Question 1

Assumptions:

1. A single order is uniquely identified by the triple *(customerNumber, orderDate, orderTime)*.
2. One employee processes the whole order (not per line).
3. Part price and storage cageCode are properties of the part at the time of this form.
4. A part is stored in exactly one cage (per the note that a cage code is the identifier where the inventory is stored).
5. Employees can help any customers; there's no special relationship between them.

ONF to 1NF

1. Assuming that the parts are repeating groups of attributes. We splite them out.

1NF

order_info	
PK	customerNumber
PK	orderDate
PK	orderTime
	customerName
	customerType
	employee

order_items	
PK	customerNumber
PK	orderDate
PK	orderTime
PK	partNumber
	partName
	partType
	cageCode
	unitPrice
	quantityOrdered

1NF to 2NF

1. Attributes: cusomterNumber, orderDate, orderTime, customerName, customerType, employee, partName, partType, cageCode, unitPrice, quantityOrdered
2. Define Important Terms
   a. Primary Key: Only a combination of customerNumber, orderDate, orderTime and partNumber can a specific order and order items be identified.
   b. Candidtate Key: The candidate key is also the primary key.
   c. Functional Dependency: customerNumber determines customerName and type. partNumber determines name, type, cage code and price.
   d. Partial Dependency: customerName and type is dependent on customer number, name, type, cage, price is depeendent on part number
   e. Transtive Dependency: Non are evident

Removing Partial Dependencies:
Since we identify the partial dependcies, we move customerName and CustomerType into a new table called customer-info with customerNumber as the PK.
We also create a new table for parts_info containning all attributes of each individual parts.

2NF

customer_info	
PK	customerNumber
	customerName
	customerType

order_items	
PK	customerNumber
PK	orderDate
PK	orderTime
PK	partNumber
	quantityOrdered

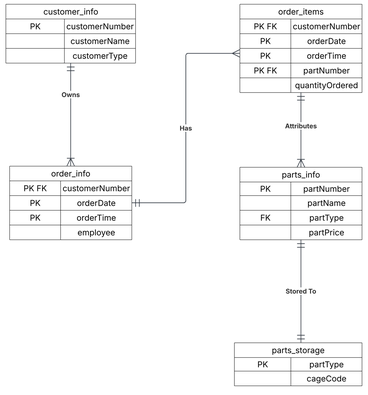
order_info	
PK	customerNumber
PK	orderDate
PK	orderTime
	employee

parts_info	
PK	partNumber
	partName
	partType
	cageCode
	unitPrice

2NF to 3NF

Removing Transitative Dependencies:
No clear transitative dependencies can be found; that is we can find a attribute dependent on a attribute.
But assuming that in a business, certain types of parts are placed normally in one area, we can infer that cageCode is dependent partType.

3NF



Question 2

Assumptions:

1. Therapists works only at one branch per day.
2. A booked time slot belongs to exactly one patients for a given therapist and day.

The table is already in 1NF form as no duplicate of attributes exists. All attributes are atomic.

1NF to 2NF

1. Attributes: staffNo, therapistName, patNo, patName, appointmentDateTime, and branchNo
2. Define Important Terms
   a. Primary Key: Only a combination of staffNo and appointmentTime can determine all other attributes.
   b. Candidtate Key: The candidate key is also the primary key.
   c. Functional Dependency: patNo is dependent on staffNo and appointment date and time. branchNo is dependent on staffNo and appoinment Date. staffName is dependent staffNo
   d. Partial Dependency: staffName is depdent on staffNo
   e. Transtive Dependency: patName is dependent on patNo

Removing Partial Dependencies:
Since we identify the partial dependcies, staffName is dependent on staffNo, and staffNo is a PK, we create a new table with just staffNo and staffName

2NF

appointments	
PK	staffNo
PK	appointment Date
PK	appointment Time
	patNo
	patName

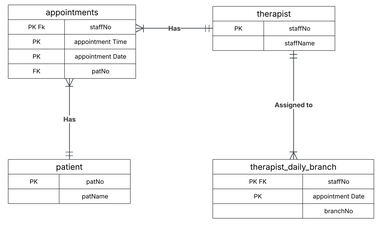
therapy_daily_branch	
PK	staffNo
PK	appointment Date
	branchNo

therapist	
PK	staffNo
	staffName

2NF to 3NF

Removing Transitative Dependencies:
patNo and patName are both attributes and not keys; they are transitative.
We create a new table to satisfy this anmoly.

3NF



Question 3

Assumptions:

1. Employees can work on different contracts.
2. Contracts can have different event locations.
3. Only one eventNo is assigned to eventLoc.

The table is already in 1NF form as no duplicate of attributes exists. All attributes are atomic.

1NF to 2NF

1. Attributes: eNo, contractNo, hours, eName, eventNo, and eventLoc
2. Define Important Terms
   a. Primary Key: Only a combination of eNo and contractNo can be a unique key.
   b. Candidtate Key: The candidate key is also the primary key.
   c. Functional Dependency: eNo detemrines eName. eventNo determines eventLoc. contractNo depents on eventNo. eNo and contractNo determines hours.
   d. Partial Dependency: eventNo and eventLoc are partially dependent on contractNo. eName is partially dependent on eNo.
   e. Transtive Dependency: eventLoc is dependent on eventNo

Removing Partial Dependencies:
From the identified partial dependency, we create a contract_location table where contractNo determines eventNo and eventLoc.
We also create a employee_info table, where eNo detmerines eName.

2NF

employee_working_hours	
PK	eNo
PK	contractNo
	hours

contract_location	
PK	contractNo
	eventNo
	eventLoc

employee_info	
PK	eNo
	eName

2NF to 3NF

Removing Transitative Dependencies:
eventNo and eventLoc are attributes and depdent on each other, thus we create a new table.

3NF