Nghi Nguyen (010872316)
CS 157B-04

1. Database used

RDBMS	MySql + JDBC connector
Database name	cs157b_project1
Server	127.0.0.1 or localhost

```
final static String DB_URL = "jdbc:mysql://127.0.0.1:3306/cs157b_project1?useSSL=false";
final static String DB_NAME = "cs157b_project1";
final static String DB_USER = "root";
final static String DB_PASSWORD = "123";
```

2. JDBC driver

Mysql-connector-java-5.1.45-bin.jar

JDBC Connector	Version 5.1.45
Set up driver	`public static int setUpDriver(){` `  int connect = 0;` `  try {` `  Class.forName("com.mysql.jdbc.Driver");` `  System.out.println("JDBC set up successfull");` `  connect=1; //` `  }catch (ClassNotFoundException e) {` `  connect=0;` `  }` `  return connect;` `}`

Connect to DB	`public static Connection connectDB(String url, String user, String pass){` `  Connection conn = null;` `  try {` `  conn = DriverManager.getConnection(url, user, pass);` `  } catch (SQLException e) {` `  // failed to establish connection` `  System.out.println("SQLException: " + e.getMessage());` `  System.out.println("SQLState: " + e.getSQLState());` `  }` `  return conn;` `}`
Disconnect	`public static void disConnectDB(Connection conn){` `  if(null!=conn){` `  try {` `  conn.close();` `  } catch (SQLException e) {` `  // TODO Auto-generated catch block` `  e.printStackTrace();` `  System.out.println("Disconnect failed.");` `  }` `  }` `}`

3. The table schema definition

Table departments	departments(deptId: int(11), deptAbbrev: char(8), deptName: varchar(50))

```
public static boolean createDepartmentTable(Connection conn){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "create table if not exists departments ( "+
                            "deptId INT AUTO_INCREMENT, "+
                            "deptAbbrev char(8) NOT NULL, "+
                            "deptName varchar(50) NOT NULL, "+
                            "primary key(deptId) "+
                       ")ENGINE=InnoDB DEFAULT CHARSET=latin1; ";

        try {
            stmt = conn.prepareStatement(query);
            stmt.executeUpdate();

            success=true;
            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

Table employees	employees(<u>empId</u>: int(11), empName: varchar(150), deptId: int(11), salary: int(11), status: varchar(50), education: varchar(50))
Foreign key	deptId references to deptId field of table department, set null on deletion

```
public static boolean createEmployeeTable(Connection conn){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "CREATE TABLE if not exists employees ( "+
                            "empId int(11) NOT NULL AUTO_INCREMENT, "+
                            "empName varchar(150) NOT NULL, "+
                            "deptId int(11) DEFAULT NULL,"+
                            "salary int(11) DEFAULT NULL, "+
                            "status varchar(50) DEFAULT 'Active', "+
                            "education varchar(50) DEFAULT NULL, "+
                            "PRIMARY KEY (empId), "+
                            "KEY FK_deptId (deptId), "+
                            "CONSTRAINT FK_deptId FOREIGN KEY (deptId) " +
                                "REFERENCES departments (deptId)  ON DELETE SET NULL "+
                       ") ENGINE=InnoDB DEFAULT CHARSET=latin1; ";

        try {
            stmt = conn.prepareStatement(query);
            stmt.executeUpdate();

            success=true;
            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

4 Insert data

Tables Departments

```java
public static boolean insertDepartments(Connection conn){
    boolean success = false;
    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "INSERT INTO `departments` (`deptAbbrev`, `deptName`) "+
                       "VALUES "+
                            "('ACCT','Acountant'), "+
                            "('IT','Technology'), "+
                            "('HR','Human Reources'), "+
                            "('TRANS','Transportation'), "+
                            "('ENGR','Engineering'), "+
                            "('MARK','Marketing'), "+
                            "('FOOD','Food services'), "+
                            "('SALE','Sales'), "+
                            "('FI','Finance'), "+
                            "('ADS','Advertisment'), "+
                            "('SUPP','Supports'), "+
                            "('DELI','Delivery'), "+
                            "('QA','Tesings'), "+
                            "('JOUR','Journalism'), "+
                            "('HEALTH','Health management'), "+
                            "('PR','Public Relation'); ";

        try {
            stmt = conn.prepareStatement(query);
            int count = stmt.executeUpdate();
            System.out.println(count + " rows updated");
            if(count>0){
                success=true;
            }
            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }
    return success;
```

