3 days session on

jdbc

jpa

i assume you have done sql on the ADAPT platform

familarity with sytnax is expected

select, insert, update,delete

session plan with you : for 3 days

what are we going to cover in these 3 days

classbook, lab book and demos of these 3 days shared with u ?

jdbc : java database connectivity

how does a java program accept data from the user ?

module core java mpt is done ?

u did not use a Scanner ?

enter emp name

enter phone

input validation ...

added the details into a hashmap : in core java test

the same data which was accepted from the user will now be inserted into the table

insert, fetch, update, delete will now be performed on the table instead of hashmap

the steps a java program uses to connect with the database

1. identify an appropriate database driver : OracleDriver.class

java program oracle

database driver

different types of dataase drivers are available

type 1

type 2

type 3

type4 : ojdbc.jar

this driver is written in java itself. it directly connects to the db

wherever oracle is installed ; search for a file called ojdbc\*.jar

ojdbc6.jar

the database driver is in a jar file

jar file contains many .class files

u will not find .java inside a jar file

OracleDriver is the main class inside this jar file

the compiler gives an error as the location of the OracleDriver.class is not known

String.class is present in rt.jar file which is automatically present in every java project

similarly, since OracleDriver.class is present in ojdbc.jar, we have to tell the compiler

about this new jar file

this is called as setting the class path

before setting classpath, copy this ojdbc6.jar in your java project

how to set the classpath from within eclipse ?

right click the java project --> build path --> configure build path --> libraries tab -->

add jars button --> select ojdbc6.jar and click ok/finish

eclipse will prompt for "import" only when the jar is included in the classpath

2. load this driver / register this driver with java : DriverManager.registerDriver(...

can also be done using Class.*forName*(driver);

both of the syntaxes can be used for jdbc code

what is the difference between teh two

DriverManager.registerDriver(Driver) : will work only with Driver classes

i.e DriverManager.registerDriver(String) will give error

Class.*forName*() : will load any class name given

Class.*forName*(String.class)

Class.*forName*(OracleDriver.class) will also work

3. using this driver try to connect to the db by providing username, password, ipaddress of oracle server and port number of oracle server

tell me something .. when u manually connect with the database, what info do u provide ?

user name , password, host string ( database name)

are u aware of a special file called tnsnames.ora within oracle installation folder ?

tell me the info in that file which u understand ?

1. option1

do u have oracle expression edition on your machine??

this means that oracle server and client are on the same machine.

each machine in the room has an individual server and individual client

2. option 2

you only have the client part of oracle on your machine and the server is located in some remote place

i have option 2 on my machine

1. ip address of the oracle server the client is pointing to :10.51.103.201:

2.port number the oracle server listens on : 1521

every server "listens" on a port number for "client request "

oracle is based on request and response kind of technology

oracle client : sends the oracle commands; this is the request

the command goes to the server which returns a response based on the request

DriverManager.*registerDriver*(driver);

the method registerDriver throws an exception which is of type checked exception , SQLException

wheenever u call a method in the java program and that method throws a checked exception, the compiler gives u an error

eclipse gives u two choices.

either

1. try catchthis exception

2.or provide a throws clause

when will this exception actually occur ?

what is the method trying to say

ans : the method says that during its processing, something may go wrong and the method

is warning u of the same

what is the relation between OracleDriver class and Driver interface

class OracleDriver class implements Driver{

}

layered architecture

dao layer : u will create a new class which will interact with the database

database properties file

why do i need to keep this in the project ?

what should i write this in file ?

any name; but extension should be .properties

this file contains entries in key value pair

key name : can be any name ; but remember that it will be used in the java code

value for hte key : will be proper values

in the TestingOracleConnectivity project u have written all the info in the main method

maintaining the same info in the properties file is a better option to make the project

more flexible and manageable

thats why the info is being maintained in a properties file

also, create a separate class , DbUtil which will have a method to read from this properties file

what is a java bean ?

