIUM WEBDRIVER	
5.1 Introduction to Selenium	39
5.2 Components of Selenium	39
CHAPTER-6: CONCLUSION and FUTURE SCOPE of WORK	41
REFERENCES	42

Project Report Undertaking

I Ananya Bajpai Roll No. 171221 Branch CSE is doing my internship with Cognizant from March 2021 to June 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.



Signature _____

Name Ananya Bajpai

Roll No. 171221

Date 22-05-2021

DECLARATION

I hereby declare that the work reported in this report entitled “**A QUALITATIVE APPROACH TO THE QUALITY ENGINEERING ASSESSMENT BASICS**” 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 and Engineering and Information Technology, Jaypee University of Information Technology Wanknaghat is an authentic record of my own work carried out over a period from March 2021 to June 2021 under the supervision of Cognizant.

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

A handwritten signature in blue ink, reading "Ananya", is written over a faint, circular watermark or stamp that is partially visible in the background.

Ananya Bajpai (171221)

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

Cognizant

Date: May 22.2021

ACKNOWLEDGEMENT

I would like to express our special thanks of gratitude to my mentor of Cognizant who gave us the golden opportunity to do this project on the topic, **A QUALITATIVE APPROACH TO THE QUALITY ENGINEERING ASSESSMENT BASICS**” which also helped us in doing a lot of Research and we came to know about so many new things.

I am really thankful to him.

Secondly, we would also like to thank Lab assistants who helped us a lot in finalizing this project within the limited time frame.

ANANYA BAJPAI (171221)

LIST OF FIGURES

Figure	Page
2.1 Graphical representation	3
2.2 Mounts in a SDLC	4
2.3 Types of SDLC	4
2.4 Stages in waterfall model	5
2.5 Steps Involved	6
2.6 Incremental Model	6
2.7 Iterative Model	7
2.8 Scrum	7
2.9 Types	8
2.10 Steps	8
2.11 Relationship between Testing and Debugging	8
2.12 Test Process	9
2.13 Test Levels	9
3.1 MySQL Database Administrators	11
3.2 Fundamentals of Relational Databases	12
3.3 Steps	12
3.4 Creating a Table with primary key	15
3.5 Creating a Table	16
3.6 Adding and removing column	16
3.7 Inserting values	17
3.8 Inner Join	18
3.9 Query of Inner Join	18

3.10 Left join	18
3.11 Query of Left Join	19
3.12 Right Join	19
3.13 Query of Right Join	20
4.1 How it works	23
4.2 Data Types in Java	24
4.3 Types of Interference in Java	25
4.4 Exception Hierarchy	27
4.5 Example of exception handling	29
4.6 Example with String methods	31
4.7 Important methods of StringBuffer class	32
4.8 Example of StringBuffer	32
4.9 Array Example in Eclipse	33
4.10 Chain of command of Collection Framework	34
4.11 Example of ArrayList	35
4.12 Java Database Connectivity with MySql	36
4.13 Code in Eclipse	36
5.1 Pros and Cons	38

ABSTRACT

My five month internship project, A QUALITATIVE APPROACH TO THE QUALITY ENGINEERING ASSESSMENT BASICS, was with Cognizant Pune Office starting from the March 2021. The report is put together to throw some light on the concepts learned and its implementations.

The main motive of the internship was to learn software testing and using Selenium Automation Tool. Selenium is the single interface that allows you to compose test scripts in various languages like Ruby, Java, PHP, C# and some others and hence is advantageous over other automation tools. I will be learning Selenium using Java language and furthermore implementing web automation.

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Today, information is the premise of any business. The universe of big business figuring is the embodiment of information driven organizations. The significance of organized stockpiling of information is verifiably in the core interest. What's more, right now, with the figuring model rapidly moving towards automation .Automation of websites is very important.

An automated site perceives the interesting requirements of guests and upgrades content for every watcher. The product assists with aiding support contacts all the more productively. It is responsive (versatile agreeable) out-of-the-case, completely incorporated with each showcasing channel, and versatile to the changing necessities of leads and clients.

Java is characterized by a determination and comprises of a programming language, a compiler, center libraries and a runtime (Java virtual machine) The Java runtime permits programming engineers to compose program code in different dialects than the Java programming language which despite everything runs on the Java viirtual machine. The Java stage is typically connected with the Java virtual machine and the Java center libraries.

Similarly SQL has the capacity to perceive the various sorts of information bases. It has the capacity to interface with the information base utilizing distinctive SQL association customers. Comprehension of the connection between data set tables, keys, and records. Capacity to compose a basic select or SQL articulation alongside more intricate join inquiries

Selenium is an open source tool that automates web browsers. Selenium is Open Source UI Automation Tool. It is used to Automate the Web-Application but not for Desk-Top Base Application Automation. It was created in 2004 by a ThoughtWorks Engineer. Selenium Web-driver was introduced later in 2006.

1.2 Organization

Cognizant Technology Solutions is a worldwide IT administrations organization headquartered in New Jersey (US).

It began as a specialist organization to Dun and Bradstreet organizations worldwide in 1994. Back then, the organization was situated in India and was called Dun and Bradstreet Satyam Software (DBSS).

In 1998, it became autonomous when D&B split into three with one gathering of organizations going under Cognizant Corporation. By at that point, DBSS had purchased over Satyam Software's stake and gotten totally free.

Known as CTS in India or the brand Cognizant all inclusive, the organization began giving IT Services and Solutions to clients worldwide with the base camp at NJ. Mindful is recorded on the NASDAQ stock trade in the US under CTSH.

From that point forward, the organization has developed a long ways in the Industry!

Cognizant Technology Solutions Corporation (Cognizant) is a supplier of custom data innovation (IT) confab and innovation administrations, just as redistributing administrations for Global 2000 organizations situated in North America, Europe and Asia. The Company's center administrations incorporate innovation methodology counseling; complex frameworks advancement; venture programming bundle execution and upkeep; information warehousing and business knowledge; application testing; application support; foundation the board, and vertically-situated business process re- appropriating (V-BPO). Aware tailor its administrations to explicit ventures, and uses a coordinated nearby/seaward plan of action. The Company works in four business portions: Financial Services, Healthcare, Manufacturing/Retail/Logistics and Other.

CHAPTER 2

SOFTWARE TESTING

2.1 Software Development Lifecycle

A Software Development Lifecycle Model tells us about the types of activities executed at every stage in a software development project, and how the ventures relate to one another aptly and chronologically.

- SDLC is the elision of Software Development Life Cycle.
- It is furthermore called as Software Development Process.
- SDLC is a complex set apart assignments executed at each parade in the product up gradation measure.
- ISO/IEC 12207 is a global typical for programming wheel of life measures. It is said to be the streamer that sets every one of the assignments needed for fabricating and looking after programming.



Fig 2.1 Graphical representation of the numerous junctures of a classic SDLC

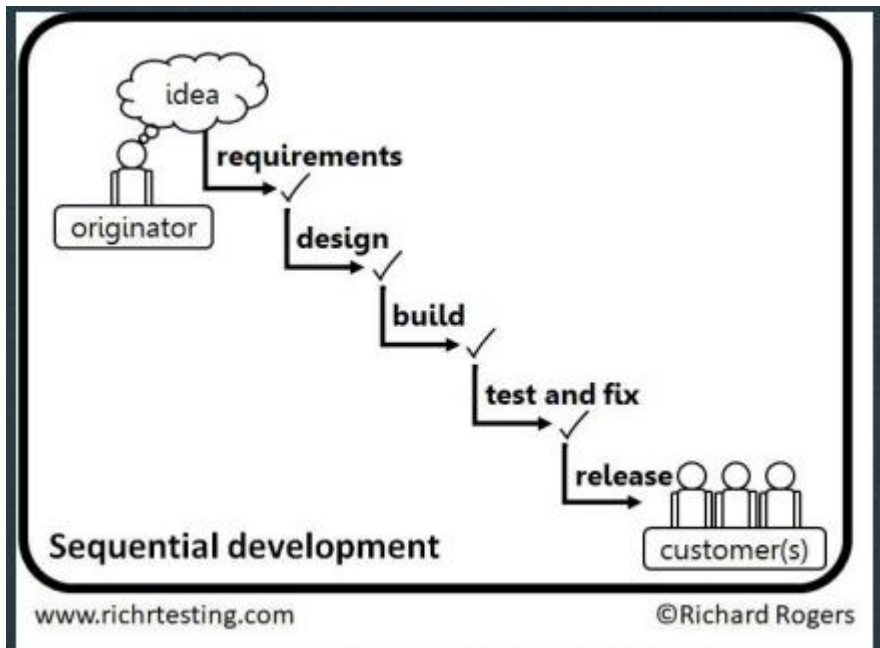


Fig 2.2 Mounts in a SDLC

2.1.1 Types of SDLC

1. Sequential
2. Iterative and Incremental

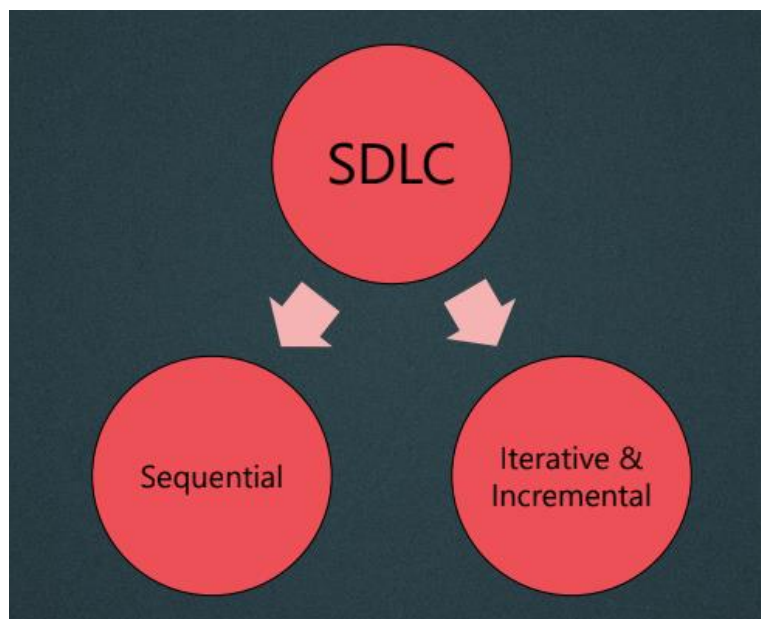


Fig 2.3 Types of SDLC

2.1.2 Sequential Software Development

A sequential development model expresses the software development process as a continuous, progressive flow of activities. This implies that any stage in the advancement, interaction should start when the past stage is finished. In principle, there is as such no cover of stages, yet by and by, it is useful to have early input from the accompanying stage.