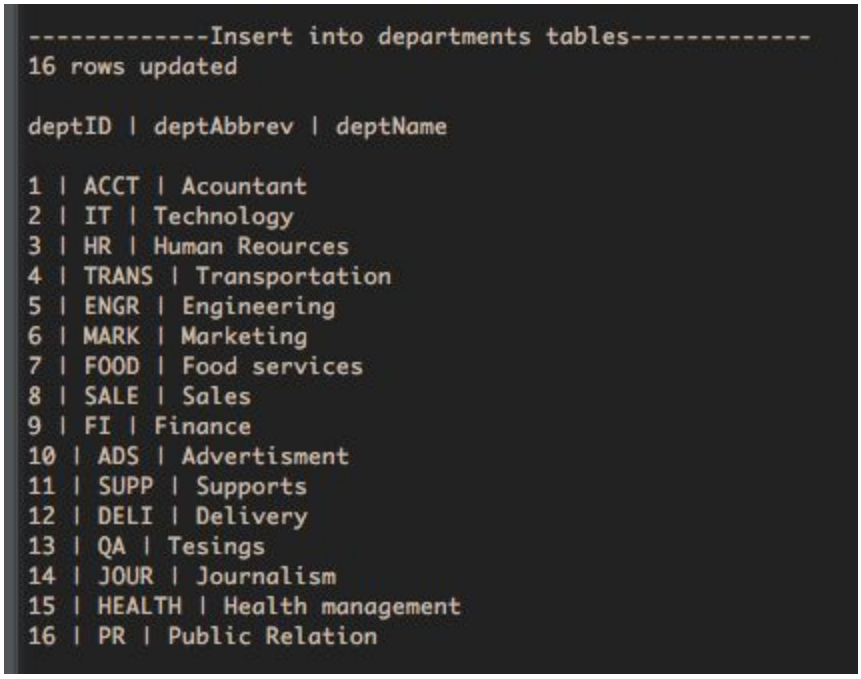
Table employees

```
public static boolean insertEmployees(Connection conn){
    boolean success = false;
    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "INSERT INTO `employees` (`empName`, `deptId`, `salary`, `education`) "+
                       "VALUES "+
                            "('Ann Lee',1, 3500, 'Certification'), "+
                            "('Ryan Gonzalez',1, 4000, 'BA'), "+
                            "('Wanda Delgado',4, 5000, 'BA'), "+
                            "('Kate Sharp',12, 5000, 'MA'), "+
                            "('Rafael Flores',1, 10000, 'PhD'), "+
                            "('Gilbert Mendoza',5, 9000, 'MA'), "+
                            "('Ronald Obrien',5, 4500, 'AA'), "+
                            "('Alvin Miller',2, 4870, 'AA'), "+
                            "('Eugene Ruiz',1, 11000, 'BA'), "+
                            "('Phillip Adkins',3, 8000,'BA'), "+
                            "('Arnetta Mccain',15, 5000, 'BA'), "+
                            "('Johanne Tobin',14, 4000, 'AA'), "+
                            "('Trent Rohr',13, 4600, 'Certification'), "+
                            "('Dick Whitman',6, 12500, 'BA'), "+
                            "('Ronni Dwyer',7, 3400, 'PhD'), "+
                            "('Doreatha Lugo',8, 3000, 'AA'), "+
                            "('Gwenn Cheung',9, 9500, 'Certification'); ";
        try {
            stmt = conn.prepareStatement(query);
            int count = stmt.executeUpdate();
            System.out.println(count + " rows updated");
            if(count>0){
                success=true;
            }
            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

5. All queries

1	Show all records of departments table select * from departments;

```java
public static boolean showDepartments(Connection conn){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "select * from departments; ";

        try {
            stmt = conn.prepareStatement(query);
            ResultSet rs = stmt.executeQuery();

