# Table Per Concrete class using Annotation

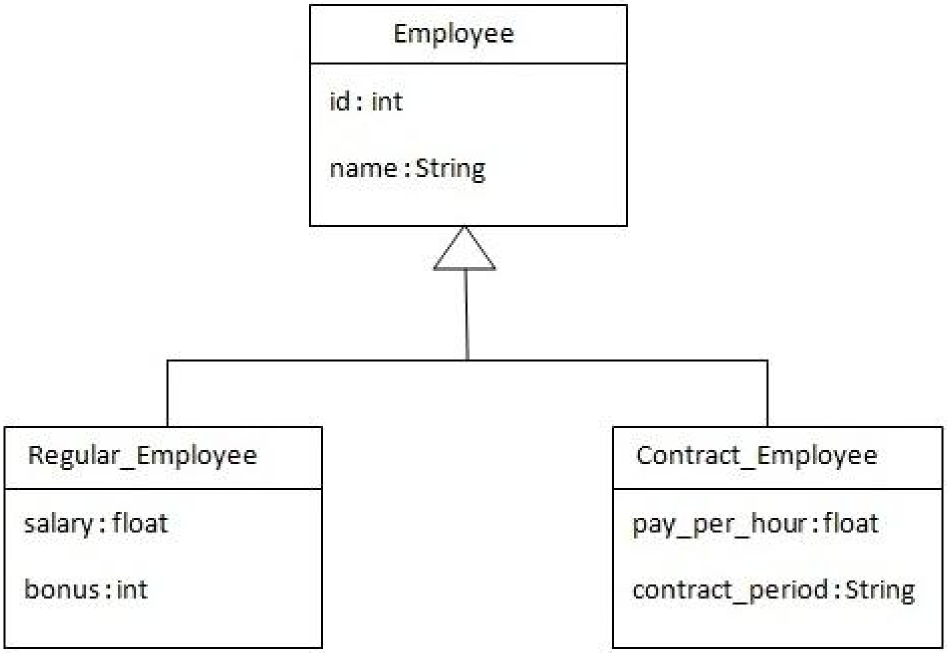
In case of Table Per Concrete class, tables are created per class. So there are no nullable values in the table. Disadvantage of this approach is that duplicate columns are created in the subclass tables.

Here, we need to use @Inheritance(strategy = InheritanceType.TABLE_PER_CLASS) annotation in the parent class and @AttributeOverrides annotation in the subclasses.

**@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)** specifies that we are using table per concrete class strategy. It should be specified in the parent class only.

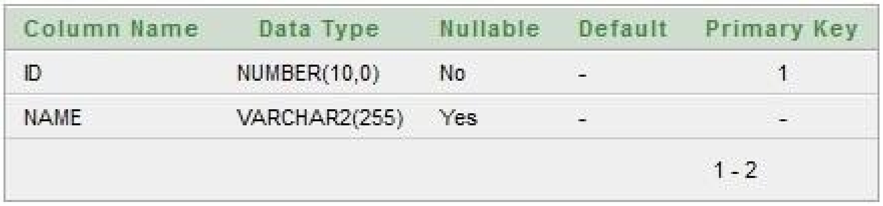
**@AttributeOverrides** defines that parent class attributes will be overriden in this class. In table structure, parent class table columns will be added in the subclass table.

The class hierarchy is given below:



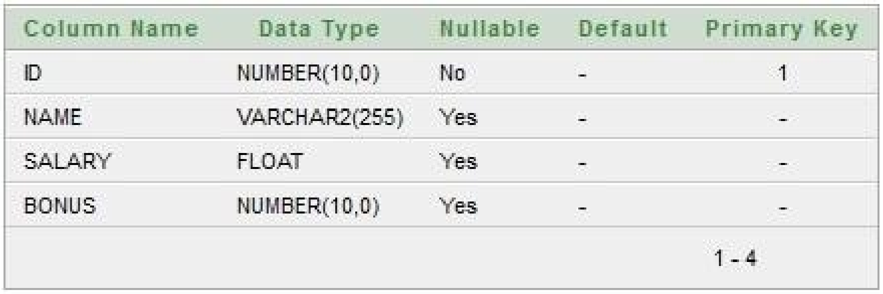
The table structure for each table will be as follows:

## Table structure for Employee class



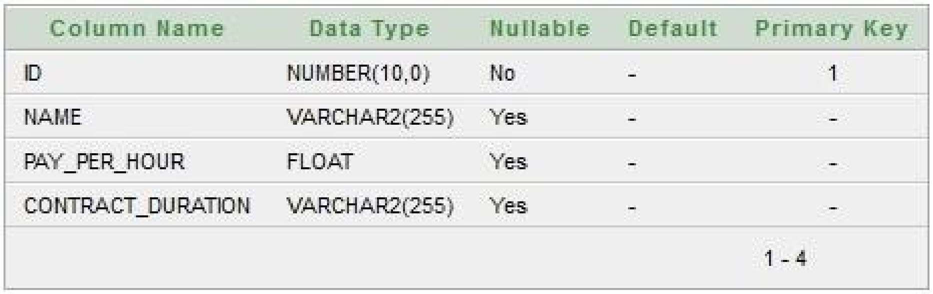
| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| ID | NUMBER(10,0) | No | - | 1 |
| NAME | VARCHAR2(255) | Yes | - | - |
| | | | | 1 - 2 |