Advantages

- Cost Effective
- Less time taking
- Suitable for different geographical positions
- It's linear
- Maximized customer satisfaction
- No pre knowledge required

Disadvantages

- Rigid
- Absence of central authority
- Lack of intuitiveness
- No centrality of the client

Waterfall Model

In the Waterfall model, the maturing ventures are completed one after another. In this model, test ventures only occur after all other maturing activities have finally been out righted. Its diagrammatic portrayal looks like a course of cascades.



Fig 2.4 Stages in waterfall model

V-Model

Disparate the Waterfall model, the V-model mingles the test operations all round the development process, executing the principle of before time testing.

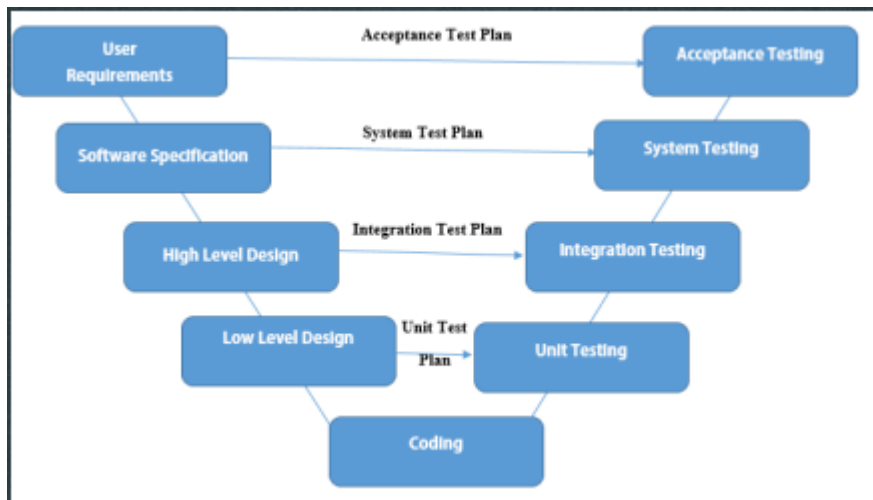


Fig 2.5 Steps Involved

2.1.3 Incremental and Iterative Development

Incremental Model is an interaction of programming advancement where prerequisites isolated into numerous independent modules of the product improvement cycle. In this replica, every single module goes through the necessities, intent, execution, and trial stages. Each ensuing advent of the segment adds magnitude to the earlier discharge. The interaction gets going until the entire skeleton attained.

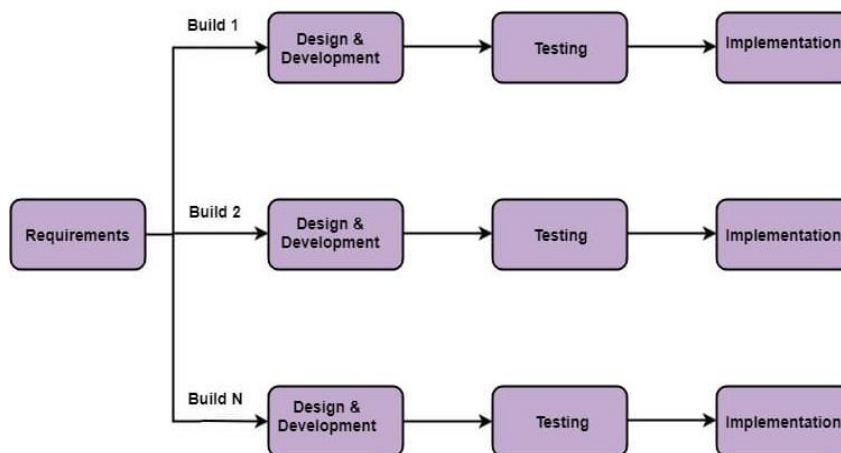


Fig 2.6 Incremental Model

In Iterative Model, you can begin with a portion of the product determinations and build up the main adaptation of the product. After the main form on the off chance that there is a need to change the product, another rendition of the product is made with another emphasis. Each arrival of the Iterative Model will result in a definite and fixed period that is called cycle.

This replica sanctions the coming by to antecedent stages, in which the diversities made discretely. The rearmost yield of the venture recharged towards the varnish of the Software Development Life Cycle measure.

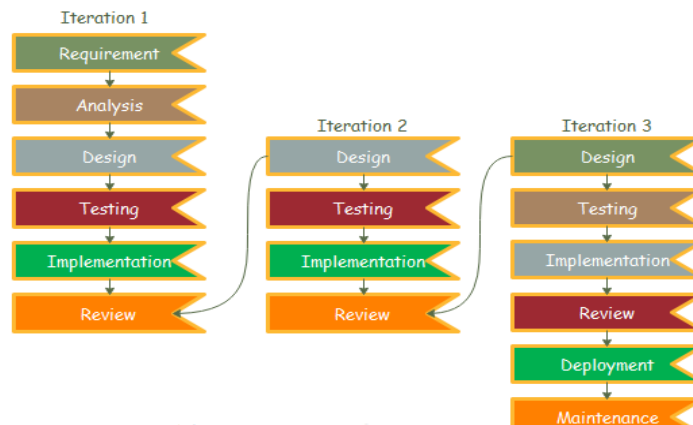


Fig 2.7 Iterative Model

Scrum

Each replay assumes to be rather small (e.g., hours, days, or a few weeks), and the trait redoes are correspondingly trivial, such as one or two intensifications and/or hardly some new features.

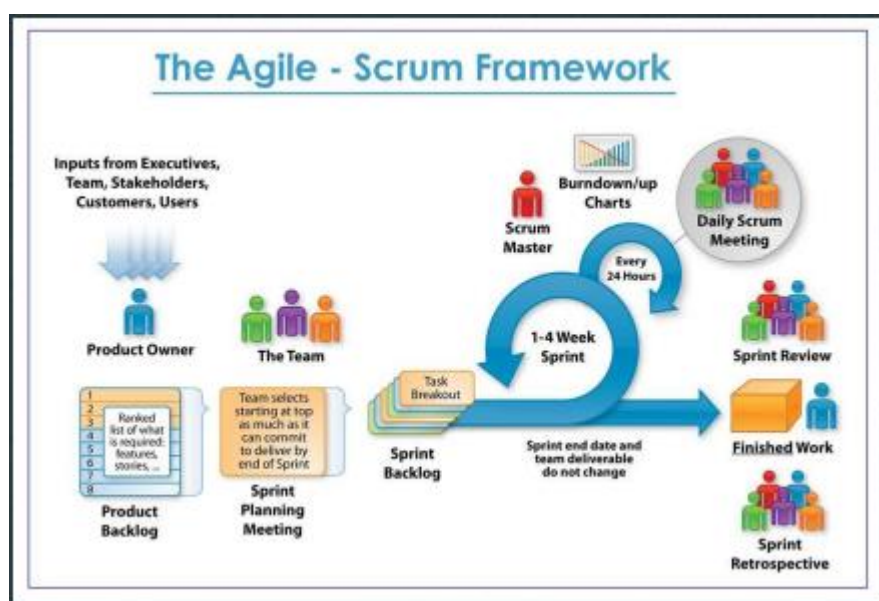


Fig 2.8 Scrum

2.2 Software Testing

Software testing can be called as an affair which comprises several distinct activities or jobs. Of which execution is only one of these jobs

Software testing does the following:

- gauges the quality of the software
- risk of software failure in operation is reduced

