

221005 수업 -1

🕒 작성일시	@2022년 10월 5일 오전 9:13
≡ 키워드	ResultSetMetaData, 트랜잭션, MVC 패턴, DTO, DAO
▼ 유형	
🔗 자료	
☑ 복습	<input type="checkbox"/>

ResultSetMetaData

- DB가 생성한 컬럼의 메타 데이터 확인

```
ResultSetMetaData rsmd = rs.getMetaData();

System.out.println("Total columns: "+rsmd.getColumnCount());
System.out.println("Column Name of 1st column: "+rsmd.getColumnName(1));
System.out.println("Column Type Name of 1st column: "+rsmd.getColumnTypeName(1));
```

하나의 DB서버를 공유해 사용하는 경우 생기는 문제점?

- 한명의 사용자가 DML 작업을 한 경우 반드시 Commit, rollback을 통해 작업을 완료 해야함
- 그렇지 않으면 다른 사용자는 DML 작업이 불가함(무한 대기 상태)

트랜잭션

- 하나의 논리적인 단위 업무

JDBC에서 DML 작업을 하는 경우 기본 autocommit

→ autocommit을 개발자가 false로 바꿀 수 있음

→ 반드시 commit, rollback을 해야함

```

public class Ex07_Transaction {

    public static void main(String[] args) {

        Connection conn = null;
        PreparedStatement pstmt = null;
        PreparedStatement pstmt2 = null;

        conn = SingletonHelper.getConnection("oracle");

        String sql = "insert into trans_A(num, name) values(100, 'A')";
        String sql2 = "insert into trans_B(num, name) values(100, 'B')";

        try {
            //둘다 성공 아니면 둘다 실패 (하나의 논리적 단위)
            conn.setAutoCommit(false); //JDBC 개발자 (commit or rollback)
            //begin tran
            pstmt = conn.prepareStatement(sql);
            pstmt.executeQuery();

            pstmt2 = conn.prepareStatement(sql2);
            pstmt2.executeQuery();
            //end tran
            conn.commit(); //둘 다 정상적으로 실행되고 예외가 발생되지 않으면 : commit
        } catch (Exception e) {
            System.out.println("문제 발생 : " + e.getMessage());
            try {
                conn.rollback(); //둘 다 취소
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
        } finally {
            SingletonHelper.close(pstmt);
            SingletonHelper.close(pstmt2);
        }

    }

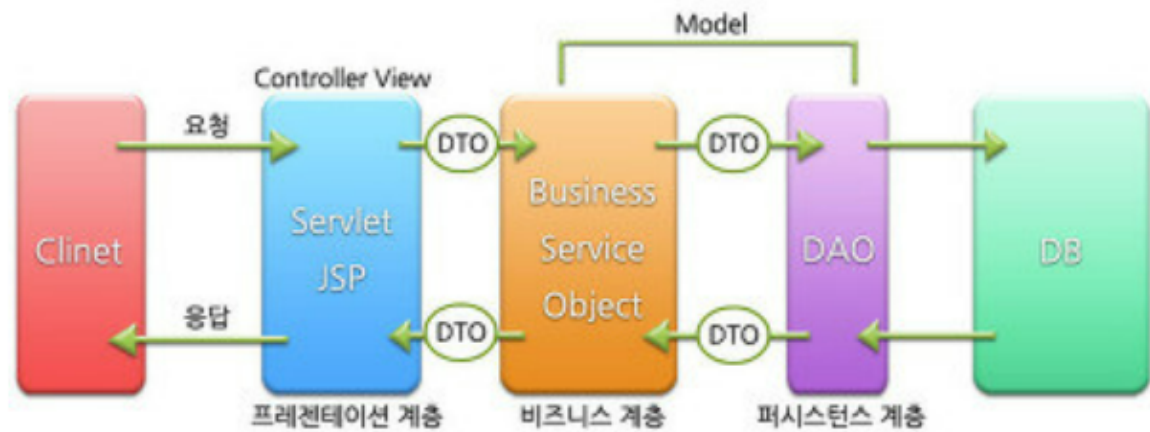
}

```

- trans_A 테이블에 데이터를 넣는 작업과 trans_B 테이블에 데이터를 넣는 작업을 하나의 트랜잭션으로 만듦
- 둘 중 하나라도 실패하면 모두 rollback

