

1.

Happy Supplies Parts Warehouse					
Customer Name: <u>Jeff Peterson</u>		Date: <u>7/1/2024</u>			
Customer Number: <u>H G54587</u>		Time: <u>10:30am</u>			
Customer Type: <u>Consumer</u>		Employee: <u>D. Harrison</u>			
Part Number	Name	Type	Cage Code	Quantity Ordered	Unit Price
10654	Float Control	Plumbing	G413	4	12
10456	Modulator	Electrical	H 433	3	7
10776	Hose Assembly	Plumbing	G413	7	9
10657	Float Assembly	Plumbing	G413	5	10

(a). Assumption 1: each part is sold for same price for each customer, and the only identifier for Unit Price is Part Number. Thus, Part Number is the only identifier for Name, Type, Cagy Code, and Unit Price.

Assumption 2: each employ can only help one customer at a given time at a given date, in order words, he or she cannot help multiple customers simultaneous, so customer (number, date, time) is the only identifier for an order. Each order shows Employee and Quantity Ordered.

Assumption 3: A customer can have more than orders, and can order more than one parts for an order.

(b). From the assumptions above, we can draw the following relationships:

Customer Number → Customer Name, Customer Type

Part Number → Name, Type, Unit Price, Cage Code

Customer Number, Date, Time → Employee, Quantity Ordered

If we put all the information in a single table, there must be redundancy because the Name, Type, Cagy Code and Unit Price will be stored by many times. Thus, we need two tables in order for 1NF

in 1NF:

The primary key for the first table is **Customer Number, Date, Time**

The primary key for the second table is **Part Number**

<u>customerNumber (PK)</u>	
<u>date (PK)</u>	
<u>time (PK)</u>	
customerName	
customerType	
employee	
quantityOrdered	

<u>partNumber (PK)</u>	
name	
type	
cageCode	
unitPrice	

To get 2NF:

(1) there must be only one primary key in each table or

(2) there are composite keys but all the non-key attributes depend on the entire primary key, not part of it.

In our 1 NF, there are composite keys in first table, so we need to make all the non-key attributes depend on the entire primary key. However, Customer Name and Customer Type on depend on Customer Number, so we can divide the first table into two tables

In 2NF:

The primary key for the first table is **Customer Number**

The primary key for the second table is **Customer Number, Date, Time**

The primary key for the third table is **Part Number**

<u>customerNumber (PK)</u>	<u>customerNumber (PK, FK)</u>	<u>partNumber (PK)</u>
customerName	<u>date (PK)</u>	name
customerType	<u>time (PK)</u>	type
	employee	cageCode
	quantityOrdered	unitPrice

To get 3NF, we must remove all transitory dependencies in the tables. In our 2NF, there isn't any transitory dependency, so it is already 3NF

(c). in 1NF:

The primary key for the first table is **Customer Number, Date, Time**

The primary key for the second table is **Part Number**

In 2NF:

The primary key for the first table is **Customer Number**

The primary key for the second table is **Customer Number, Date, Time**

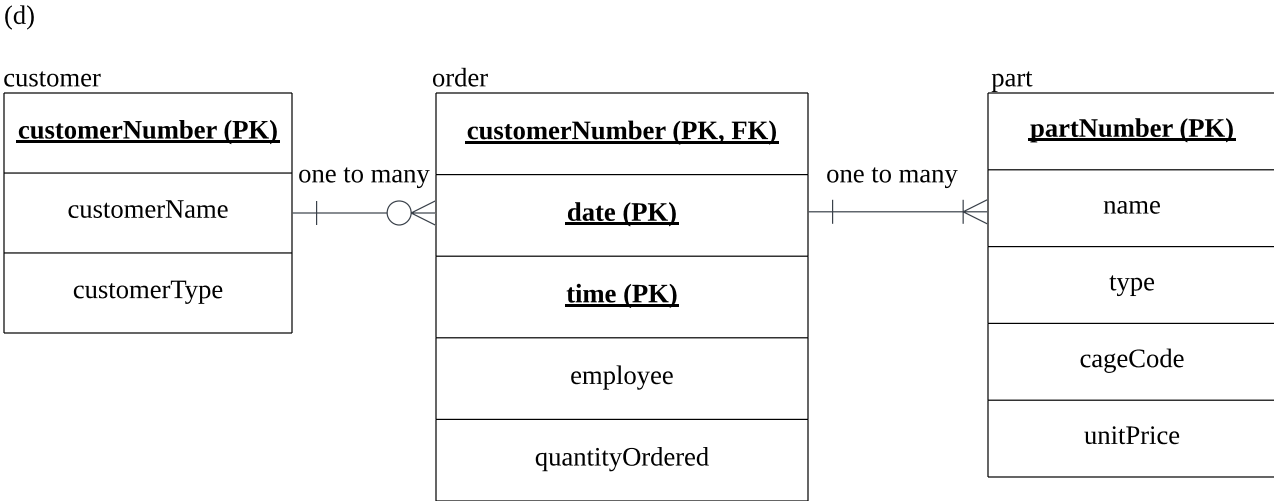
The primary key for the third table is **Part Number**