Software Testing \neq Test Execution

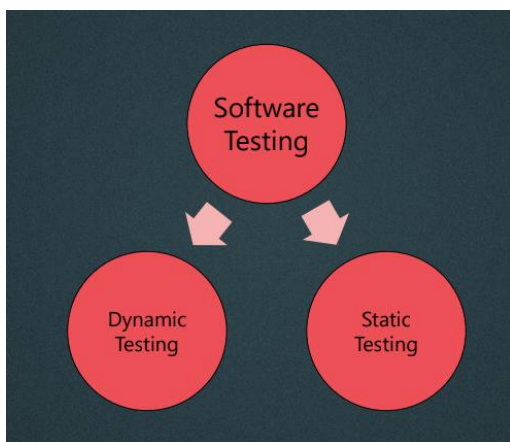


Fig 2.9Types

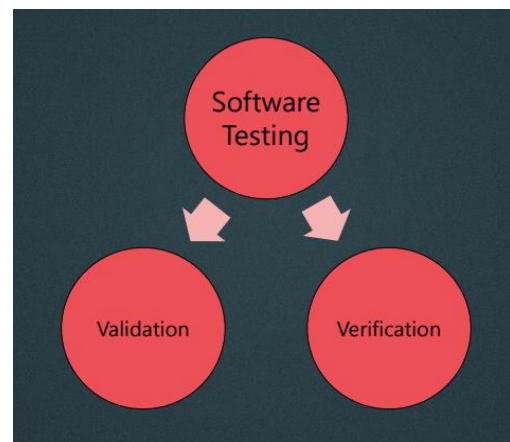


Fig 2.10 Steps

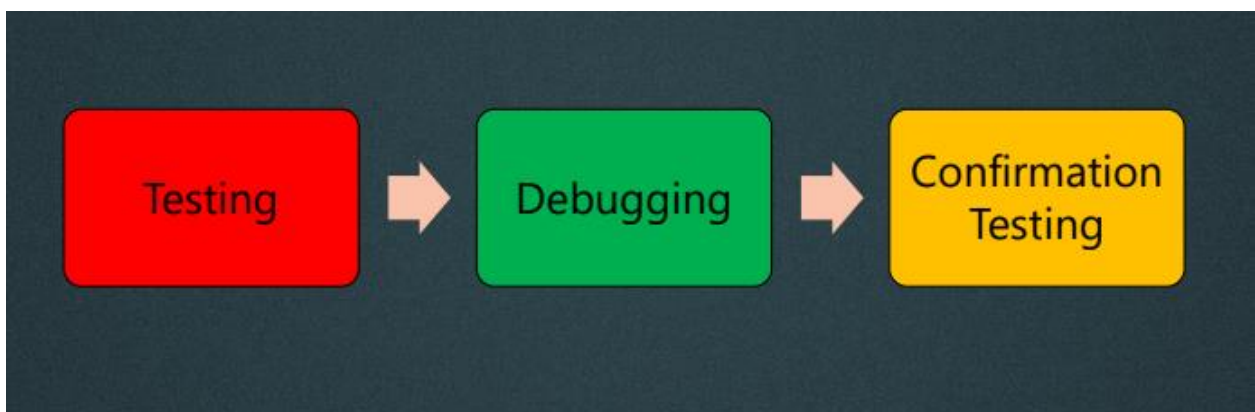


Fig 2.11 Relationship between Testing and Debugging

2.3 Test Process

There is nobody all inclusive programming test measure, yet there are basic arrangements of test exercises without which testing will be more averse to accomplish its set up destinations.



Fig 2.12 Test Process

2.4 Test Levels

Test levels are gatherings of test exercises that are coordinated and overseen together. Each test level is an occasion of the test interaction. Test levels are identified with different exercises inside the product improvement lifecycle.

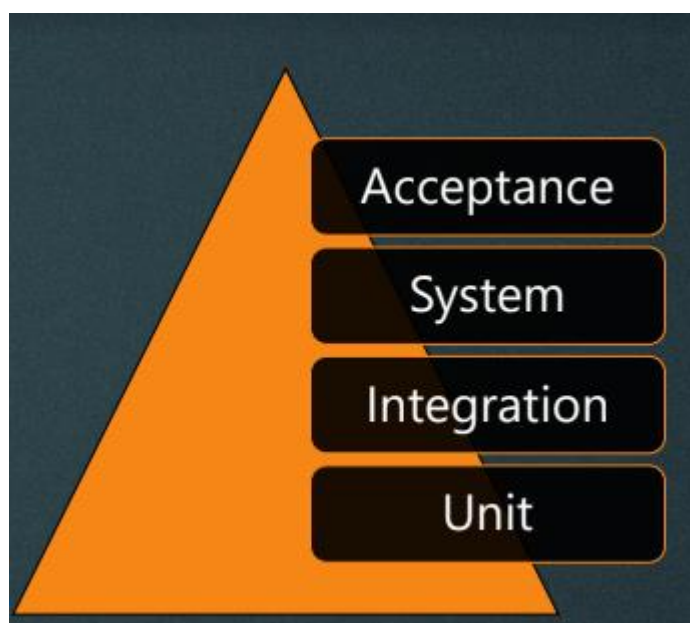


Fig 2.13 Test Levels

2.5 Test Case Writing

A bunch of preconditions, inputs, activities (where relevant), anticipated outcomes and post conditions, created dependent on test conditions.

It involves:

- Test Case Title
- Pre-Conditions
- Test Steps
- Expected Results
- Test Suite
- Test Environment
- Actual Results
- Status
- Defect Report

Defect Report - Documentation of the event, nature, and status of a deformity. Wherein a blemish or inadequacy in a work item where it doesn't meet its prerequisites or particulars is a deformity.

CHAPTER 3

Introduction to MySQL

SQL means Structured Query Language. It is a standard language used to have conversation with the databases.

SQL is an exceptional programming language that is utilized to confederate with databases. It works by grasping and breaking down information bases that incorporate information fields in their tables.

3.1 What is MySQL?

MySQL is a community database the board shell. It gives a UI to us to connect with the database. The application is employed for a vast scope of causes, counting details warehousing, online trading, and logging applications. The most familiar use for MySQL notwithstanding is with the final intent of a web information base.

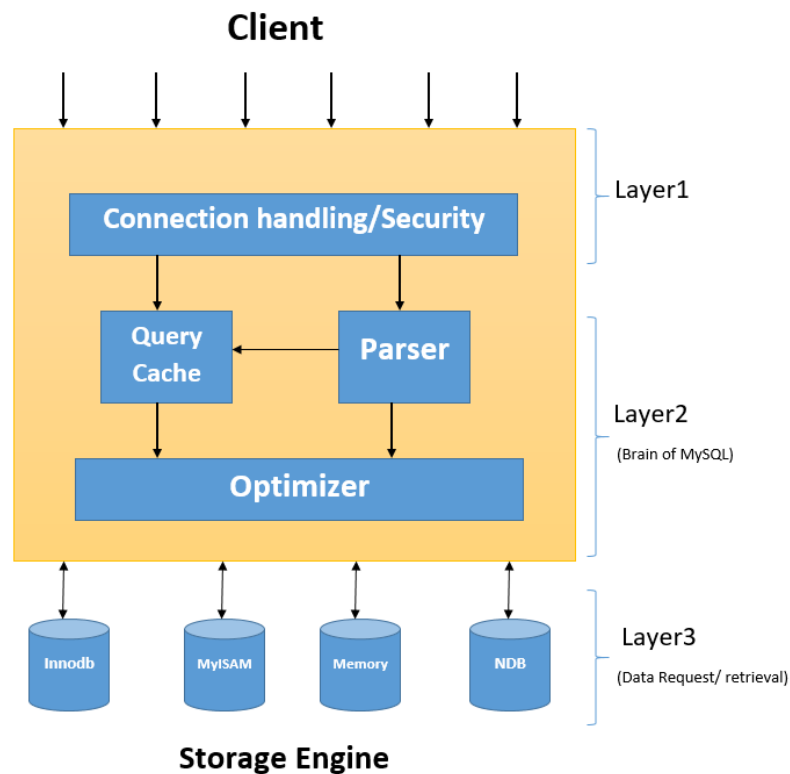


Fig 3.1 MySQL Database Administrators

3.2 What is RDBMS?

A relational data set touch on to a data set that reserves statistics in a well-structured group, availing lines and segments. Hence it is easier to find and ingress obvious traits under the details base. The social design makes it tenable to sprint doubts across unrelated tables immediately.

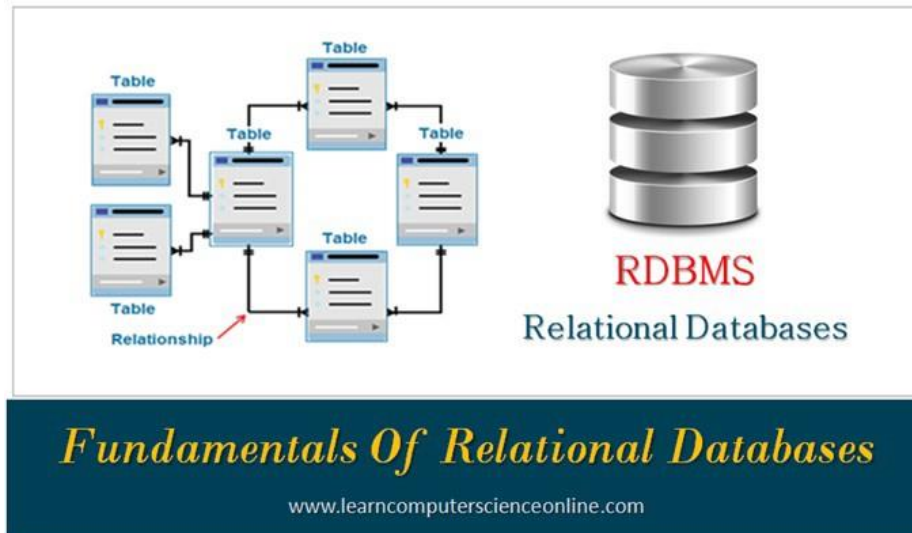


Fig 3.2 Fundamentals of Relational Databases

3.3 Installation of MySQL

Step1:

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

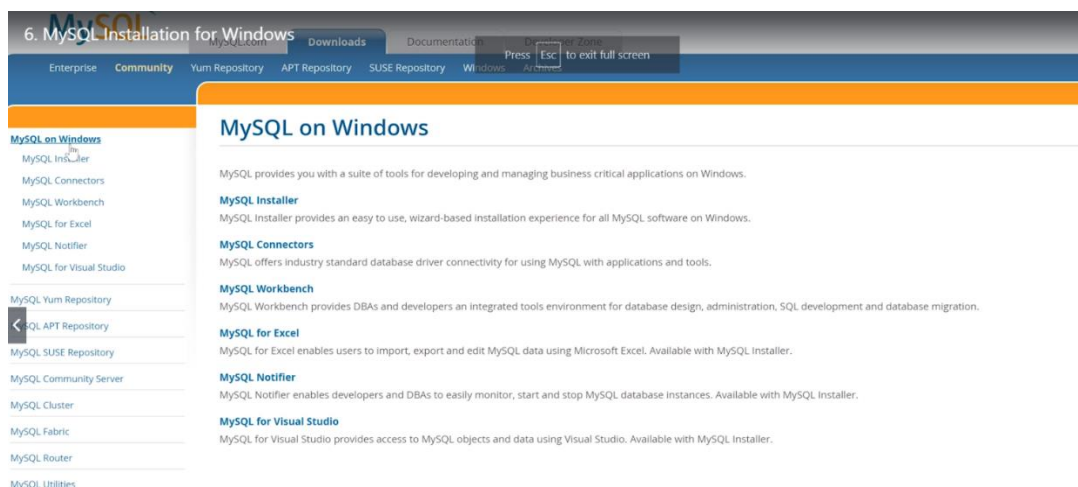


Fig 3.3 (a) Step 1

Step2:

Ensuing to acquainting the MySQL with your system, fire setting up the workbench stage where you can create the inquiries.