            System.out.println("\ndeptID | deptAbbrev | deptName\n");
            while(rs.next()){
                int deptId = rs.getInt("deptId");
                String deptAbbrev = rs.getString("deptAbbrev");
                String deptName = rs.getString("deptName");
                System.out.println(deptId+" | "+deptAbbrev+" | "+deptName);

            }

            success = true;

            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

Result	

2	Shows all records of table employees Select * from employees

```java
public static boolean showEmployees(Connection conn){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "select * from employees; ";

        try {
            stmt = conn.prepareStatement(query);
            ResultSet rs = stmt.executeQuery();

            System.out.println("\nempId | empName | deptId | salary | status | education\n");
            while(rs.next()){
                int empId = rs.getInt("empId");
                int deptId = rs.getInt("deptId");
                String empName = rs.getString("empName");
                String salary = rs.getString("salary");
                String status = rs.getString("status");
                String education = rs.getString("education");
                System.out.println(empId+" | "+empName+" | "+deptId+" | "+salary+" | "+status+" | "+education);

            }

            success = true;

            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }
    return success;
}
```

Result

```
------------Insert into employees tables------------
17 rows updated

empId | empName | deptId | salary | status | education

1 | Ann Lee | 1 | 3500 | Active | Certification
2 | Ryan Gonzalez | 1 | 4000 | Active | BA
3 | Wanda Delgado | 4 | 5000 | Active | BA
4 | Kate Sharp | 12 | 5000 | Active | MA
5 | Rafael Flores | 1 | 10000 | Active | PhD
6 | Gilbert Mendoza | 5 | 9000 | Active | MA
7 | Ronald Obrien | 5 | 4500 | Active | AA
8 | Alvin Miller | 2 | 4870 | Active | AA
9 | Eugene Ruiz | 1 | 11000 | Active | BA
10 | Phillip Adkins | 3 | 8000 | Active | BA
11 | Arnetta Mccain | 15 | 5000 | Active | BA
12 | Johanne Tobin | 14 | 4000 | Active | AA
13 | Trent Rohr | 13 | 4600 | Active | Certification
14 | Dick Whitman | 6 | 12500 | Active | BA
15 | Ronni Dwyer | 7 | 3400 | Active | PhD
16 | Doreatha Lugo | 8 | 3000 | Active | AA
17 | Gwenn Cheung | 9 | 9500 | Active | Certification
```

3	Join 2 tables employees and departments

```
public static boolean DepartmentsJoinEmployees(Connection conn){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "select employees.empId, employees.empName, employees.salary, " +
                               "departments.deptAbbrev,  departments.deptName "+
                       "from employees, departments "+
                       "where employees.deptId=departments.deptId; ";

        try {
            stmt = conn.prepareStatement(query);
            ResultSet rs = stmt.executeQuery();

            System.out.println("\nempId | empName | salary | deptAbbre | deptName\n");
            while(rs.next()){
                int empId = rs.getInt("empId");
                String empName = rs.getString("empName");
                String salary = rs.getString("salary");
                String deptAbbrev = rs.getString("deptAbbrev");
                String deptName = rs.getString("deptName");
                System.out.println(empId+" | "+empName+" | "+salary+" | "+deptAbbrev+" | "+deptName);

            }

            success = true;

            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

result	

```
-------------Join: Employee and department-------------

empId | empName | salary | deptAbbre | deptName

1 | Ann Lee | 3500 | ACCT | Acountant
2 | Ryan Gonzalez | 4000 | ACCT | Acountant
3 | Wanda Delgado | 5000 | TRANS | Transportation
4 | Kate Sharp | 5000 | DELI | Delivery
5 | Rafael Flores | 10000 | ACCT | Acountant
6 | Gilbert Mendoza | 9000 | ENGR | Engineering
7 | Ronald Obrien | 4500 | ENGR | Engineering
8 | Alvin Miller | 4870 | IT | Technology
9 | Eugene Ruiz | 11000 | ACCT | Acountant
10 | Phillip Adkins | 8000 | HR | Human Reources
11 | Arnetta Mccain | 5000 | HEALTH | Health management
12 | Johanne Tobin | 4000 | JOUR | Journalism
13 | Trent Rohr | 4600 | QA | Tesings
14 | Dick Whitman | 12500 | MARK | Marketing
15 | Ronni Dwyer | 3400 | FOOD | Food services
16 | Doreatha Lugo | 3000 | SALE | Sales
17 | Gwenn Cheung | 9500 | FI | Finance
```

4	Find total number of employees in each department

