Week1 – RDBMS – Relational Database Management System (MySQL, MS-SQL, Oracle, Postgres…. )

Flat file (.txt, .rtf, .doc…) VS DB Files

In Database, everything is entity (Table, View, Cursor, Trigger, Function, Stored Procedure, Sequence ….)

Bcos of using Mathematical Relation’s theory it is called as RDBMS.

MySQL – Open Source RDBMS which has many enterprise features. It’s easy to use database.

1. Using Command Line Client (using command prompt – mysql -u root -p) Enter the root password. Knowledge of SQL queries are important to work in command line client
2. Using GUI – MySQL Workbench – Interactively you can create /modify entities. Delete and read data.
3. Using programming Lang (Any programming Lang like java, C++, JS, Node, Python, php, jsp….)

DDL, DML, DCL, TCL, DQL

Relationships between Tables (1-1, 1-n, m-n)

Various Clause (Where, any, in, between, like)

Constraints – Primary Key, Unique, Not null, Check, Foreign Key

Sub-query – Query within a query (The inner query will be evaluated first and the result of the inner query will be used to retrieve the data in the outer query)

Joins (Self Join, inner & Outer Join, Left & Right join)

DB Normalization (1NF, 2NF, 3NF, 3.5NF (BCNF), 4NF,5NF…7NF)

ER Diagram – Using MySQL workbench. Creating tables, views in the GUI with the help of diagrams.

NO-SQL Database (MongoDB)

The data will be stored in the form of BSON. It’s highly recommended for un-structured data storage.

The database will store the values as collections of key-value pairs, in each insert this collection can hold different set of value.

1. Mongo – Mongo DB command line client (Can execute Mongo commands) (use <db\_name>,db, show dbs, show collections, )

CRUD – insert(), save(), upsert(), find(), remove()

1. Using GUI – MongoDB Compass – Easy & interactive way to connect and handle the data
2. MongoDB can connect with Local/Remote Mongo Server
3. Mongo Atlas – Cloud variant of MongoDB (Free Account – Only one private repo, n – no of public repo)
4. This is also can connected with the help of programming Lang.

Core JAVA

1. Lang Fundamentals – Data types, keywords, operators,
2. OOPs Concepts – Inheritance, Polymorphism, Encapsulation & Abstraction)
3. Inheritance (using extends class & implements interface) – Acquiring property of other class
4. Encapsulation (using access modifiers – Using private data member & public methods) – data hiding
5. Abstraction – Implementation hiding (Abstract class and interface)
6. Polymorphism – Re-using the same code either by changing / without changing the signature

(Static & Dynamic)

1. String, StringBuffer & StringBuilder
2. Packages, Interfaces, Statements (Conditional -control, looping control statements)

Agenda –

1. What is mean by Annotations
2. Enums
3. Exceptions Handling

Annotation – Meta data – Data about data – Real time Example – Book (The JAVA Complete Reference – This book is all about JAVA programming lang – TOC (Table of contents – Chapter wise details – What topics are we discussing in each chapter, Index – Alphabetical order the concepts related to java will be given and in pages these concepts are discussed will also be there)

In Java, meta data/ annotations are helpful to provide more information about the code to the JVM.

Annotations can be applied in various level (class level, method level, property level annotations)

Annotations in Java starts with “@” symbol. It’s introduced in JAVA 5

It helps to reduce lot of configuration code like web.xml, spring-config.xml, hibernate-mapping.xml and so on….

Annotations are divided into two types

1. Built-in/ Creator defined annotations (@Override, @SupressWarnings({“unused”,”deprecation”}), @Deprecated
2. Custom/User defined annotations

Annotations are widely used in frameworks like spring, hibernate, adv java (servlet, servlet-mapping, - @WebServlet – will help to create an URL mapping to the servlet.

Java Enterprise/ dynamic Web Application – web.xml (Deployment descriptor – Compulsory if we are using java version < 1.5 (Java 5)

If we are using java5, web.xml is optional. Annotations are introduced in Java5 (jdk1.5)

Meta- Annotations are nothing but annotation used to create custom annotation

1. @Retention (source/class/runtime)
2. @Documented
3. @Target

Annotations are mainly used in frameworks.

Enum – Derived data type – Used to define set of values to pass it to a particular data type;

Month Names – Jan, Feb, Mar, …Dec.

WeekDays – Sunday, Monday, … Saturday

Weekdays – SUN, MON, TUE,WED, THU, FRI,SAT – It’s a kind of Array.

Enum – is a derived data type which contains fixed set of Constants.

Exceptions Handling

Exceptions – It’s a un-expected condition while running the program which may lead to pre-mature closure/terminate of the program. [Ending a program in half-way or before completing execution of all lines]

Exceptions Types (By creators)

1. Built-in Exception /Creator developed/defined Exceptions
2. Custom Exceptions / User defined exception

Category Of Exception (Depending on it’s throwing time)

1. Compile-time – Checked Exceptions (IO, SQL Exceptions)
2. Run-Time - Un-checked Exceptions ( All Exception class which is extending Run-Time Exception class) Ex: Arithmetic, IndexOutOfBound, NullPointer Exceptions

Exception Hierarchy

Throwable – 1) Error 2) Exception

Two ways of Handling Exceptions

1. Using try/catch/finally block – (Recommended approach – bcos, we are handling the exceptions and telling what to do when it is thrown)
2. Using throws keyword (Here we are just passing the control to the JVM)

In System class there are 3 streams 1) in (input) 2)out (output) 3) err (error) – The text in console will be red colour.

Few points to remember in Exceptions Handling

1. When you use try/catch/ finally block, it is very important to have try block followed by either catch or finally block
2. We can have a try block with finally block, without catch block. (valid)
3. We can have a try block with catch block only without finally block. (Valid)
4. It’s not allowed to write a try block without catch/finally block
5. One catch block could catch multiple exceptions (Valid)
6. Try block could be followed by multiple catch block.
7. When using try/catch/finally block, only one try & finally block should be used.
8. Usually resource releasing code/ file closing code will be written inside finally block.
9. They code inside finally block will get executed all the times irrespective of the exception status.
10. Try with resource is also possible instead of finally block. When using try(resource) { };
11. Try with resource will automatically close the resource after using it.
12. Finally block, closing files, closing db connections and other sql based object references

Custom Exception