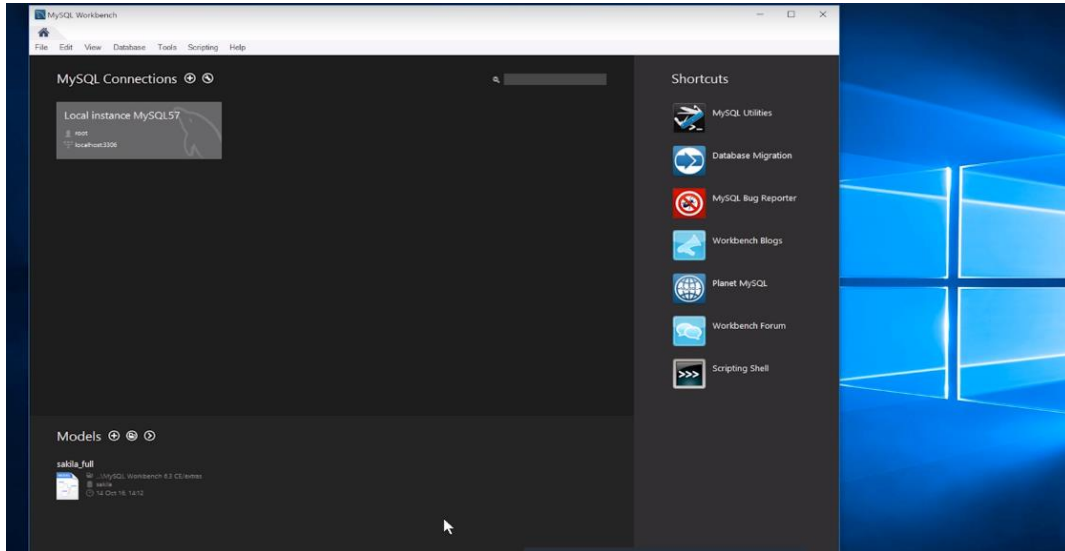


Fig 3.3 (b) Step 2 of Installing MySQL

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.

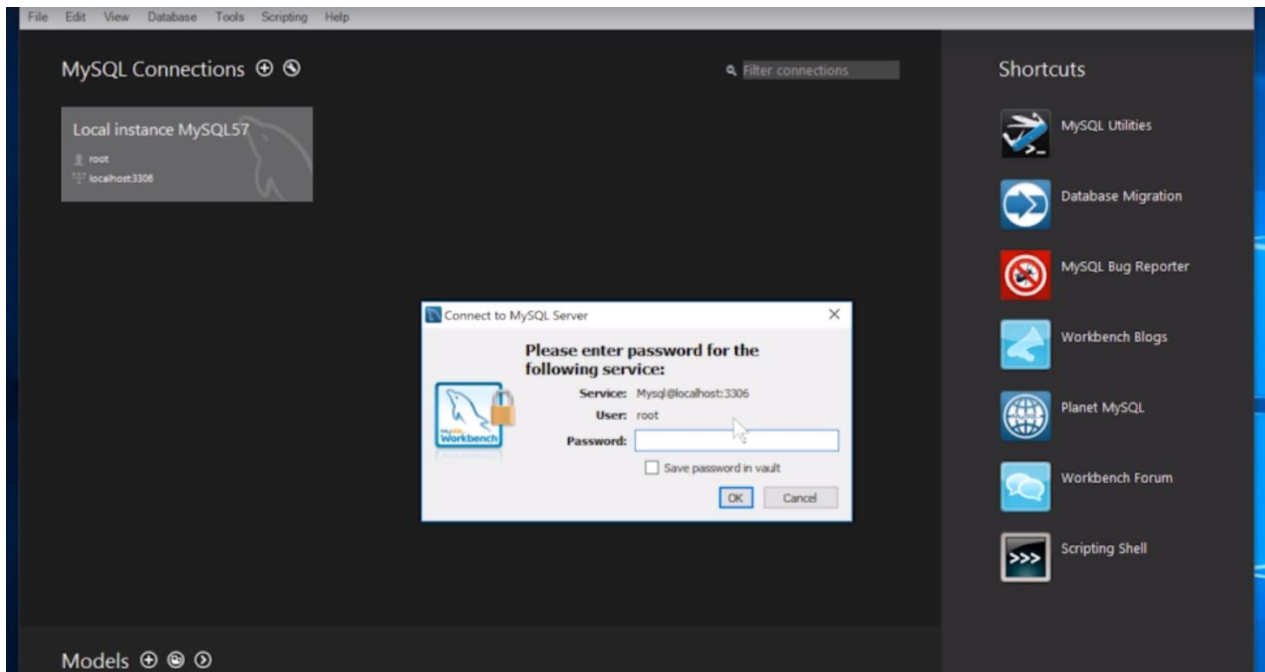


Fig 3.3 (c) Step 3 of installing MySQL

Step4:

Then comes the time when you can make your first database inside the workbench.

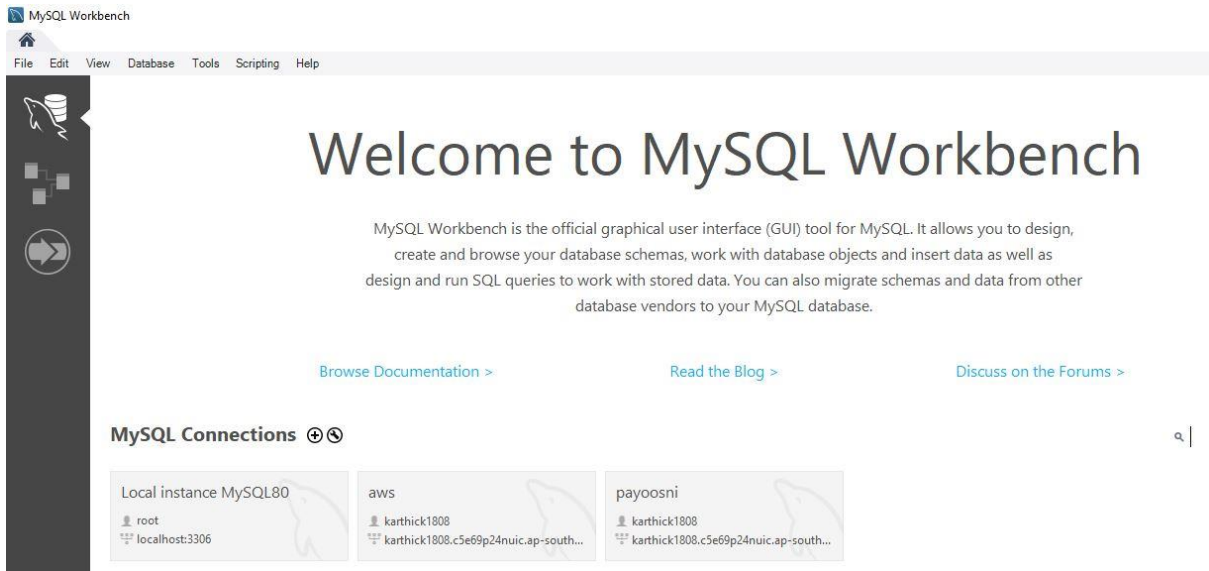


Fig 3.3 (d) Step 4 of Installing MySQL

3.4 Data Definition Language

Information definition language is a grammar for manufacturing and switching data set items like tables, lists, and clients. DDL manifestations are like a PC programming language for setting apart information frameworks, particularly data set blueprints. The common statements used are: CREATE, ALTER, DROP, TRUNCATE.

3.4.1 Data Types

- 1) Numeric Data type:
 - INT: Whole numbers
 - FLOAT(M,D): Decimal numbers(approx.)
 - DECIMAL(M,D): Decimal numbers(precise)
- 2) Non-Numeric Data type:
 - CHAR(N): Fixed length character
 - VARCHAR(N): Varying length character
 - ENUM('M','F'): Value from a characterized list

- BOOLEAN: True or False qualities
- 3) Date and Time type:
- DATE: Date(yyyy-mm-dd)
 - DATETIME: Date and time (yyyy-mm-dd hh-mm-ss)
 - TIME: Time(hh-mm-ss)
 - YEAR: Year(yyyy)

3.4.2 Primary and Foreign Keys

Primary Key: An essential key is a field in a table which exceptionally recognizes each column/record in a data set table. Essential keys should contain unusual caliber. An essential key segment is not allowed to have NULL qualities. A table has just one essential key, which may contain single or various fields.

Foreign Key: This imperative is utilized to forestall activities that would annihilate joins between tables.

```

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 );

```

Messages

Table test_results created
Execution time: 0.11s

Fig 3.4 Creating a Table with primary key

3.4.3 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>".

Furthermore, to eradicate all the data from inside the tables abbreviate request is used.