```
public static boolean TotalEmployeesinDepartments(Connection conn){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "select departments.deptAbbrev, departments.deptName, empCount.totalEmployees "+
                       "from departments, "+
                             "(select departments.`deptId`, count(empId) as totalEmployees "+
                              "from employees, departments where employees.deptId=departments.deptId "+
                              "group by departments.deptId "+
                              "order by departments.deptName asc) empCount "+
                       "where departments.deptId=empCount.deptId; ";

        try {
            stmt = conn.prepareStatement(query);
            ResultSet rs = stmt.executeQuery();

            System.out.println("\ndeptAbbre | deptName | totalEmployees\n");
```

```
            while(rs.next()){

                String deptAbbrev = rs.getString("deptAbbrev");
                String deptName = rs.getString("deptName");
                int totalEmployees = rs.getInt("totalEmployees");
                System.out.println(deptAbbrev+" | "+deptName+" | "+totalEmployees);

            }

            success = true;

            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }
```

Result

```
-------------find how many employees in each department-------------

deptAbbre | deptName | totalEmployees

ACCT | Acountant | 4
IT | Technology | 1
HR | Human Reources | 1
TRANS | Transportation | 1
ENGR | Engineering | 2
MARK | Marketing | 1
FOOD | Food services | 1
SALE | Sales | 1
FI | Finance | 1
DELI | Delivery | 1
QA | Tesings | 1
JOUR | Journalism | 1
HEALTH | Health management | 1
```

5	Find top 5 highest paid employees

```java
public static boolean top5HighestPaid(Connection conn){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "select employees.empName, employees.salary "+
                       "from employees "+
                       "ORDER BY employees.salary desc "+
                       "limit 5; ";

        try {
            stmt = conn.prepareStatement(query);
            ResultSet rs = stmt.executeQuery();

            System.out.println("\nempName | salary\n");
            while(rs.next()){

                String empName = rs.getString("empName");

                int salary = rs.getInt("salary");
                System.out.println(empName+" | "+salary);

            }

            success = true;

            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

Result

```
------------find top 5 highest paid employees------------

empName | salary

Dick Whitman | 12500
Eugene Ruiz | 11000
Rafael Flores | 10000
Gwenn Cheung | 9500
Gilbert Mendoza | 9000
```

6	Sort employee by name Select * from employees order by empName asc/desc

```
public static boolean sortEmployeesByName(Connection conn, String dir){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "select * from employees order by empName " + dir ;

        try {
            stmt = conn.prepareStatement(query);
            ResultSet rs = stmt.executeQuery();

            System.out.println("\nempId | empName | deptId | salary | status | education\n");
            while(rs.next()){
                int empId = rs.getInt("empId");
                int deptId = rs.getInt("deptId");
                String empName = rs.getString("empName");
                String salary = rs.getString("salary");
                String status = rs.getString("status");
                String education = rs.getString("education");
                System.out.println(empId+" | "+empName+" | "+deptId+" | "+salary+" | "+status+" | "+education);

            }

            success = true;

            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

Result	

7	Find every employee who earn >= 8000 in a department Accountant

```
public static boolean findEmpByDeptAndSalary(Connection conn, int salary, String deptAbbr){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "select * from employees, departments "+
                       "where employees.deptId = departments.deptId " +
                              "and employees.salary >= ? and deptAbbrev = ?; ";

        try {
            stmt = conn.prepareStatement(query);
            stmt.setInt(1, salary);
            stmt.setString(2, deptAbbr);
            ResultSet rs = stmt.executeQuery();