ans ; there will be many classes in a java application

atleast one of the class will be mapped to a database table

such a class is called a java bean

PatientBean :Pascal case convention

teh first letter of every word is in capitals

**private** **int** patient\_id;//wrong naming conventin

right one should hav been patientId;//camel case convention

is it a good practice to keep attribute names as private ?? yes

so that abstraction and encapsulation can take place

there will be a table called Patient in the oracle datbaase

the attributes of this class will become column names in this table

service layer will contain methods for enforcing business rules

ie. validation methods will be kept here

validateFirstName

validatePhoneNumber etc etc

userEnteredName.matches("[A-Z][a-z]{1,19}"

above means first leter should be capital

the rest can be small letters from 1,19

in this working code, u accepted the patient name, age, phone num, description

otuput of this code: a record was inserted into the patient table

let us debug the application to find out why record was not inserted

the values entered by the user should be stored into the bean class object. yes

but why ? why cant i pass the values individually to the service layer method

service.addPatientDetails(patient);

if many values are related to each other, it is better to pass an object

instead of passing individual values

service.addPatientDetails(name,age,description,phone);this is not a right way

why is this method returning an integer ?

because after inserting a patient record; we have to tell the user what his patient id is

eg. welcome kavita; you have been registered with this clinic and your patient id

for further communication is 1001

msg shown to the end user should be meaningful . it shoudl not be technical

when the service layer method forwards the request to the dao layer method, we call

this as delegating the request

this layered architecture is followed by each one of us in our day to day life

eg. if i tell u to check whether oracle is working in the lab; u might delegate this request to the

itimd person because he has the skills to do the same

for every method /functionality in the application , u have to identify an sql staetment

eg. for adding a patieent in the table; the sql statemnet is "insert "

there are some interfaces availabel in java.sql package for working with different types of sql statements

1. static select

2.dynamic select

3.insert /update/delete : this will involve some variables

i.e the value entered by the user has to be passed to the insert statement. do u agree ?

the number of question marks in the insert statement depends on the number of columns in which data has to be inserted

does the order of the column matter ?

if the query in sql is insert/update/delete : the corresponding java method will be

executeUpdate()

if the query is select then the jav amethod will be executeQuery()

do u have the emp table in oracle ?

desc emp

your task after our discussion will be to accept data from user and insert a record

into emp table

can the dao class method print a msg to the end user ?

ans : no

the dao class should tell the service class which in turn will tell main method which

will inform to the user

create sequence empseq;

//above command will create a sequence which will have initial value as 1; increment value as 1

sequence is a database object

it is used for automatically incrementing numbers

u can determine the initial value; the increment value and the max value

so.. how to tell hte sequence to generate a value and give it to me

select empseq.nextval from dual;

otuput 1;

select empseq.nextval from dual; //dual is a default dummy table provided by oracle

otuput 2;

what should i do with the values generated by a sequence?

ans : it will be used as primary key values

create sequence myseq start with 1000 ;

first time : 1000

next time : 1001

when u all joined CG, your emp num was automatically generated and told to u right ?

u did not choose your emp num ; primary key value is never accepted from the user

it is generated using a sequence

ResultSet concept in java is an interface

it is hte return type of executeQuery method

whenever u use select statement in java, u will have to work with REsultSet

this is similar to cursor concept in pl sql

it will point to the output of select statement in the memory

to process a result set, next method has to be used

when used for the first time, next will point to the first record in the output of select

and then on, it will point to subsequent records

JPA :

java persistence API

API ( application programming interface )

consists of classes and itnerfaces , its methods

work with JDBC API means to understand the different classes and itnerfaces availabel in java.sql

pkg, learn which method to use for which functionality

to work with JPA : is to understnad the differnet classes and interface and its methods and how

to use which method to interact with the datbaase

JPA is a specification /standard

there are many vendors who have implemented JPA

hibernate is one of the vendors/providers

there can be other providers also

JPA via hibernate is what u r going to learn

so what is the concept behind JPA?

today, as a java developer if u wnat to itneract with the datbaase, u need to know the following

1. knowledge of sql is a must ; sql syntax is needed to be known

2. working with JDBC API ; working with Connection,PreparedStatement,ResultSet etc will not be required

so JPA says, why should a java developer need to know sql

if u use JPA via hibernate, hibernate would generate sql on your behalf

no need to handle sqlexceptions

hibernate would establish connection on your behalf

hibernate would generate sql statements based on which API method u choose in the code

hibernate would handle exceptions for u

so , the advantage is that

as a developer, your code becomes shorter /reduced

in teh following line, con is an object reference of Connection interface

Connection con;

con.prepareStatement()

what is meant by calling method on an interface ?

ans : method is called on the class which implements the interface Connection

do u need to know the name of this child class of Connection interface ?

