1. Local Supply Company

(a). Assumptions:
• A customer can have many orders.
• The customer number uniquely identifies a customer.
• Each customer has a customer type.
• Employees help any and all different customers, but one employee helps one customer at a time.
• The part number uniquely identifies each part.
• Different parts can have the same cage code.

(b). The raw table (all the data in one table):

partNumber	partName	partType	cageCode	quantityOrdered	unitPrice	customerNumber	customerName	customerType	orderDate	orderTime	employee
10654	Float Control	Plumbing	G413	4	12	HG54587	Jeff Peterson	Consumer	07-01-2024	10:30 a.m	D. Harrison
10456	Modulator	Electrical	H433	3	7	HG54587	Jeff Peterson	Consumer	07-01-2024	10:30 a.m	D. Harrison
10776	Hose Assembly	Plumbing	G413	7	9	HG54587	Jeff Peterson	Consumer	07-01-2024	10:30 a.m	D. Harrison
10657	Float Assembly	Plumbing	G413	5	10	HG54587	Jeff Peterson	Consumer	07-01-2024	10:30 a.m	D. Harrison

attributes: partNumber, partName, partType, cageCode, quantityOrdered, unitPrice, customerNumber, customerName, customerType, orderDate, orderTime, employee

candidate key: {partNumber, customerNumber, orderDate, orderTime}
reason: A customer can order different parts at different time. So only partNumber + customerNumber + orderDate + orderTime can uniquely identify a record.

dependencies:
partNumber → partName, partType, unitPrice
partNumber, customerNumber, orderDate, orderTime → quantityOrdered
customerNumber → customerName, customerType
customerNumber, orderDate, orderTime → employee
partType → cageCode (transitive dendency: partNumber → cageCode

normal form:
1NF. All the values are atomic values and there's no repeating groups. But there exits partial dependencies so it's not in 2NF and thus is not in 3NF.

convert to 2NF:

Data:

part

partNumber	partName	partType	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

order

partNumber	customerNumber	orderDate	orderTime	quantityOrdered
10654	HG54587	07-01-2024	10:30 a.m	4
10456	HG54587	07-01-2024	10:30 a.m	3
10776	HG54587	07-01-2024	10:30 a.m	7
10657	HG54587	07-01-2024	10:30 a.m	5

customer

customerNumber	customerName	customerType
HG54587	Jeff Peterson	Consumer

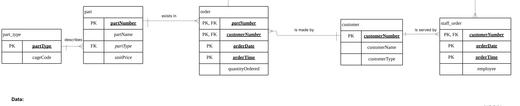
staff_order

customerNumber	orderDate	orderTime	employee
HG54587	Jeff Peterson	Consumer	D. Harrison

Explanation:
I separated the raw table (1NF) into four tables to remove partial and transitive dependencies. All part attributes that depend only on partNumber either were moved to the Part table. All customer attributes that depend only on customerNumber were moved to the Customer table. Order-level information such as employee that depends only on {customerNumber, orderDate, orderTime} was moved to the Staff_Order table. The remaining Order table keeps the composite key {partNumber, customerNumber, orderDate, orderTime} with quantityOrdered, so every non-key attribute fully depends on the whole key and no transitive dependencies remain.

And since there still exists the transitive dependency: partType → cageCode, it's not 3NF yet, and we need to seperate the cageCode and create a new table part_type to make it 3NF.

So the ultimate 3NF is:



Data:

part

partNumber	partName	partType	unitPrice
10654	Float Control	Plumbing	12
10456	Modulator	Electrical	7
10776	Hose Assembly	Plumbing	9
10657	Float Assembly	Plumbing	10

order

partNumber	customerNumber	orderDate	orderTime	quantityOrdered
10654	HG54587	07-01-2024	10:30 a.m	4
10456	HG54587	07-01-2024	10:30 a.m	3
10776	HG54587	07-01-2024	10:30 a.m	7
10657	HG54587	07-01-2024	10:30 a.m	5

customer

customerNumber	customerName	customerType
HG54587	Jeff Peterson	Consumer

part_type

partType	cageCode
Plumbing	G413
Electrical	H433

staff_order

customerNumber	orderDate	orderTime	employee
HG54587	Jeff Peterson	Consumer	D. Harrison

2. Therapists Appointment

(a). Assumptions:
• staffNo uniquely identifies a therapist.
• patNo uniquely identifies a patient.
• branchNo uniquely identifies a branch.
• A therapist may work at multiple branches across days, but on any given date a therapist works at one branch only.
• An appointment is for one patient with one therapist at one branch, at a specific appointmentDate and appointmentTime.
• A patient may have multiple appointments in a day, including with different therapists and at different times.

