**DBMS ASSIGNMENT**

**1.Employee information**

All the data is atomic. Therefore, the table is in 1NF.

**Employee table**

| EmployeeID | EmployeeName | Salary | HireDate |
| --- | --- | --- | --- |
| 1 | John smith | 50000 | 2022-01-15 |
| 2 | Alice Brown | 60000 | 2022-02-20 |
| 3 | Mark Johnson | 55000 | 2022-03-10 |

EmployeeName, Salary, HireDate are fully dependent on EmployeeID .So the table is split into Emplyee table therefore, eliminating partial dependencies

Now a department table is created

**Department table**

| DepartmentID | Department |
| --- | --- |
| D1 | HR |
| D2 | IT |
| D3 | Sales |

Now by mapping Employee ID from Employee table and Department Id from Department table and since ManagerID is a standalone entity we could a normalized table

**EmployeeDepartmentMapping** table

| employeeID | departmentID | ManagerID |
| --- | --- | --- |
| 1 | D1 | 101 |
| 2 | D2 | 102 |
| 3 | D3 | 101 |

**2. Training programs**

Removing the partial dependency by creating a new table called programs where programName and trainer are functionally dependent on programID.

| programID | programName | Trainer |
| --- | --- | --- |
| 1 | Java Fundamentals | John Smith |
| 2 | Project Management | Sarah White |
| 3 | Sales Techniques | Mark Johnson |

EmployeeID is dependent on programName and programName is dependent on programID which causes the transitive dependency. To remove the transitive dependency we need to create a new table for employee which has employeeID, department and employee name as fields.

| employeeId | department | employeename |
| --- | --- | --- |
| 101 | IT | Alice Brown |
| 102 | HR | Bob Green |
| 103 | Sales | Charlie Black |

| programID | employeeID | date |
| --- | --- | --- |
| 1 | 101 | 2022-03-01 |
| 2 | 102 | 2022-03-10 |
| 3 | 103 | 2022-03-20 |

.

Change in employee table.

| employeeId | employeename |
| --- | --- |
| 101 | Alice Brown |
| 102 | Bob Green |
| 103 | Charlie Black |

Program table

| programID | programName | Trainer |
| --- | --- | --- |
| 1 | Java Fundamentals | John Smith |
| 2 | Project Management | Sarah White |
| 3 | Sales Techniques | Mark Johnson |

This would be a new department table which will help to resolve the deletion anomaly.

| departmentID | department Name |
| --- | --- |
| d1 | IT |
| d2 | HR |
| d3 | Sales |

programEmployeeDepartmentMapping table

| programID | employeeID | departmentID | date |
| --- | --- | --- | --- |
| 1 | 101 | d1 | 2022-03-01 |
| 2 | 102 | d2 | 2022-03-10 |
| 3 | 103 | d3 | 2022-03-20 |

**3. Customer orders**

The initial table is already in 1NF form because no multivalued attributes are there and all the data are atomic.

Removing the partial dependency from the table by creating a new table for product using the fields productID, productName, unitPrice. Here the productName and unitPrice are functionally dependent on productID. This new table removes partial dependency from the table.

| productID | productName | unitPrice |
| --- | --- | --- |
| 101 | Laptop | 800 |
| 102 | Smartphone | 500 |
| 103 | Printer | 200 |

In the new table which has fields productID, orderID, customerName, qty, totalAmount, orderDate.

| productID | orderId | customerName | qty | totalAmount | orderDate |
| --- | --- | --- | --- | --- | --- |

customerName is fully dependent on orderID and orderID is fully dependent on productID which causes the transitive dependency. So to remove the transitive dependency from the table we need to create a new table order which has orderID, customerName, qty, totalAmount, orderDate. Thus removing transitive dependency.

| orderId | customerName | qty | totalAmount | orderDate |
| --- | --- | --- | --- | --- |
| 1 | John Doe | 2 | 1600 | 2022-01-15 |
| 2 | Jane Smith | 1 | 500 | 2022-02-20 |
| 3 | John Doe | 1 | 200 | 2022-03-10 |

The final table is called the orderProductMapping where the products are mapped to orders in which the products are purchased.

order Product Mapping table

| orderid | productid |
| --- | --- |
| 1 | 101 |
| 2 | 102 |
| 3 | 103 |

4. Stress management

Since the initial table is already in the 1NF form.