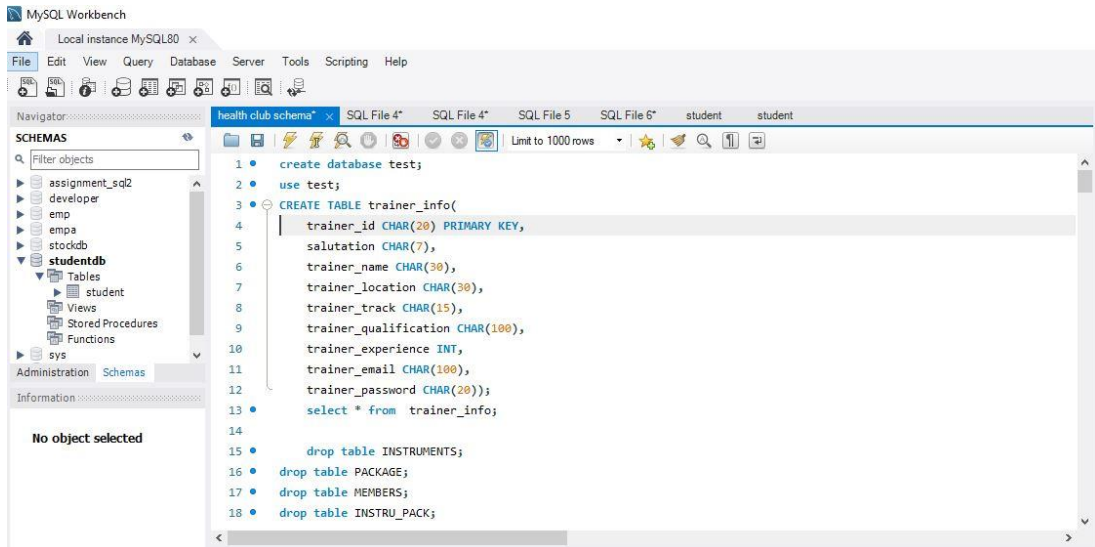


Fig 3.5 Creating a Table

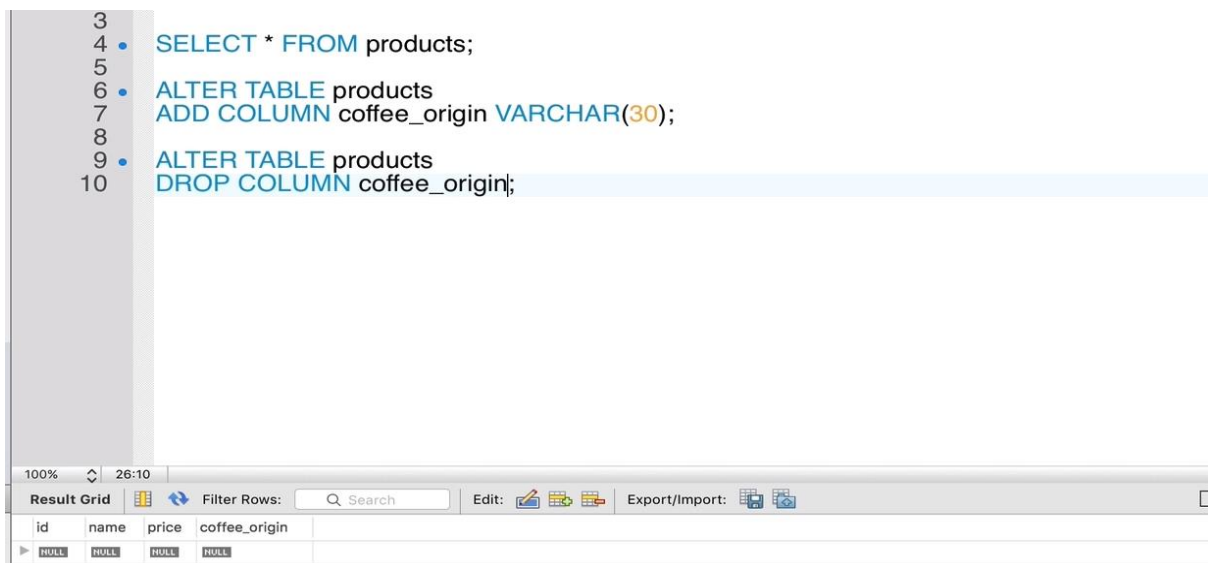


Fig 3.6 Adding and removing column

3.5 Data Manipulation Language

A DML is customarily a sublanguage of a more sizeable data set language like SQL, with the DML containing a chunk of the controllers in the language. Read-only choosing of data is some of the time accepted as being vital for a different DQL, however it is steadily allied and at times likewise thought to be a part of a DML.. The common commands under DML are: INSERT, SELECT, UPDATE, DELETE.

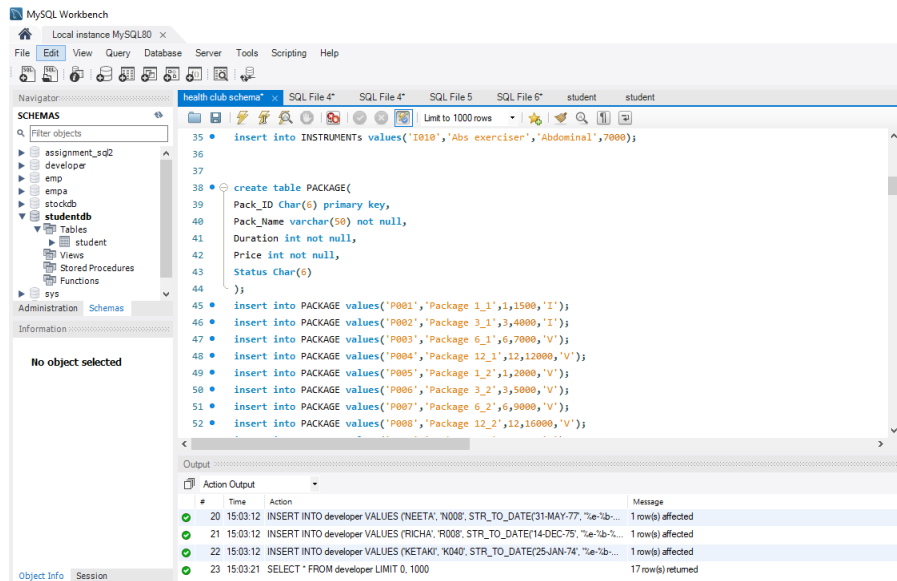


Fig 3.7 Inserting values

3.6 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.

3.6.1 Inner Join

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

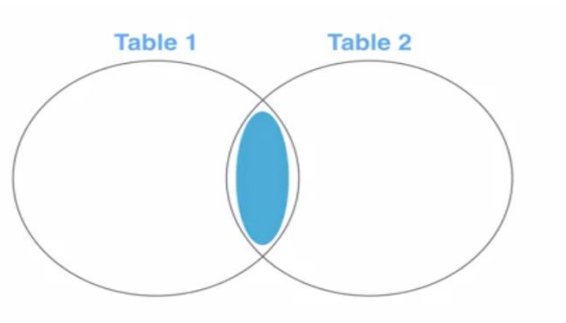


Fig 3.8 Inner Join

```

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

```

Fig 3.9 Query of Inner Join

3.6.2 Left Join

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

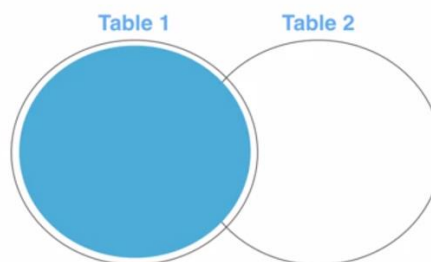


Fig 3.10 Left join

```
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 orders o
17 LEFT JOIN customers c ON o.customer_id = c.id
18 ORDER BY o.order_time
19 LIMIT 10;
```

Fig 3.11 Query of Left Join

3.6.3 Right Join

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

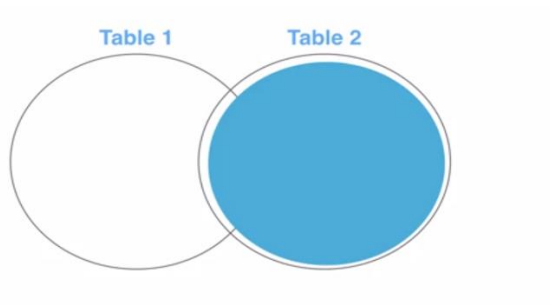


Fig 3.12 Right Join

```
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 customers c
17 RIGHT JOIN orders o ON c.id = o.customer_id
18 ORDER BY o.order_time
19 LIMIT 10;
20
```

Fig 3.13 Query of Right Join

3.7 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.

2.7.1 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.

2.7.2 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.

3.8 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 extra room
- Reduce addition, update, and cancellation inconsistencies
- Improve inquiry execution
- Reduce the extra room
- Reduce addition, update, and cancellation inconsistencies
- Improve inquiry execution

Following are the degrees of standardization:

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

3.8.1 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 solitary worth
- The table has an essential key

3.8.2 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

3.8.3 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.

CHAPTER 4

JAVA

4.1 Introduction

Java is an extensively valuable programming language that is class-based, object-arranged, and planned to have as very few execution conditions as could sensibly be anticipated. It is relied upon to let application engineers make once, run wherever (WORA), inferring that arranged Java code can run on all stages that help Java without the necessity for recompilation. Beginning at 2019, Java was quite possibly the most notable programming vernaculars being utilized by GitHub, particularly for client worker web applications.

Real world applications

There are numerous spots where Java is utilized in reality, beginning from an online trading 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.