MVC 패턴

- Model, View, Controller로 나누어 개발

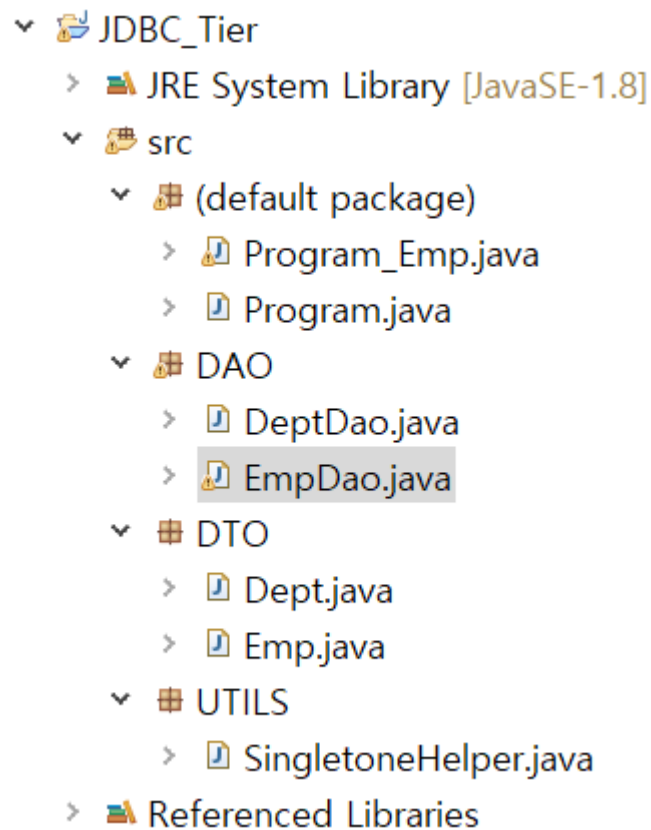


- DTO : 데이터 전송 위한 객체

Model : 데이터(DTO, DAO) → java

View : 화면 → jsp

Controller : 조정(제어) → servlet



DTO(VO), Domain 클래스

- 데이터를 1건 담을 수 있는 클래스
- DB 테이블과 1:1 매핑구조
- 1건 데이터 → `Dept dept = new Dept();`
- 여러 건 데이터 → `List<Dept> deptlist = new ArrayList<Dept>();`

[DTO 패키지 > Dept.java]

```
public class Dept {  
  
    private int deptno;  
    private String dname;  
    private String loc;  
    public int getDeptno() {
```

```

        return deptno;
    }
    public void setDeptno(int deptno) {
        this.deptno = deptno;
    }
    public String getDname() {
        return dname;
    }
    public void setDname(String dname) {
        this.dname = dname;
    }
    public String getLoc() {
        return loc;
    }
    public void setLoc(String loc) {
        this.loc = loc;
    }

    @Override
    public String toString() {
        return "Dept [deptno=" + deptno + ", dname=" + dname + ", loc=" + loc + "]";
    }
}

```

DAO

1. DB 연결
2. DRUD 구현

- 기본적으로 함수 5개 (하나의 테이블에 대해서)
 - 전체 조회, 조건 조회, 삽입, 수정, 삭제 기능

```

public class DeptDao {

    //1. 전체 조회 >> select * from dept >> return multi row
    public List<Dept> getDeptAllList(){
        Connection conn = null;
        PreparedStatement pstmt = null;
        ResultSet rs = null;
        List<Dept> deptList = new ArrayList<Dept>();

        try {

```

```

    conn = SingletonHelper.getConnection("oracle");
    String sql = "select * from dept";
    pstmt = conn.prepareStatement(sql);
    rs = pstmt.executeQuery();

    if(rs.next()) {
        do {

            Dept dept = new Dept();
            dept.setDeptno(rs.getInt("deptno"));
            dept.setDname(rs.getString("dname"));
            dept.setLoc(rs.getString("loc"));

            deptList.add(dept);

        }while(rs.next());
    }else {
        System.out.println("조회된 데이터 없음");
    }

} catch (Exception e) {
    System.out.println(e.getMessage());
} finally {
    SingletonHelper.close(rs);
    SingletonHelper.close(pstmt);
}

return deptList;
}

//2. 조건 조회 >> select * from dept where deptno=?
//return signlerow
public Dept getDeptListByDeptNo(int deptno) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    Dept dept = new Dept();

    try {
        conn = SingletonHelper.getConnection("oracle");
        String sql = "select * from dept where deptno=?";
        pstmt = conn.prepareStatement(sql);

        pstmt.setInt(1, deptno);

        rs = pstmt.executeQuery();

        dept.setDeptno(rs.getInt("deptno"));
        dept.setDname(rs.getString("dname"));
        dept.setLoc(rs.getString("loc"));

    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        SingletonHelper.close(rs);
        SingletonHelper.close(pstmt);
    }
}

