Name: Nadella, Aakarsh **NN:70**

MEMO: #2

According to the data given in Memo1, we get the following results for first table

	ApartmentID	Number	NumberRents	ComplexID	RenterID	Name	StartDate	Rent	PaymentNumber	DateDue	DatePaid	Amount
ApartmentID	-				2&3	2&3	2&3	2&3	1&2	1&2	1&2	2&3
Number	4&5	-		4&5	2&3	2&3	2&3	2&3	1&2	1&2	1&2	2&3
NumberRents	4&5	1&9	-	1&4	1&3	1&3	1&3	1&3	1&2	1&2	1&2	1&3
ComplexID	1&4	1&4	1&4	-	1&3	1&3	1&3	1&3	1&2	1&2	1&2	1&3
RenterID	5&6	5&6		5&6	-		5&6	5&6	1&2	1&2	1&2	8&9
Name	5&6	5&6		5&6		-	5&6	5&6	1&2	1&2	1&2	8&9
StartDate	1&4	1&4	1&4		1&4	1&4	-	3&6	1&2	1&2	1&2	7&8
Rent	1&4	1&4	1&4	3&5	1&4	1&4	1&4	-	1&2	1&2	1&2	8&9
PaymentNumber									-			
DateDue	1&4	1&4	1&4	2&5	1&4	1&4		3&6	1&4	-	1&4	7&8
DatePaid	3&7	3&7	3&7	3&7	3&7	3&7	3&7	3&7	3&7	3&7	-	3&7
Amount	1&4	1&4	1&4	3&5	1&4	1&4	3&5	3&6		1&2	1&2	-

The dependencies from the above analysis are:

{ApartmentID} → {Number}

{ApartmentID} → {NumberRents}

{ApartmentID} → {ComplexID}

{Number} → {NumberRents}

{RenterID} → {NumberRents}

{RenterID} → {Name}

{Name} → {NumberRents}

{Name} → {RenterID}

{StartDate} → {ComplexID}

{PaymentNumber} → {ApartmentID}

{PaymentNumber} → {Number}

{PaymentNumber} → {NumberRents}

{PaymentNumber} → {ApartmentID}

{PaymentNumber} → {ComplexID}

Name: Nadella, Aakarsh **NN:70**

{PaymentNumber} → {RenterID}

{PaymentNumber} → {Name}

{PaymentNumber} → {StartDate}

{PaymentNumber} → {Rent}

{PaymentNumber} → {DateDue}

{PaymentNumber} → {DatePaid}

{PaymentNumber} → {Amount}

{DateDue} → {StartDate}

{DateDue} → {ComplexID}

Now for Table 2, the analysis is:

	ApartmentID	**Number**	**ComplexID**	**ProspectNumber**	**Name**	**Address**	**Phone**	**StatusCode**
ApartmentID	-			10 & 11	10 & 11	10 & 11	10 & 11	10 & 11
Number	12 & 13	-	12 & 13	10 & 11	10 & 11	10 & 11	10 & 11	10 & 11
ComplexID	10 & 12	10 & 12	-	10 & 11	10 & 11	10 & 11	10 & 11	10 & 11
ProspectNumber	11 & 12	11 & 12		-				
Name	11 & 12	11 & 12			-			
Address	10 & 16	10 & 16	10 & 16	10 & 16	10 & 16	-		10 & 16
Phone	10 & 16	10 & 16	10 & 16	10 & 16	10 & 16		-	10 & 16
StatusCode	11 & 12	11 & 12	11 & 14	11 & 14	11 & 14	11 & 14	11 & 14	-

The dependencies from the above analysis are:

{ApartmentID} → {Number}

{ApartmentID} → {ComplexID}

{ProspectNumber} → {ComplexID}

{ProspectNumber} → {Name}

{ProspectNumber} → {Address}

{ProspectNumber} → {Phone}

{ProspectNumber} → {StatusCode}

{Name} → {ComplexID}

{Name} → {ProspectNumber}

{Name} → {Address}

{Name} → {Phone}

{Name} → {StatusCode}

{Address} → {Phone}

{Phone} → {Address}

The functional dependencies from the Enterprise Statement:

{ApartmentID} → {Number, ComplexID}

{ApartmentID, RenterID} → {Rent}

{ApartmentID, RenterID} → {StartDate}

{ApartmentID} → {Number, ComplexID}

{ApartmentID, RenterID} → {ProspectNumber}

So, based on all FD's above, after removing redundancies using primary key and removing transitive dependencies, the irreducible cover generated is as follows

{ApartmentID} → {Number, ComplexID}

{Number} → {NumberRents}

{RenterID} → {Name}

{RenterID} → {ProspectNumber}

{ApartmentID, RenterID} → {Rent}

{ApartmentID, RenterID} → {StartDate}

{PaymentNumber} → {ApartmentID}

{PaymentNumber} → {RenterID}

{PaymentNumber} → {DateDue}

{PaymentNumber} → {DatePaid}

{PaymentNumber} → {Amount}

{ProspectNumber} → {Name}

{ProspectNumber} → {Phone}

{ProspectNumber} → {StatusCode}

{Phone} → {Address}

Relational Headers derived from irreducible cover is:

{

{ApartmentID} → {Number, ComplexID}

{Number} → {NumberRents}

{RenterID} → {Name, ProspectNumber}

{ApartmentID, RenterID} → {Rent, StartDate}

{PaymentNumber} → {ApartmentID, RenterID, DateDue, DatePaid, Amount}

{ProspectNumber} → {Name, Phone, StatusCode}

{Phone} → {Address}

}

Apartment

ApartmentID (PK)

Number

ComplexID

Renter

RenterID (PK)

Name

StartDate

Rent

ApartmentID (FK)

Payment

PaymentNumber (PK)

DateDue

DatePaid

Amount

ApartmentID (FK)

RenterID (FK)

Prospects

ProspectNumber (PK)

Name

Address

Phone

Waitlist

ProspectNumber (FK)

ApartmentID (FK)

StatusCode