Database PROJECT

Robert Gabriel R00102430

Table of Contents

[Part 1 2](#_Toc373731725)

[Using the above Functional Dependencies, derive 2NF tables. 3](#_Toc373731726)

[Using the above Functional Dependencies and your answer to part (a) derive 3NF tables. 4](#_Toc373731727)

[Using the above Functional Dependencies and your answer to part (a) & (b) derive BCNF tables. 5](#_Toc373731728)

[Part 2 6](#_Toc373731729)

[Identify the candidate keys. 7](#_Toc373731730)

[Normalize to 2NF 7](#_Toc373731731)

[Normalize to 3NF 8](#_Toc373731732)

[Normalize to BCNF 9](#_Toc373731733)

[Part 3 10](#_Toc373731734)

[List the functional dependencies and normalize this relation into BCNF. 11](#_Toc373731735)

# Part 1

Consider the following Functional Dependencies:

|  |
| --- |
| CustNo → CustBal, CustDiscount |
| OrdNo → CustNo, ShipAddr, OrderDate |
| ItemNo → ItemDesc |
| ItemNo, PlantNo → ReorderPoint, QtyOnHand |
| OrderNo, ItemNo → LineNo, QtyOrdered, QtyOutstanding |
| OrderNo, LineNo → ItemNo, QtyOrdered, QtyOutstanding |

## 

## Using the above Functional Dependencies, derive 2NF tables.

The entity under consideration should already be in the 1NF and all attributes within the entity should depend solely on the entity's unique identifier

First we need to add CustNo → CustBal, CustDiscount and OrdNo → CustNo, ShipAddr, OrderDate.

1) Orderno→ shipaddr orderdate,custno, custbal , custdiscount

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Orderno | Shipaddr | Orderdate | Curstno | Crustbal | Custdiscount |
|  |  |  |  |  |  |

2) ItemNo → ItemDesc

|  |  |
| --- | --- |
| Itemno | Itemdesc |
|  |  |

3) OrderNo, ItemNo → LineNo, QtyOrdered, QtyOutstanding

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Orderno | Itemno | Lineno | QtyOrdered | QtyOutstanding |
|  |  |  |  |  |

4) ItemNo, PlantNo → ReorderPoint, QtyOnHand

|  |  |  |  |
| --- | --- | --- | --- |
| ItemNo | PlantNo | ReorderPoint | QtyOnHand |
|  |  |  |  |

## Using the above Functional Dependencies and your answer to part (a) derive 3NF tables.

“The entity should already be in the 2NF and no column entry should be dependent on any other entry (value) other than the key for the table.  
If such an entity exists, move it outside into a new table. “

In order to express the same facts without violating 3NF, I split the table into two:

CustNo → CustBal, CustDiscount

|  |  |  |
| --- | --- | --- |
| CustNo | CustBal, | CustDiscount |
|  |  |  |

OrdNo → CustNo, ShipAddr, OrderDate

|  |  |  |  |
| --- | --- | --- | --- |
| OrdNo | CustNo | ShipAddr, | OrderDate |
|  |  |  |  |

ItemNo → ItemDesc

|  |  |
| --- | --- |
| Itemno | Itemdesc |
|  |  |

OrderNo, ItemNo → LineNo, QtyOrdered, QtyOutstanding

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Orderno | Itemno | Lineno | QtyOrdered | QtyOutstanding |
|  |  |  |  |  |

ItemNo, PlantNo → ReorderPoint, QtyOnHand

|  |  |  |  |
| --- | --- | --- | --- |
| ItemNo | PlantNo | ReorderPoint | QtyOnHand |
|  |  |  |  |

## Using the above Functional Dependencies and your answer to part (a) & (b) derive BCNF tables.

The database should be in 3NF and all tables can have only one primary key.

The Primary keys I would pick are the following, OrderNo ItemNo.