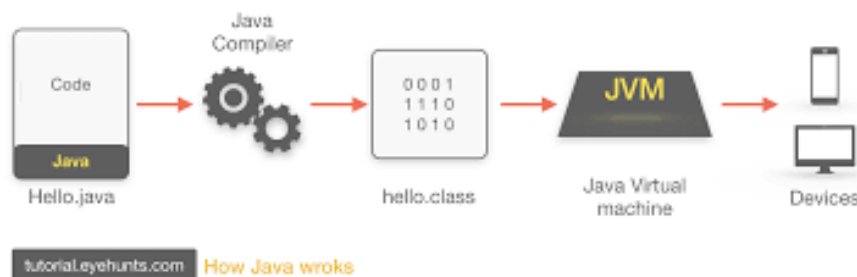


Fig 4.1 How it works

4.1.1 Java virtual Machine

Java Virtual Machine (JVM) is a motor that gives runtime states to push the Java Code or applications. It switches over Java byte code into machines language. JVM is a masterpiece of Java

Run Environment (JRE). In other programming dialects, the compiler will produce machine code for specific structures. In any case, Java compiler produces code for a Virtual Machine

4.1.2 Data types in java:

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

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

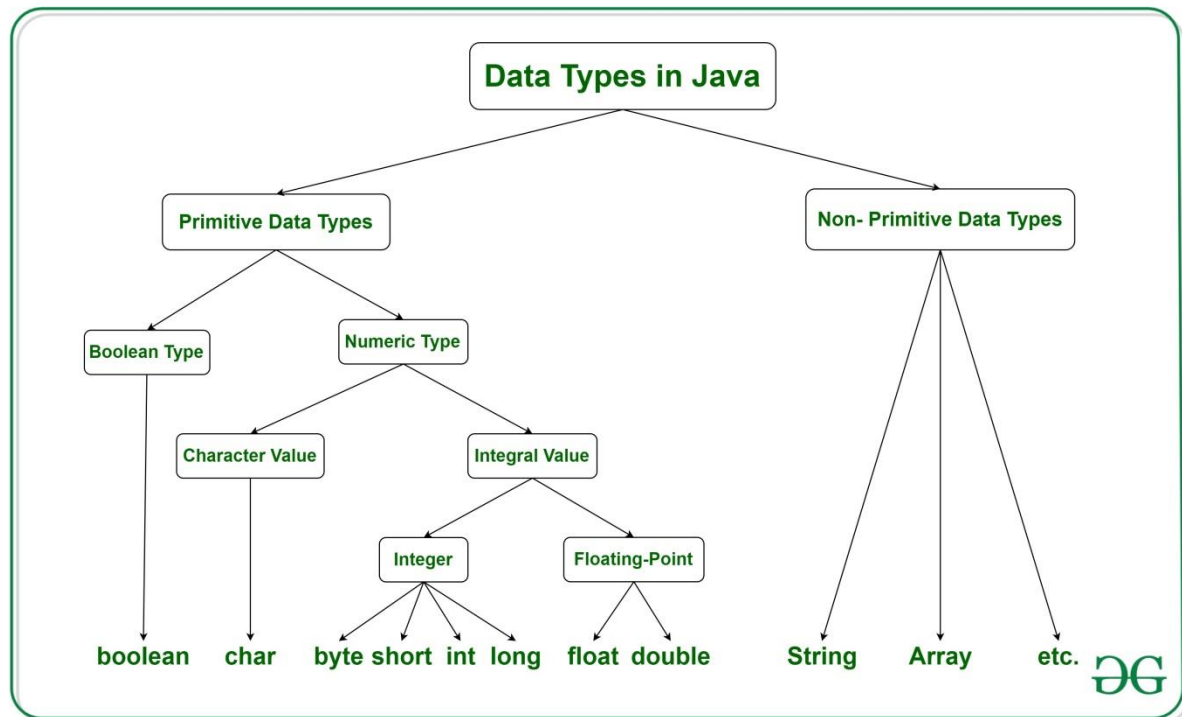


Fig 4.2 Data Types in Java

4.2 What is OOPS?

Object-oriented programming System (OOPS) is a programming perspective ward on "objects" that contain data and procedures. The essential part of thing masterminded composing PC programs is to construct the flexibility and common sense of activities. It joins data and its conduct (techniques) in a singular area (object) makes it all the more clear how a program works.

Concepts of OOPS:

- **Class:** It is a conference of near components. For instance, you had a class called "Costly Cars". Its properties (data) can be cost or speed of these vehicles. While the systems may be performed with these vehicles are driving, easing back down, etc.
- **Objects:** An object can be designated as a development of a class, and there can be different adaptations of a class in a program. For example - seat, bike, marker, pen, table, vehicle, etc.
- **Inheritance:** Classes can share, acquire or "acquire" properties and techniques that have a place with existing classes. This allows you to reuse existing code and reduces the time you spend coding. A class that gets from another is known as a derived class or a young person class. A class that shares its properties and methodologies is known as a base class or parent class.

The different types of inheritance in java are:

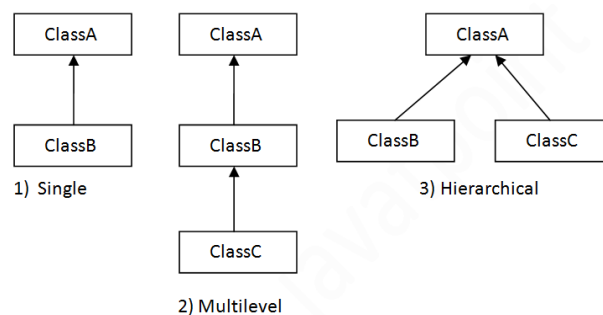


Fig 4.3 Types of Interference in Java

Note: Java does not support multiple inheritances.

- **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.

4.3 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, no body).

