Question 1

a. I'm making the assumption that there are three entities in this scenario: customer, employee, and parts. Customer and employee has a many to many relationship which has to be resolved through a bridging table later on, and customer and parts has a one to many relationship.

b. c. and d. :
I first created this table which is in 1NF:

transaction

customerName	customerNumber	customerType	date	time	employee	partNumber	name	type	cageCode	quantityOrdered	unitPrice
jeff peterson	HG54587	Consumer	7/1/2024	10:30am	D.Harrison	10654	Float Control	Plumbing	G413	4	12
jeff peterson	HG54587	Consumer	7/1/2024	10:30am	D.Harrison	10456	Modulator	Electrical	H433	3	7
jeff peterson	HG54587	Consumer	7/1/2024	10:30am	D.Harrison	10776	Hose Assembly	Plumbing	G413	7	9
jeff peterson	HG54587	Consumer	7/1/2024	10:30am	D.Harrison	10657	Float Assembly	Plumbing	G413	5	10

Since neither customerNumber or employee are unique identifiers on their own, the two come together to make the composite primary key **(customerNumber, employee)**. By doing this, I separate the original raw table into two: transation, and orderofTransaction. The first table captures the customer-employee interaction and the second the order of the customer. This is now in 2NF. In the latter table, customerNumber and partNumber come together to make its composite primary key.

transaction

customerNumber	**employee**	customerName	customerType	date	time
HG54587	D.Harrison	Jeff Peterson	Consumer	7/1/2024	10:30am

orderofTransaction

customerNumber	**partNumber**	name	type	cageCode	quantityOrdered	unitPrice
HG54587	10654	Float Control	Plumbing	G413	4	12
HG54587	10456	Modulator	Electrical	H433	3	7
HG54587	10776	Hose Assembly	Plumbing	G413	7	9
HG54587	10657	Float Assembly	Plumbing	G413	5	10

But the orderofTransaction table still has partial dependencies: the name of the part is functionally dependent on the partNumber which is then dependent on the customerNumber, but there's no direct dependency of name and customerNumber, same with the type and cageCode , and unitPrice. The quantityOrdered is dependent on both the partNumber and the customerNumber, so it stays in this table.

The next step is to split the orderofTransaction table into an orderofTransaction and a part table. In the part table, partNumber is both a PK and a FK.

transaction

customerNumber	**employee**	customerName	customerType	date	time
HG54587	D.Harrison	Jeff Peterson	Consumer	7/1/2024	10:30am

orders

orderofTransaction

customerNumber	***partNumber***	quantityOrdered
HG54587	10654	4
HG54587	10456	3
HG54587	10776	7
HG54587	10657	5

part

partNumber	name	type	cageCode	unitPrice
10654	Float Control	Plumbing	G413	12
10456	Modulator	Electrical	H433	7
10776	Hose Assembly	Plumbing	G413	9
10657	Float Assembly	Plumbing	G413	10

parts make up the order

Question 2

a. I'm assuming that staffNo and patNo are unique to therapists and patients. These two entities have a many to many relationship as shown in the raw table: the patient ses many therapists a day and a therapist sees many patients a day.

b.
The raw table is already in 1NF, but not all of the attributes are dependent on the PK, which is the staffNo. We recognize here also that (staffNo, patNo) are not even unique identifiers since the same patient+therapist combo can have many appointments on different days. So we can use **(staffNo, appointmentDay, appointmentTime)** as the composite PK since a therapist can only have one unique appointment on a specific date and time and at one unique branch.

c and d.:

appointment

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

We can rearrange the table this way so that the composite PK (**staffNo, appointmentDate, appointmentTime)** can act as a unique identifier. Here we can take the next step to remove partial dependencies: therapistName is only dependent on part of PK (staffNo).

appointment

staffNo	**appointmentDate**	**appointmentTime**	patNo	patName	branchNo
S1011	9/12/2022	10:00	P100	Lily White	M15
S1011	9/12/2022	12:00	P105	Jill Baker	M15
S1024	9/12/2022	10:00	P108	Andy McKee	Q10
S1024	9/14/2022	14:00	P108	Andy McKee	Q10
S1032	9/14/2022	16:30	P105	Jill Baker	M15
S1032	9/15/2022	18:00	P110	Jimmy Winter	B13

staff

staffNo	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levin

We can see here that we can take another step to remove the transitive dependency in the appointment table: patName is functionally dependent on patNo which is dependent on the PK. So we can thus spit the appointment table into two: appointment and patient tables.

appointment

staffNo	**appointmentDate**	**appointmentTime**	*patNo*	branchNo
S1011	9/12/2022	10:00	P100	M15
S1011	9/12/2022	12:00	P105	M15
S1024	9/12/2022	10:00	P108	Q10
S1024	9/14/2022	14:00	P108	Q10
S1032	9/14/2022	16:30	P105	M15
S1032	9/15/2022	18:00	P110	B13

staff takes appointments

staff

staffNo	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levin

makes appointment

patient

patNo	patName
P100	Lily White
P105	Jill Baker
P108	Andy McKee
P110	Jimmy Winter

Question 3

a. eNo is unique for each member of staff, and members and contracts have a many-to-many relationship: each employee can work multiple contracts and each contract can have multiple employees.

b. This raw table is in 1NF but not all attributes are dependent on the PK eNo. eName is only dependent on eNo, part of the PK. we can make the composite PK **(eNo, contractNo)** since eNo alone isn't a unique identifier for one employee can have multiple contracts. Here we can split the original table into two, one a contract table and the other an employee table.

c. and d.
contract

eNo	**contractNo**	hours	eventNo	eventLoc
1135	C1024	16	H25	Queens
1057	C1024	24	H25	Queens
1068	C1025	28	H4	Yonkers
1135	C1025	15	H4	Yonkers
1135	C1026	10	H25	Queens

employee

eNo	eName
1135	Smith J
1057	Hocine D
1068	White T

For our final step, we must resolve the partial dependency still left in the contract table: eventLoc is dependent on eventNo which is dependent on the PK. And thus we split the contract table into the contract and event tables as shown below.

employee works a contract

contract

eNo	**contractNo**	hours	*eventNo*
1135	C1024	16	H25
1057	C1024	24	H25
1068	C1025	28	H4
1135	C1025	15	H4
1135	C1026	10	H25

employee

eNo	eName
1135	Smith J
1057	Hocine D
1068	White T

a contract works towards an event

event

eventNo	eventLoc
H25	Queens
H4	Yonkers