# Part 2

Consider the following relation:

Shipping (ShipName, ShipType, VoyageID, Cargo, Port, Date)

Hint: Date is the date the ship arrives in the given Port

With the functional dependencies:

ShipName -> ShipType

VoyageID -> ShipName, Cargo

ShipName, Date -> VoyageID, Port

## Identify the candidate keys.

What does *Candidate Key* mean?

“A candidate key is a column, or set of columns, in a table that can uniquely identify any database record without referring to any other data. Each table may have one or more candidate keys, but one candidate key is special, and it is called the primary key.”

Answer: Key is ShipName and Date

## Normalize to 2NF

Starting with the initial relation:

Shipping (ShipName, ShipType, VoyageID, Cargo, Port, Date)

The partial key of ShipName alone can determine ShipType.

So I normalize it to:

SHIPS (ShipName, ShipType)

|  |  |
| --- | --- |
| ShipName | ShipType |
|  |  |
|  |  |

VOYAGES (ShipName, VoyageID, Cargo, Port, Date)

|  |  |  |  |
| --- | --- | --- | --- |
| ShipName | Date | VoyageID | Port |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| VoyageID | ShipName | Cargo |
|  |  |  |

## Normalize to 3NF

As in the last page (2nf), VOYAGES has a transitive dependency:

ShipName, Date → VoyageID

VoyageId → Cargo

So I normalized VOYAGES:

SHIPPORTS (ShipName, VoyageID, Port, Date)

ShipName, Date→VoyageID, Port

|  |  |  |  |
| --- | --- | --- | --- |
| ShipName | Date | VoyageID | Port |

VoyageID→ShipName

|  |  |
| --- | --- |
| VoyageID | ShipName |

CARGO (VoyageID, Cargo)

VoyageId→Cargo

|  |  |
| --- | --- |
| VoyageID | Cargo |

SHIPS( ShipName, ShipType)

ShipName → ShipType

|  |  |
| --- | --- |
| ShipName | ShipType |
|  |  |

## Normalize to BCNF

SHIPPORTS is not in BCNF since it has VoyageID as a determinant key but VoyageID is not a candidate key.

SHIPDATES (ShipName, Port, Date)

ShipName, Date→Port

|  |  |  |
| --- | --- | --- |
| ShipName | Date | Port |

SHIPVOYAGE (VoyageID, ShipName)

VoyageID→ShipName

|  |  |
| --- | --- |
| VovageID | ShipName |

CARGO (VoyageID, Cargo)

VoyageId→Cargo

|  |  |
| --- | --- |
| VoyageID | Cargo |

SHIPS ( ShipName, ShipType)

ShipName →ShipType

|  |  |
| --- | --- |
| ShipName | ShypType |

# Part 3

Given the following relation and example data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PartNumber | Description | Supplier | SupplierAddress | Price |
| 10010 | 20 GB Disk | Seagate | Cuppertino, CA | $100 |
| 10010 | 20 GB Disk | IBM | Armonk, NY | $90 |
| 10220 | 256 MB RAM card | Kensington | San Mateo, CA | $220 |
| 10220 | 256 MB RAM card | IBM | Armonk, NY | $290 |
| 10220 | 256 MB RAM card | Sun Microsystems | Palo Alto, CA | $310 |
| 10440 | 17" LCD Monitor | IBM | Armonk, NY | $2,100 |

## List the functional dependencies and normalize this relation into BCNF.

a) PartNumber →Description

b) PartNumber, Supplier →Price

1. Supplier→SupplierAddress

1NF: I picked partNumber, Supplier as the key so I am in 1NF

2NF: We have a partial key dependency in that

Supplier →SupplierAddress

So I normalized:

AB (PartNumber, Description, Supplier, Price)

I added tables a and b here

PartNumber →Description

PartNumber, Supplier→Price

C (Supplier, SupplierAddress)

Supplier→SupplierAddress

B (PartNumber, Supplier, Price)

PartNumber, Supplier→Price

A (PartNumber, Description)

I normalized table AB again

PartNumber → Description

C (Supplier, SupplierAddress)

Supplier → SupplierAddress