```

```

    return dept;
}

//3. 데이터 삽입 >> insert into deptno(deptno, dname, loc) vlaues(?, ?, ?)
//return null;
public int insertDept(Dept dept) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    int row = 0;

    try {
        conn = SingletonHelper.getConnection("oracle");
        String sql = "insert into deptno(deptno, dname, loc) values(?, ?, ?)";
        pstmt = conn.prepareStatement(sql);

        pstmt.setInt(1, dept.getDeptno());
        pstmt.setString(2, dept.getDname());
        pstmt.setString(3, dept.getLoc());

        row = pstmt.executeUpdate();

    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        SingletonHelper.close(rs);
        SingletonHelper.close(pstmt);
    }

    return row;
}

//4. 데이터 수정 >> update dept set deptno=?, dname=?, loc=?
public int updateDept(Dept dept) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    int row = 0;

    try {
        conn = SingletonHelper.getConnection("oracle");
        String sql = "update dept set deptno=?, dname=?, loc=?";
        pstmt = conn.prepareStatement(sql);

        pstmt.setInt(1, dept.getDeptno());
        pstmt.setString(2, dept.getDname());
        pstmt.setString(3, dept.getLoc());

        row = pstmt.executeUpdate();

    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        SingletonHelper.close(rs);
        SingletonHelper.close(pstmt);
    }

    return row;
}

```

```

}

//5. 데이터 삭제 >> delete from dept where deptno = ?
public int deleteDept(int deptno) {

    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    int row = 0;

    try {
        conn = SingletonHelper.getConnection("oracle");
        String sql = "delete from dept where deptno=?";
        pstmt = conn.prepareStatement(sql);

        pstmt.setInt(1, deptno);

        row = pstmt.executeUpdate();

    } catch (Exception e) {
        System.out.println(e.getMessage());
    } finally {
        SingletonHelper.close(rs);
        SingletonHelper.close(pstmt);
    }

    return row;
}

//부서이름을 LIKE 검색해서 해당하는 부서번호, 부서이름, 부서위치 를 반환하는 함수
//select deptno, dname, loc from dept where dname like '%?%'
//무식하게 표현하면 where dname like '%' + dname + '%"
public List<Dept> getDeptListByDname(String dname){

    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    List<Dept> deptList = new ArrayList<Dept>();

    try {
        conn = SingletonHelper.getConnection("oracle");
        String sql = "select deptno, dname, loc from dept where dname like ?";
        pstmt = conn.prepareStatement(sql);

        pstmt.setString(1, "%" + dname + "%"); //'%?%'

        rs = pstmt.executeQuery();
        if(rs.next()) {
            do {

                Dept dept = new Dept();
                dept.setDeptno(rs.getInt("deptno"));
                dept.setDname(rs.getString("dname"));
                dept.setLoc(rs.getString("loc"));

                deptList.add(dept);

            }while(rs.next());
        }
    }
}

```



```

        }else {
            System.out.println("조회된 데이터 없음");
        }

    } catch (Exception e) {
        System.out.println(e.getMessage());
    }

    return deptList;
}
}

```

[default 패키지 Program.java]

```

public class Program {

    public static void main(String[] args) {

        //main 함수 (view) 영역
        //DAO 객체 요청
        DeptDao dao = new DeptDao(); //6개 함수

        System.out.println("[전체 조회]");
        List<Dept> deptList = dao.getDeptAllList();
        //화면 출력
        if(deptList != null) {
            deptPrint(deptList);
        }

    }

    public static void deptPrint(Dept dept) {
        System.out.println(dept.toString());
    }

    public static void deptPrint(List<Dept> deptList) {
        for(Dept dept : deptList) {
            System.out.println(dept.toString());
        }
    }

}

```