We remove the partial dependency from the initial table by removing the firstName, lastName and employeeID as employee tables where firstName and lastName are fully dependent on employeeID.

| employeeeID | firstName | lastName |
| --- | --- | --- |
| 101 | Sarah | White |
| 102 | Bob | Green |
| 103 | Charlie | Black |
| 104 | David | Miller |
| 105 | Jane | Doe |

Again hoursOfWork and BreaksTaken are functionally dependent on employeID so we create a new table called workingHours.

| employeeID | hoursOfWork | BreaksTaken |
| --- | --- | --- |
| 101 | 45 | 3 |
| 102 | 50 | 2 |
| 103 | 40 | 4 |
| 104 | 48 | 1 |
| 105 | 42 | 3 |

StressLevel, PhysicalActivity and counsellingSession are associated with employeeID. So we are again creating a new table called stressManagement.

| employeeID | stressLevel | PhysicalActivity | counsellingSesssion |
| --- | --- | --- | --- |
| 101 | moderate | yoga | 2 |
| 102 | high | joggin | 1 |
| 103 | low | meditation | 3 |
| 104 | high | gym | 2 |
| 105 | moderate | walking | 1 |

Now the table is normalized.

5. Flea market

The initial table is already in 1NF. Removing the partial dependency by creating a new table called sellers which has the sellerID, sellerName and also the location. Here the sellerName and location is fully dependent on the key sellerID. Thus remove the partial dependency. Hence the normalized version is given below. One table is for the items and the other table is called seller, where the sellerID in the items table is referenced from the seller table using foreign key constraints.

| itemID | sellerId | itemName | category | price | quantity | description | condition | dateListed |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 101 | s1 | Vintage chair | Furniture | 50.00 | 2 | Beautiful vintage  chair, excellent  condition | Lik new | 2022-01-15 |
| 102 | s2 | Antique clock | Home Decor | 80.001 | 1 | Authentic  antique clock  with Roman  numerals | good | 2022-02-20 |
| 103 | s3 | Vinyl Records | Music | 15.00 | 10 | Various artists  and genres, in  good condition | used | 2022-03-10 |
| 104 | s4 | Vintage Jewelery | Accessories | 35.00 | 5 | Assorted vintage  jewelry pieces,  unique designs | excellent | 2022-04-05 |
| 105 | s5 | Retro Camera | Electronics | 60.00 | 1 | Vintage Polaroid  camera with  original case | good | 2022-05-15 |

| sellerID | sellerName | location |
| --- | --- | --- |
| s1 | John’s Treasure | Booth 15, section A |
| s2 | Alice’s Finds | Stall 8, section B |
| s3 | Mark’s collectibles | Booth 20, Section C |
| s4 | Emma’s Treasure | Stall 12, Section D |
| s5 | Robert’s Find | Booth 5, Section A |

**6. Learning management system**

Removing partial dependency. Creating a course table where courseName and credits are fully dependent on CID.

| CID | courseName | credits |
| --- | --- | --- |
| 101 | Introduction to biology | 3 |
| 102 | Programming in python | 4 |
| 103 | Financial accounting | 3 |
| 104 | English literature | 3 |
| 105 | Web development fundamentals | 4 |

Removing transitive dependency, that is instructor is functionally dependent on courseName and courseName is fully dependent on CID.An Instructor table is created. The new table is shown below

| InstructorID | instructor |
| --- | --- |
| c1 | Prof. smith |
| c2 | Prof. brown |
| c3 | Prof. green |
| c4 | Prof. white |
| c5 | Prof. Black |

Removing transitive dependency, that is department is functionally dependent on instructorID and instructorID is fully dependent on CID. So we are creating another table for department to remove the partial dependency. The new table is shown below

| departmentID | department |
| --- | --- |
| d1 | science |
| d2 | Computer science |
| d3 | finance |
| d4 | humanities |
| d5 | IT |

| CID | instructorID | departmentID | Enrolled students | Start date | End date | location | availability |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 101 | c1 | d1 | 25 | 2022-01-15 | 2022-05-10 | Room 101 | open |
| 102 | c2 | d2 | 30 | 2022-02-20 | 2022-06-15 | Lab 3, building B | closed |
| 103 | c3 | d3 | 20 | 2022-03-10 | 2022-07-05 | Room 201 | open |
| 104 | c4 | d4 | 22 | 2022-04-05 | 2022-08-20 | Room 301 | open |
| 105 | c5 | d5 | 28 | 2022-05-15 | 2022-09-25 | Lab 2, building A | closed |