### Table structure for Regular_Employee class



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| ID | NUMBER(10,0) | No | - | 1 |
| NAME | VARCHAR2(255) | Yes | - | - |
| SALARY | FLOAT | Yes | - | - |
| BONUS | NUMBER(10,0) | Yes | - | - |
| | | | 1 - 4 | |

### Table structure for Contract_Employee class



| Column Name | Data Type | Nullable | Default | Primary Key |
|---|---|---|---|---|
| ID | NUMBER(10,0) | No | - | 1 |
| NAME | VARCHAR2(255) | Yes | - | - |
| PAY_PER_HOUR | FLOAT | Yes | - | - |
| CONTRACT_DURATION | VARCHAR2(255) | Yes | - | - |
| | | | 1 - 4 | |

## Example of Table per concrete class

In this example we are creating the three classes and provide mapping of these classes in the employee.hbm.xml file.

### 1) Create the Persistent classes

You need to create the persistent classes representing the inheritance. Let's create the three classes for the above hierarchy:

*File: Employee.java*

```
package com.javatpoint.mypackage;
import javax.persistence.*;


@Entity
@Table(name = "employee102")
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)


public class Employee {
@Id
@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name = "id")
private int id;

@Column(name = "name")
private String name;

//setters and getters
}
```

*File: Regular_Employee.java*

```
package com.javatpoint.mypackage;
import javax.persistence.*;

@Entity
@Table(name="regularemployee102")
@AttributeOverrides({
    @AttributeOverride(name="id", column=@Column(name="id")),
    @AttributeOverride(name="name", column=@Column(name="name"))
})
public class Regular_Employee extends Employee{

@Column(name="salary")
private float salary;

@Column(name="bonus")
private int bonus;

//setters and getters
}
```

*File: Contract_Employee.java*

```
package com.javatpoint.mypackage;
import javax.persistence.*;
@Entity
```

```
@Table(name="contractemployee102")
@AttributeOverrides({
    @AttributeOverride(name="id", column=@Column(name="id")),
    @AttributeOverride(name="name", column=@Column(name="name"))
})
public class Contract_Employee extends Employee{

    @Column(name="pay_per_hour")
    private float pay_per_hour;

    @Column(name="contract_duration")
    private String contract_duration;

    public float getPay_per_hour() {
        return pay_per_hour;
    }
    public void setPay_per_hour(float payPerHour) {
        pay_per_hour = payPerHour;
    }
    public String getContract_duration() {
        return contract_duration;
    }
    public void setContract_duration(String contractDuration) {
        contract_duration = contractDuration;
    }
}
```

## 2) Add mapping of hbm file in configuration file

*File: hibernate.cfg.xml*

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
          "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
          "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<!-- Generated by MyEclipse Hibernate Tools.                   -->
```

```
<hibernate-configuration>
    <session-factory>
        <property name="hbm2ddl.auto">update</property>
        <property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>
        <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
        <property name="connection.username">system</property>
        <property name="connection.password">oracle</property>
<property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>

        <mapping class="com.javatpoint.mypackage.Employee"/>
        <mapping class="com.javatpoint.mypackage.Contract_Employee"/>
        <mapping class="com.javatpoint.mypackage.Regular_Employee"/>
    </session-factory>
</hibernate-configuration>
```

The hbm2ddl.auto property is defined for creating automatic table in the database.

## 3) Create the class that stores the persistent object

In this class, we are simply storing the employee objects in the database.

*File: StoreData.java*

```
package com.javatpoint.mypackage;

import org.hibernate.*;
import org.hibernate.cfg.*;

public class StoreData {
public static void main(String[] args) {
AnnotationConfiguration cfg=new AnnotationConfiguration();
Session session=cfg.configure("hibernate.cfg.xml").buildSessionFactory().openSession();

    Transaction t=session.beginTransaction();

    Employee e1=new Employee();
    e1.setName("sonoo");
    Regular_Employee e2=new Regular_Employee();
    e2.setName("Vivek Kumar");
    e2.setSalary(50000);
    e2.setBonus(5);

    Contract_Employee e3=new Contract_Employee();
    e3.setName("Arjun Kumar");
    e3.setPay_per_hour(1000);
    e3.setContract_duration("15 hours");

    session.persist(e1);
    session.persist(e2);
    session.persist(e3);

    t.commit();
    session.close();
    System.out.println("success");
}
}
```

download this example developed using myeclipse ide

← prev

next →