            System.out.println("\nempId | empName | deptId | salary | status | education\n");
            while(rs.next()){
                int empId = rs.getInt("empId");
                int deptId = rs.getInt("deptId");
                String empName = rs.getString("empName");
                String sa = rs.getString("salary");
                String status = rs.getString("status");
                String education = rs.getString("education");
                System.out.println(empId+" | "+empName+" | "+deptId+" | "+sa+" | "+status+" | "+education);

            }

            success = true;

            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
```

Result

```
-------------Find employees who earn more than 8000 in ACCT department----------

empId | empName | deptId | salary | status | education

5 | Rafael Flores | 1 | 10000 | Active | PhD
9 | Eugene Ruiz | 1 | 11000 | Active | BA
```

8	Delete a department by deptId

```
public static boolean deleteDepartment(Connection conn, int deptId){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "delete from departments "+
                       "where deptId=?; ";


        try {
            stmt = conn.prepareStatement(query);

            stmt.setInt(1, deptId);
            int count = stmt.executeUpdate();

            success = true;
            System.out.println(count+" rows affected");


            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

Result

```
------------Delete 'Public Relation' depId=16 off department------------

Before-----

deptID | deptAbbrev | deptName

1 | ACCT | Acountant
2 | IT | Technology
3 | HR | Human Reources
4 | TRANS | Transportation
5 | ENGR | Engineering
6 | MARK | Marketing
7 | FOOD | Food services
8 | SALE | Sales
9 | FI | Finance
10 | ADS | Advertisment
11 | SUPP | Supports
12 | DELI | Delivery
13 | QA | Tesings
14 | JOUR | Journalism
15 | HEALTH | Health management
16 | PR | Public Relation

After-----
1 rows affected

deptID | deptAbbrev | deptName

1 | ACCT | Acountant
2 | IT | Technology
3 | HR | Human Reources
4 | TRANS | Transportation
5 | ENGR | Engineering
6 | MARK | Marketing
7 | FOOD | Food services
8 | SALE | Sales
9 | FI | Finance
10 | ADS | Advertisment
11 | SUPP | Supports
12 | DELI | Delivery
13 | QA | Tesings
14 | JOUR | Journalism
15 | HEALTH | Health management
```

9	Show an employee by id
	`` public static boolean showEmployeeById(Connection conn, int empId){ boolean success = false; if(conn!=null){ PreparedStatement stmt = null; String query = "select * "+ "from employees "+ "where employees.`empId`= ?; "; try { stmt = conn.prepareStatement(query); stmt.setInt(1, empId); ResultSet rs = stmt.executeQuery(); System.out.println("\nempId | empName | salary | status | education | deptId \n"); while(rs.next()){ int eId = rs.getInt("empId"); int salary = rs.getInt("salary"); String deptId = rs.getString("deptId"); String empName = rs.getString("empName"); String status = rs.getString("status"); String education = rs.getString("education"); System.out.println(eId+" | "+empName+" | "+salary+" | "+status+" | "+education+" | "+deptId); } success = true; if(stmt!=null){ stmt.close(); } } catch (SQLException e) { // TODO Auto-generated catch block e.printStackTrace(); success=false; } finally{ Connections.disConnectDB(conn); } ```
Result	``` empId | empName | salary | status | education | deptId 10 | Phillip Adkins | 8000 | Active | BA | 3 ```

10	Set inactive status to an employee. The deptId will be set to null
	

```
public static boolean updateSetInActive(Connection conn,  int empId){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "update employees "+
                       "set status = 'Inactive', deptId= NULL "+
                       "where empId=?; ";


        try {
            stmt = conn.prepareStatement(query);
            stmt.setInt(1, empId);

            int count = stmt.executeUpdate();

            success = true;
            System.out.println(count+" rows affected");


            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