- 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. }
```

4.4 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:

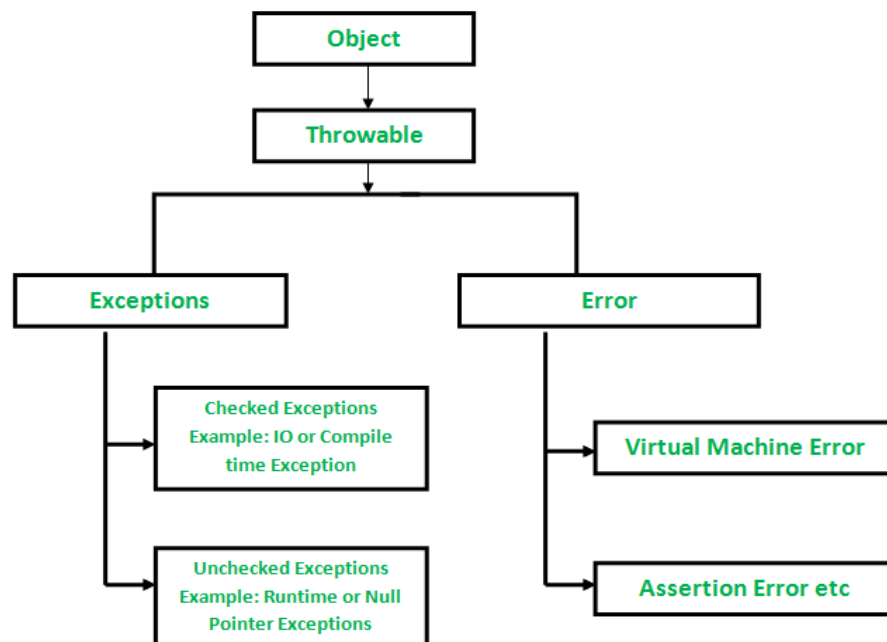


Fig 4.4 Exception Hierarchy

Types of Exceptions:

- Built-in Exceptions
- User Defined Exceptions

4.4.1 Java Exception Keywords:

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 special case in the technique. It is constantly utilized with strategy signature.

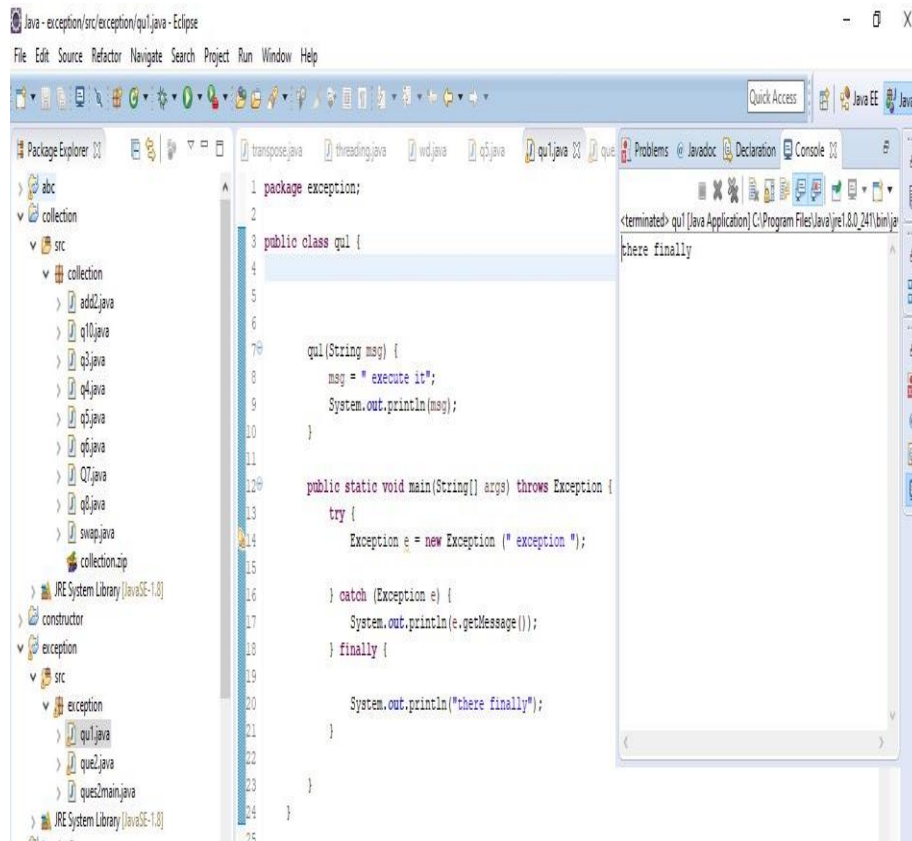


Fig 4.5 Example of exception handling

4.5 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 R = "ANANYA "

Using new keyword

String R = new String ("ANANYA");

4.5.1 String Methods:

1. `char charAt(int list)` - Restores the character at the predefined list.
2. `int compareTo(Object o)` - Examines this String to another Object.
3. `int compareTo(String anotherString)` -Examines two strings lexicographically.
4. `int compareToIgnoreCase(String str)` - Examines two strings lexicographically, disregarding case contrasts.
5. `String concat(String str)` - Connects the predefined string as far as possible of this string.
6. `boolean equals(Object anObject)` -Looks at this string to the predetermined item.
7. `int indexOf(int ch)` -Returns the list inside this string of the principal event of the predetermined character.
8. `int indexOf(String str)` -Restores the file inside this string of the primary event of the predefined substring.
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 predefined record.
10. `int lastIndexOf(String str)` -Restores the file inside this string of the furthest right event of the predetermined substring.

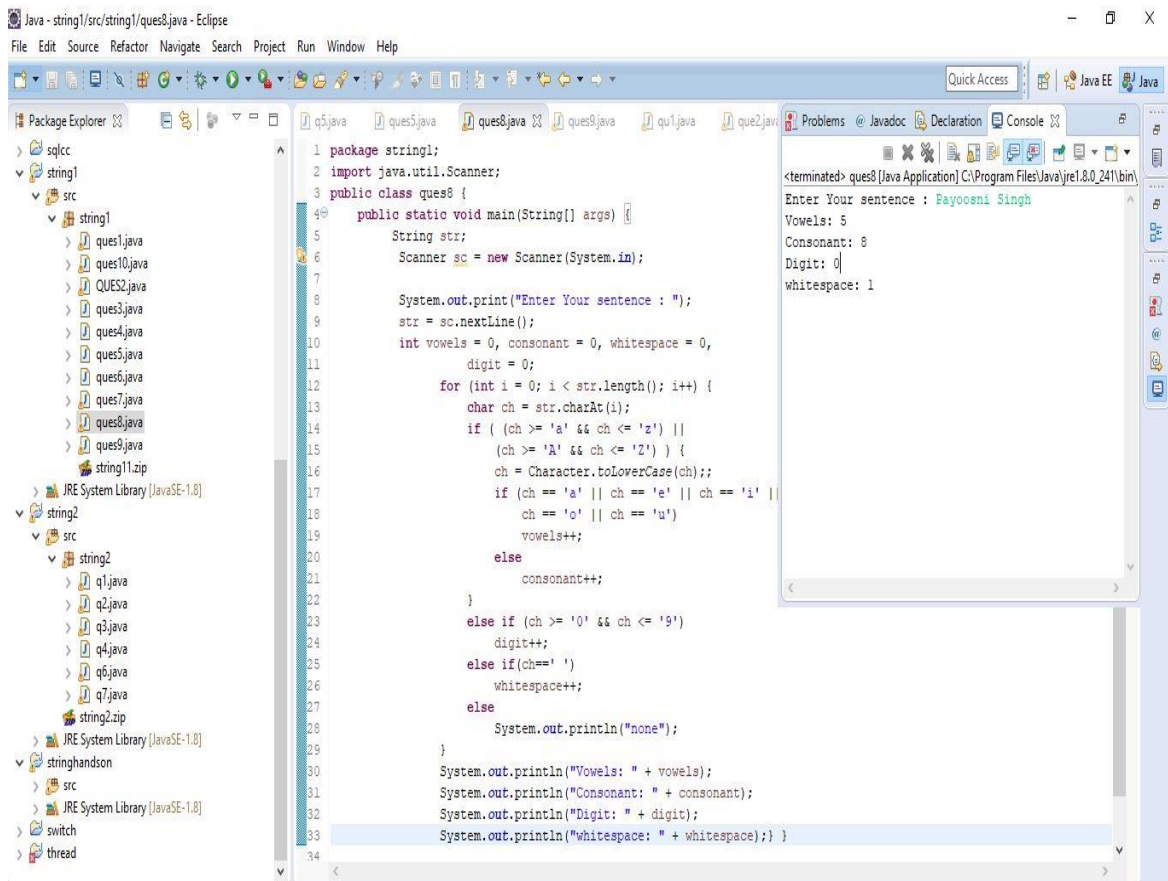


Fig 4.6 Example with String methods

4.5.2 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.	<code>StringBuffer()</code>	makes an unfilled string support with the underlying limit of 16.
2.	<code>StringBuffer(String str)</code>	creates a string support with the predetermined string.
3.	<code>StringBuffer(int capacity)</code>	creates an unfilled string support with the predefined limit as length.

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

Fig 4.7 Important methods of StringBuffer class

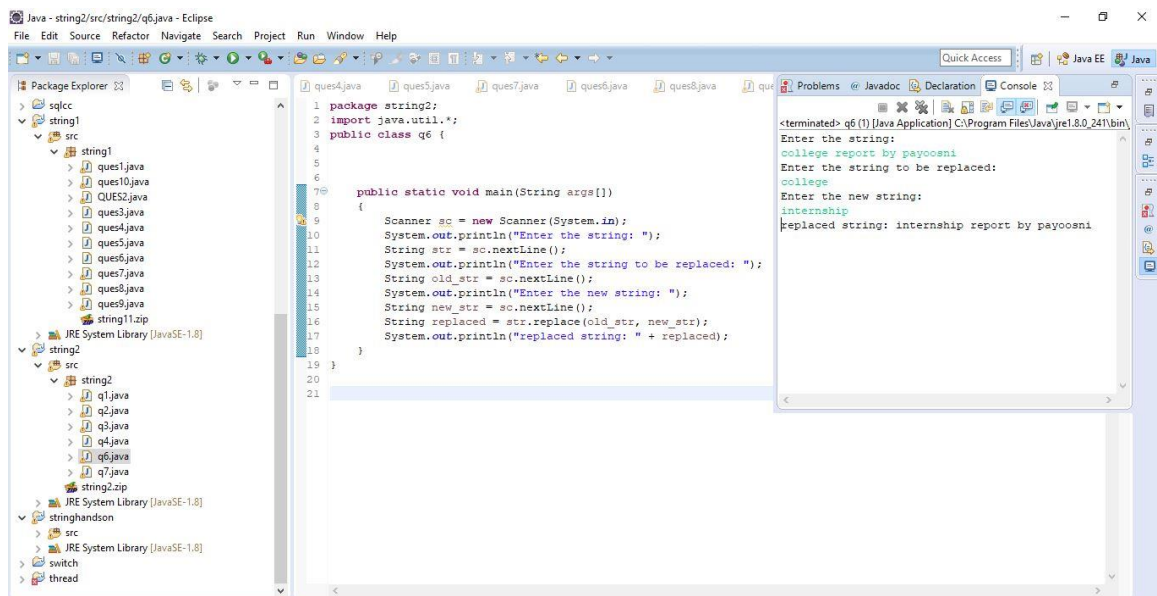


Fig 4.8: Example of StringBuffer

4.6 Arrays in Java

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]);
```

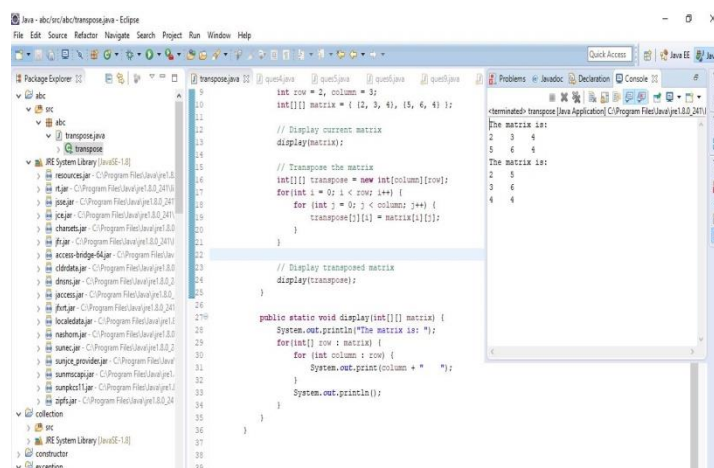


Fig 4.9 Array Example in Eclipse

4.7 Collections in Java

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

Java Collections can achieve all the exercises that you perform on a 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).