ans : no

polymorphism is what is taking place here

u only worked with interface name; u dont need to know the child class name

class OracleCOnnection implements Connection {

prepareStatement(){

//ojdbc.jar file

}

}

output of today's code will be same as yesterday

i.e a record shoudl be inserted into the table

fetch records from table

update records

delete records

but the code written by u today will be much shorter than yest

as hibernate would do all the work for u

so, while working with jpa, shoudl i create the table by myself or will hibernate create it for me ?

ans : u can do both

scenario 1

if the application's first page requires you to fetch all the records of the table and show to user :

then this means that data should already be present in the table

which means that user has to create the table and insert records by himself from hte back end

i.e logging into oracle

scenario 2:

application has a hyper link which displays a form; user enters value into this form and on click of submit

button; record has to be inserted

in this case; u can ask hibernate to create the table also for u

@Entity : compulsory annotation

this is given at class level

this annotation tells hibernate that this class will be mapped to a table in the database

if the table does not exist, hibernate can create this for u

should hibernate create the tbale for u

or should the develper create, u have to decide this

hibernate will assume that table name is class name ; i.e Employee

if the table name is different, then use another annotation called @Table

@Table is not compulsory to be used. why ?

@Entity //compulsory

class Employee // table name will be assumed to be employee

{

@Id //compulsory annotation; used with teh primary key column of the table

int empId; //attributes of the class will become column names of the table

if u wnat to change teh column name; use @Column annotation which is optional

@Column("employee\_name",length=20)//if hibernate creates the table; it uses varchar2(255) for String datatype which becomes an issue for us

String empName;

}

how to tell hibernate which class to work with : @Entity

first program using JPA

1. bean class which has been annotated

2. main program : use the JPA API to interact with hibernate and persist/insert a record

3.persistence.xml file : file name has to be this and should be in a special folder called

META-INF

<employees>

<employee>

<emp\_id>7809</emp\_id>

<emp\_name>2234</emp\_name>

</employee>

<employee>

<emp\_id>7809</emp\_id>

<emp\_name>2234</emp\_name>

</employee>

</employees>

<property name=*"hibernate.hbm2ddl.auto"* value=*"create"*/>

if the above line is written and the table does not exist, hibernate will create for u

if the table exists, it will get overwritten

and u have written code to insert records;

u executed the code for the first time and checkced the table; one row was inserted

u executed program second time and u r wondering there shoudl have been two rows but u

r able to see only one row. what u think has happened ? ans : table was overwritten

<property name=*"hibernate.hbm2ddl.auto"* value=*"update"*/>

find out that when this entry is not written in xml, what is the default value

if the above line is written and the table does exist, hibernate will ignore it

if the tbale does not exists, it will get created

hibernate\_sequence : sequence has to exist ; before executing the code

@GeneratedValue(strategy = GenerationType.***SEQUENCE***, generator = "USERS\_SEQ")

@SequenceGenerator(name = "USERS\_SEQ", sequenceName = "SEQUENCE\_USERS",allocationSize=10)

try out

if @ Id is not given; find out the error which occurs. dont make the same mistake again

CRUd : create(insert), read, update and delete

these are teh operations whch are performed on a table

find out what happens when u write persist but dont start and commit a transaction

i.e only write persist; comment out begin and commit transaction

persist has to be written within a transaction only ; otherwise error occurs

similar to dbutil class, there is an other class involved in jpa, can be any name but in this project

it is called jpautil

new metods in entity manager

persist : for inserting

remove : for deleting row

merge: for updating row

find : for finding one record based on primary key

for finding many records :u will have to write JPQL : java persistence query language which v will do tmrw

understand the crud operation java project and create one on your own

either work with Product table or Employee table

and try out all the CRUD operations on this table

we have to discuss jpa queries project also today

so take time to try out CRUD operations

u have 30 mins

all the best

why should v use JPQL when find method is already present ?

when u need to fetch records from the table either

1. based on a condition which does not involve primary key ie search based on book name, price etc

2.to fetch more than one record

for fetching one record based on primary key , u can directly use find method

how is JPQL different from SQL ?

jpql is case sensitive ; sql is not

in jpsql syntax, class names and property names are given ; book is different from Book right ?

in sql syntax, column names and table names are given

SELECT book FROM Book book : will be similar to select \* from book

use query.getResultList if the query returns many rows

query.getSingleResult() : use this when the query returns only one row

named query syntax is imp

named query is a query which has been given a name so that it can be reused

this annotation is given at bean class level