```
------------Set inactive status to 1 employee------------

Before-----

empId | empName | salary | status | education | deptId

10 | Phillip Adkins | 8000 | Active | BA | 3

After-----
1 rows affected

empId | empName | salary | status | education | deptId

10 | Phillip Adkins | 8000 | Inactive | BA | null
```

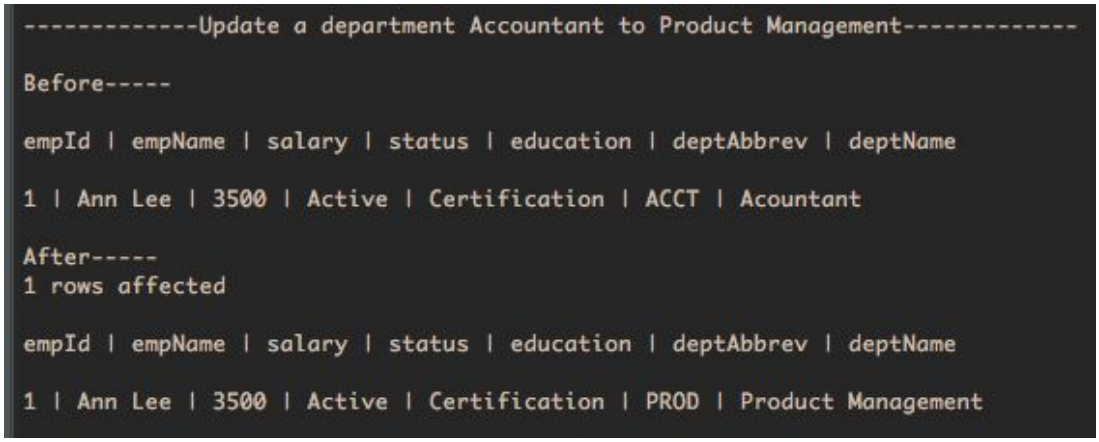
11	Show an employee information with department information

```
public static boolean showEmployeeandDeptById(Connection conn, int empId){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "select employees.`empId`, employees.`empName`, employees.`salary`, " +
                       "employees.`status`, employees.`education`, departments.`deptAbbrev`, "+
                       "departments.`deptName` "+
                   "from employees, departments "+
                   "where  employees.`deptId` = departments.`deptId` and employees.`empId`= ?; ";


        try {
            stmt = conn.prepareStatement(query);
            stmt.setInt(1, empId);
            ResultSet rs = stmt.executeQuery();

            System.out.println("\nempId | empName | salary | status | education | deptAbbrev | deptName\n");
       while(rs.next()){

           String deptAbbrev = rs.getString("deptAbbrev");
           String deptName = rs.getString("deptName");
           int eId = rs.getInt("empId");
           int salary = rs.getInt("salary");
           String empName = rs.getString("empName");
           String status = rs.getString("status");
           String education = rs.getString("education");
           System.out.println(eId+" | "+empName+" | "+salary+" | "+status+" | "+education+" | "+deptAbbrev+" | "+deptName);

       }

       success = true;

       if(stmt!=null){
           stmt.close();
       }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        success=false;
    } finally{
        Connections.disConnectDB(conn);
    }

}else {
    success = false;
}

return success;
```

Result	empId | empName | salary | status | education | deptAbbrev | deptName 1 | Ann Lee | 3500 | Active | Certification | ACCT | Acountant

12	Update a department given a deptId

```
public static boolean updateDepartment(Connection conn, String abbr, String deptName, int deptId){
    boolean success = false;

    if(conn!=null){
        PreparedStatement stmt = null;
        String query = "update departments "+
                       "set deptAbbrev = ?, deptName= ? "+
                       "where deptId=?; ";


        try {
            stmt = conn.prepareStatement(query);
            stmt.setString(1, abbr);
            stmt.setString(2, deptName);
            stmt.setInt(3, deptId);
            int count = stmt.executeUpdate();

            success = true;
            System.out.println(count+" rows affected");


            if(stmt!=null){
                stmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            success=false;
        } finally{
            Connections.disConnectDB(conn);
        }

    }else {
        success = false;
    }

    return success;
}
```

Result	 ----------Update a department Accountant to Product Management---------- Before----- empId | empName | salary | status | education | deptAbbrev | deptName 1 | Ann Lee | 3500 | Active | Certification | ACCT | Acountant After----- 1 rows affected empId | empName | salary | status | education | deptAbbrev | deptName 1 | Ann Lee | 3500 | Active | Certification | PROD | Product Management