In 3NF:

The primary key for the first table is **Customer Number**

The primary key for the second table is **Customer Number, Date, Time**, The foreign key for the second table is **Customer Number**

The primary key for the third table is **Part Number**



2.

staffNo	therapistName	patNo	patName	appointment date	time	branchNo
S1011	Fred Smith	P100	Lily White	9/12/2022	10:00	M15
S1011	Fred Smith	P105	Jill Baker	9/12/2022	12:00	M15
S1024	Heidi Pierce	P108	Andy McKee	9/12/2022	10:00	Q10
S1024	Heidi Pierce	P108	Andy McKee	9/14/2022	14:00	Q10
S1032	Richard Levin	P105	Jill Baker	9/14/2022	16:30	M15
S1032	Richard Levin	P110	Jimmy Winter	9/15/2022	18:00	B13

(a). Assumption 1: staffNo is unique for each therapist, patNo is unique for each patient, and branchNo is unique for each branch

Assumption 2: each patient and make only one appointment at a given time on a given date, and each therapist can only be appointed once at a given time on a given date. No double appointment at the same time is allowed.

Assumption 3: appointmentDate and appointmentTime are inseperable, and they compose appointment. Thus, even only appointmentDate and staffNo determine branchNo, we should add appointTime, make appointment as whole be a determinant of branchNo.

(b). from the table above, we can draw the following relationships:

staffNo → therapistName

patNo → patName

staffNo, appointment → branchNo

To get 1NF, we need to ensure there are no repeating groups. Our original table has satisfy this requirement, so it is already in 1NF.

in 1NF:

the primary key is **staffNo, patNo, appointment**

<u>staffNo(PK)</u>
<u>patNo (PK)</u>
<u>appointment (PK)</u>
therapistName
patName
branchNo

To get 2NF:

(1) there must be only one primary key in each table or

(2) there are composite keys but all the non-key attributes depend on the entire primary key, not part of it.

Since we have composite keys in our 1NF, we need to make sure no non-key attribute depends on part of the primary key. In 1NF table, therapistName only depends on staffNo, and patName only depends on patNo, so we need to make extra tables for these two attributes. We should divide the 1NF table into 3 tables

in 2NF:

The primary key for the first table is **staffNo**

The primary key for the second table is **patNo**

The primary key for the third table is **staffNo, appointment**

<u>staffNo (PK)</u>	
therapistName	

<u>patNo (PK)</u>	
patName	

<u>staffNo (PK, FK)</u>	
<u>appointment (PK)</u>	
branchNo	

To get 3NF, we must remove all transitory dependencies in the tables. In our 2NF, there isn't any transitory dependency, so it is already 3NF

(c). in 1NF:

the primary key is **staffNo, patNo, appointment**

in 2NF:

The primary key for the first table is **staffNo**

The primary key for the second table is **patNo**

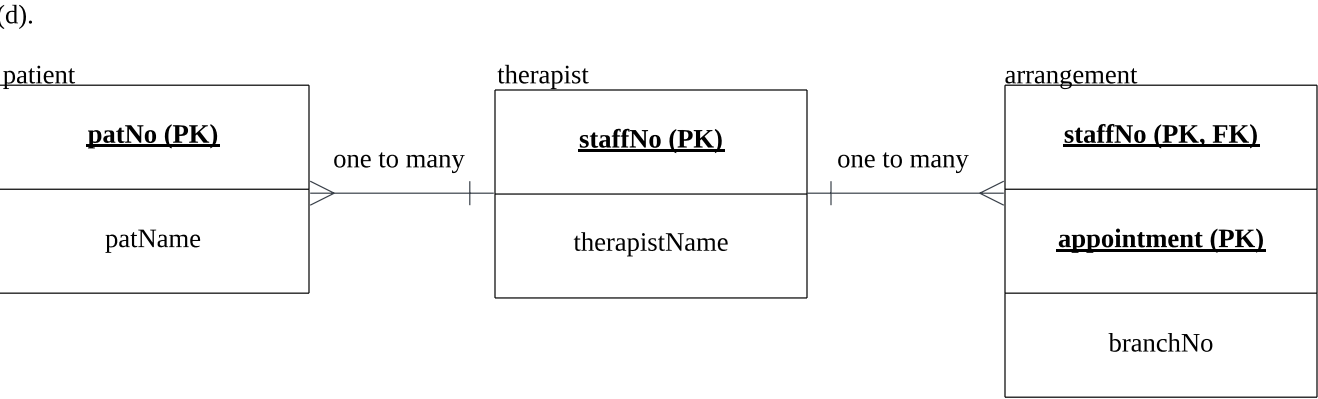
The primary key for the third table is **staffNo, appointment**

in 3NF:

The primary key for the first table is **staffNo**

The primary key for the second table is **patNo**

The primary key for the third table is **staffNo, appointment**. The foregine key for the third table is **staffNo**



3.

eNo	contractNo	hours	eName	eventNo	eventLoc
1135	C1024	16	Smith J	H25	Queens
1057	C1024	24	Hocine D	H25	Queens
1068	C1025	28	White T	H4	Yonkers
1135	C1025	15	Smith J	H4	Yonkers
1135	C1026	10	Smith J	H25	Queens

(a). Assumption 1: eventNo is unique for each event

Assumption 2: an employee can be included in many contracts, and a contract can include many employees. There is a many-to-many relationship between employee and contract.

Assumption 3: eventNo depends on contractNo; eventLoc depends on eventNo, and it implicitly depend on contractNo.

Assumption 4: hours worked only depends on eNo, not dependent on other attributes.

(b). from the table above, we can draw the following relationships:

eNo → eName, hours

contractNo → eventNo, eventLoc

To get 1NF, we need to ensure there are no repeating groups. Our original table has satisfy this requirement, so it is already in 1NF.

in 1NF:

the primary key is **eNo, contractNo**

<u>eNo(PK)</u>
<u>contractNo (PK)</u>
eName
hours
eventNo
eventLoc

To get 2NF:

(1) there must be only one primary key in each table or

(2) there are composite keys but all the non-key attributes depend on the entire primary key, not part of it.

Since we have composite keys in our 1NF, we need to make sure no non-key attribute depends on part of the primary key. In 1NF table, eName and hours only depends on eNo, and eventNo and eventLoc only depend on contractNo, so we need to divide the 1NF table into 2 tables

in 2NF:

The primary key for the first table is **eNo**

The primary key for the second table is **contractNo**

<u>eNo (PK)</u>	
eName	
hours	

<u>contractNo (PK)</u>	
eventNo	
eventLoc	

To get 3NF, we must remove all transitory dependencies in the tables. In our 2NF, there is a transitory dependency among contractNo, eventNo, and eventLoc. eventNo directly depends on contractNo, and eventLoc directly depends on eventNo, and indirectly depends on contractNo, so we should add a table.

in 3NF:

The primary key for the first table is **eNo**

The primary key for the second table is **contractNoNo**

The primary key for the third table is **eventNo**

<u>eNo (PK)</u>	
eName	
hours	

<u>contractNo (PK)</u>	
eventNo (FK)	

<u>eventNo (PK)</u>	
eventLoc	

(c). in 1NF:

the primary key is **eNo, contractNo**

in 2NF:

The primary key for the first table is **eNo**

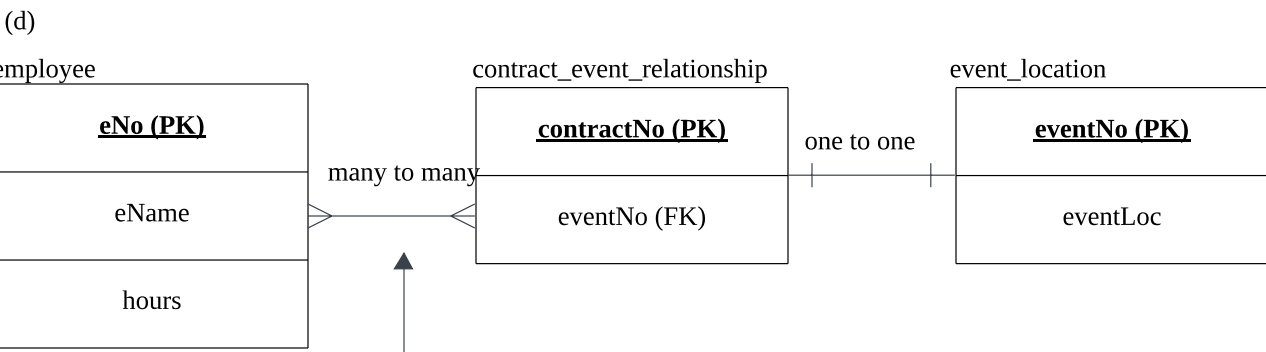
The primary key for the second table is **contractNo**

in 3NF:

The primary key for the first table is **eNo**

The primary key for the second table is **contractNoNoNo**. The foreign key for the second table is eventNo

The primary key for the third table is **eventNo**



Here, we can make a briging table such as "membership" to transform many-to-many relationship into two one-to-many relationships