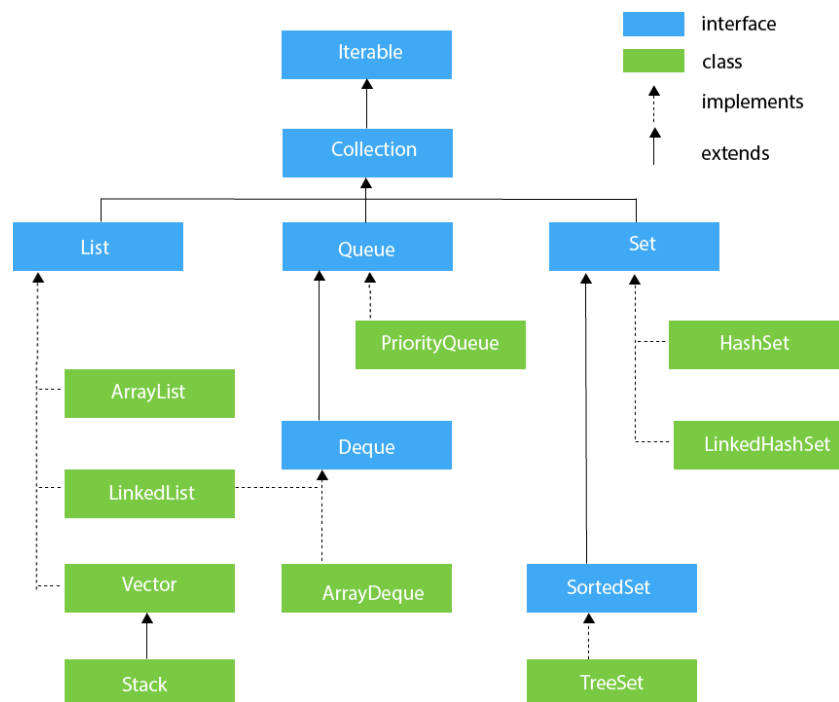


Fig 4.10 Chain of command of Collection Framework.

4.7.1 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.

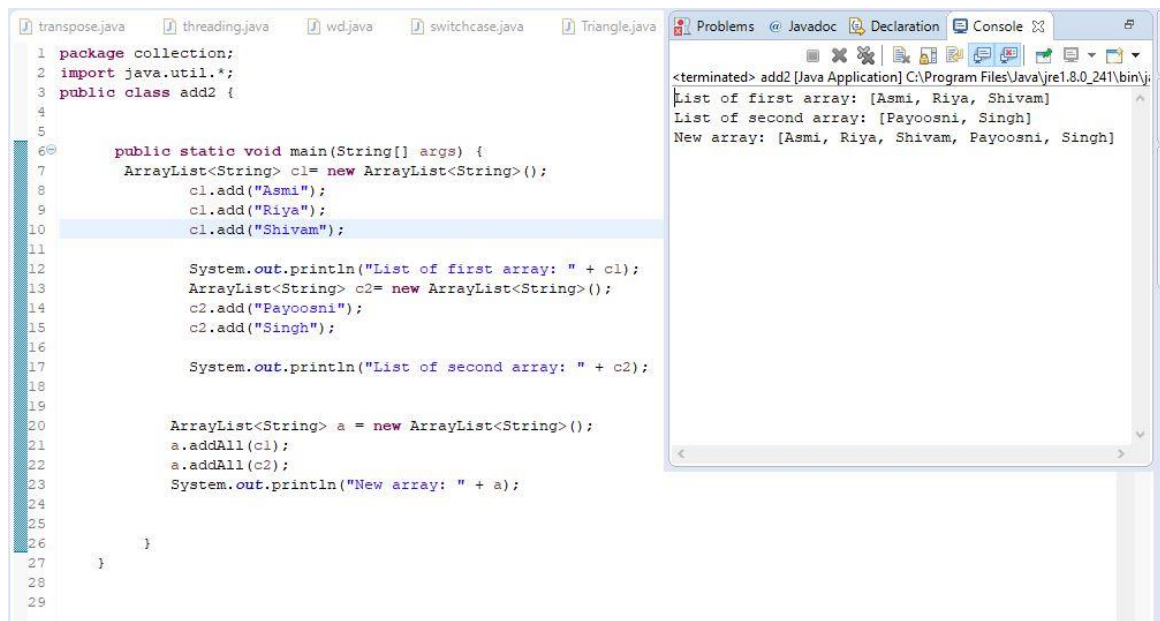


Fig 4.11 Example of ArrayList

4.8 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 is `root`.
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 secret key.

```
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    private static ConnectionFactory connectionFactory = null;
13
14    private ConnectionFactory()
15    {
16        try {
17            Class.forName(driverClassName);
18        } catch (ClassNotFoundException e) {
19            e.printStackTrace();
20        }
21    }
22
23    public Connection getConnection() throws SQLException {
24        Connection conn = null;
25        conn = DriverManager.getConnection(connectionUrl, dbUser, dbPwd);
26        return conn;
27    }
28
29    public static ConnectionFactory getInstance() {
30        if (connectionFactory == null) {
31            connectionFactory = new ConnectionFactory();
32        }
33        return connectionFactory;
34    }
35
36 }
```

Fig 4.12 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 }
```

Fig 4.13 Code in Eclipse

CHAPTER 5

SELENIUM WEBDRIVER

5.1 INTRODUCTION TO SELENIUM

Selenium is Open Source UI Automation Tool. It is used to Automate the Web-Application but not for Desk-Top Base Application Automation. It was created in 2004 by a ThoughtWorks Engineer. Selenium Web-driver was introduced later in 2006.

5.2 Components of Selenium

Selenium IDE

- Selenium IDE is Record and Play base Tool.
- Selenium IDE is basically a Browser Plugin, which is compatible with Mozilla Only.
- It's easy to learn, use and install.
- User is not allowed to put conditions in IDE.

Selenium RC

- Selenium RC stands for Remote Control
- RC is the first automated web testing tool that allowed users to use a programming language they prefer
- User is allowed to use Java, C#, PHP, Perl, Ruby, and Python with Selenium RC.

Selenium RC	
Pros	Cons
Cross Browser and Platform	More Complicated the IDE
Perform Loops and Conditional Statements	Programming Knowledge is Must
Data Driven Testing is Possible	Selenium RC Server needs to be run to start the execution
Can Support Multiple Browsers and faster then IDE	Slow then WebDriver and Poor Support to Java Script

Fig 5.1 Pros and Cons

Selenium WebDriver

- Selenium WebDriver is better than Selenium IDE and RC
- Selenium WebDriver is more stable than RC
- WebDriver, unlike Selenium RC, does not rely on JavaScript for Automation. It controls the browser by directly communicating with it.
- Selenium WD supports multiple Programming language. Java, C#, Python, PHP, Perl, Ruby

Selenium Grid

- Selenium Grid is tool to use for Parallel Execution.
- Selenium Grid is compatible with RC and WD both
- It works like an Execution Engine for Selenium Tests.
- Enables simultaneous running of tests in multiple browsers and environments
- Utilizes the hub-and-nodes concept. The hub acts as a central source of Selenium commands to each node connected to it.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

We learnt various concepts such as Software Testing methods, MySql, Java programming and Selenium WebDriver. We utilized all these concepts or features to learn about automation using Selenium.

6.2 Future scope of work

In future I will learn more about Selenium and also will make a project to automate it involving all the concepts learnt during my internship.

REFERENCES

- [1] <https://cognizant.udemy.com/course/testerbootcamp/learn/lecture/20915540#overview>
- [2] <https://cognizant.udemy.com/course/selenium-webdriver-web-based-automation-testing/learn/lecture/13207744#notes>
- [3] <https://www.javatpoint.com/java-tutorial>
- [4] GeeksforGeeks

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ...24-06-2021.....

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: Ananya Bajpai _____ Department: CSE _____ Enrolment No 171221

Contact No. 7007061212 _____ E-mail. ananyabajpai11@gmail.com _____

Name of the Supervisor: Dr. Jagpreet Singh Sidhu

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _

A QUALITATIVE APPROACH TO THE QUALITY ENGINEERING ASSESSMENT BASICS

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

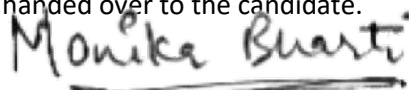
- Total No. of Pages = 49
- Total No. of Preliminary pages = 6
- Total No. of pages accommodate bibliography/references =1



(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 26 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.



(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none">• All Preliminary Pages• Bibliography/Images/Quotes• 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

QUALITATIVE APPROACH TO THE QUALITY ENGINEERING ASSESSMENT BASICS"

ORIGINALITY REPORT

26%
SIMILARITY INDEX

20%
INTERNET SOURCES

2%
PUBLICATIONS

16%
STUDENT PAPERS

PRIMARY SOURCES

1 www.coursehero.com 5%
Internet Source

2 blog.udemy.com 3%
Internet Source

3 www.tutorialspoint.com 2%
Internet Source

4 www.javatpoint.com 2%
Internet Source

5 www.guru99.com 1%
Internet Source

6 sonisoft.simplesite.com 1%
Internet Source

7 Submitted to University of Bolton 1%
Student Paper

8 Submitted to Harare Institute of Technology 1%
Student Paper

9 Submitted to Westcliff University 1%
Student Paper

10	Submitted to Kingston University Student Paper	1 %
11	Submitted to Harrisburg University of Science and Technology Student Paper	1 %
12	Submitted to Icon College of Technology and Management Student Paper	1 %
13	"Relationship Identification & Prediction of Diseases Association using Micro-RNA of Genomic Data", International Journal of Innovative Technology and Exploring Engineering, 2019 Publication	1 %
14	Submitted to Kent Institute of Business and Technology Student Paper	1 %
15	docplayer.net Internet Source	1 %
16	javarevisited.blogspot.lu Internet Source	1 %
17	Submitted to Campbellville University Student Paper	1 %
18	Submitted to Jaypee University of Information Technology Student Paper	<1 %

19	www.scribd.com Internet Source	<1 %
20	blog.ssdnodes.com Internet Source	<1 %
21	Submitted to Colaiste Dhualigh College of Further Education Student Paper	<1 %
22	Submitted to South University Student Paper	<1 %
23	Submitted to Alliance University Student Paper	<1 %
24	Submitted to Anglia Ruskin University Student Paper	<1 %
25	ebooks.lpude.in Internet Source	<1 %
26	iugspace.iugaza.edu.ps Internet Source	<1 %
27	www.slideshare.net Internet Source	<1 %
28	epub.lib.aalto.fi Internet Source	<1 %
29	eprints.utm.my Internet Source	<1 %