(b). The raw table (all the data in one table):

staffNo	therapistName	patNo	patName	appointmentDate	appointmentTime	branchNo
S1011	Fred Smith	P100	Lily White	09-12-2022	10:00	M15
S1011	Fred Smith	P105	Jill Baker	09-12-2022	12:00	M15
S1024	Heidi Pierce	P108	Andy McKee	09-12-2022	10:00	Q10
S1024	Heidi Pierce	P108	Andy McKee	09-14-2022	14:00	Q10
S1032	Richard Levis	P105	Jill Baker	09-14-2022	16:30	M15
S1032	Richard Levis	P110	Jimmy Winser	09-15-2022	18:00	B13

attributes: staffNo, therapistName, patNo, patName, appointmentDate, appointmentTime, branchNo

candidate key: {staffNo, patNo, appointmentDate, appointmentTime}
reason: A patient is given an appointment at a specific time and date at a particular branch with one therapist. So only staffNo + patNo + appointmentDate + appointmentTime can uniquely identify a record.

dependencies:
staffNo → therapistName
patNo → patName
staffNo, appointmentDate → branchNo

normal form:
1NF. All the values are atomic values and there's no repeating groups. But there exits partial dependencies so it's not in 2NF and thus is not in 3NF.

convert to 2NF:

Data:

therapist

staffNo	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levis

patient

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

appointment

staffNo	patNo	appointmentDate	appointmentTime
S1011	P100	09-12-2022	10:00
S1011	P105	09-12-2022	12:00
S1024	P108	09-12-2022	10:00
S1024	P108	09-14-2022	14:00
S1032	P105	09-14-2022	16:30
S1032	P110	09-15-2022	18:00

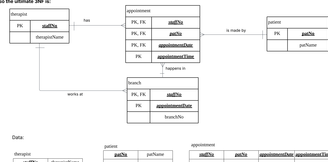
branch

staffNo	appointmentDate	branchNo
S1011	09-12-2022	M15
S1024	09-12-2022	Q10
S1024	09-14-2022	Q10
S1032	09-14-2022	M15
S1032	09-15-2022	B13

Explanation:
I separated the raw table (1NF) into four tables to remove partial and transitive dependencies. All part attributes that depend only on staffNo were moved to the Therapist table. All patient attributes that depend only on patNo were moved to the Patient table. Branch-level information, branchNo, that depends only on (staffNo, appointmentDate) was moved to the Branch table. The remaining Appointment table keeps the composite key (staffNo, patNo, appointmentDate, appointmentTime), so every non-key attribute fully depends on the whole primary key.

And since in all four tables every non-key attribute depends directly on the whole primary key and no non-key attribute depends on another non-key attribute, there are no transitive dependencies remaining and the design is already in 3NF.

So the ultimate 3NF is:



Data:

therapist

staffNo	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levis

patient

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

appointment

staffNo	patNo	appointmentDate	appointmentTime
S1011	P100	09-12-2022	10:00
S1011	P105	09-12-2022	12:00
S1024	P108	09-12-2022	10:00
S1024	P108	09-14-2022	14:00
S1032	P105	09-14-2022	16:30
S1032	P110	09-15-2022	18:00

branch

staffNo	appointmentDate	branchNo
S1011	09-12-2022	M15
S1024	09-12-2022	Q10
S1024	09-14-2022	Q10
S1032	09-14-2022	M15
S1032	09-15-2022	B13

3. Event Staff

Assumptions & The raw table (all the data in one table):

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

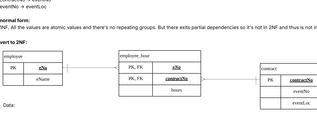
attributes: eNo, contractNo, hours, eName, eventNo, eventLoc

candidate key: (eNo, contractNo)
reason: Each row records the hours worked by a specific employee on a specific contract. Because each contract applies to one event, eNo + contractNo uniquely identifies a record.

dependencies:
eNo → eName
eNo, contractNo → hours
contractNo → eventNo
eventNo → eventLoc

normal form:
1NF. All the values are atomic values and there's no repeating groups. But there exits partial dependencies so it's not in 2NF and thus is not in 3NF.

convert to 2NF:



Data:

employee

eNo	eName
1135	Smith J
1057	Hocine D
1068	White T

employee_hour

eNo	contractNo	hours
1135	C1024	16
1057	C1024	24
1068	C1025	28
1135	C1025	15
1135	C1026	10

contract

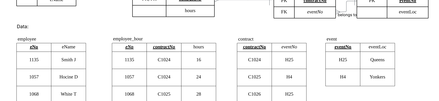
contractNo	eventNo	eventLoc
C1024	H25	Queens
C1025	H4	Yonkers
C1026	H25	Queens

Explanation:
I separated the raw table (1NF) into four tables to remove partial and transitive dependencies. All employee attributes that depend only on eNo were moved to the Employee table. All event attributes that depend only on eventNo were moved to the Event table. Contract-level information, such as the eventNo and eventLoc , that depends only on contractNo was moved to the Contract table. The remaining Employee_Hour table keeps the composite key (eNo, contractNo) with hours, so every non-key attribute fully depends on the whole primary key.

Since eventLoc still depends only on eventNo rather than on the whole primary key of the Contract table, there is a transitive dependency between contractNo and eventLoc through eventNo. So it's not 3NF .

To remove this redundancy and ensure that every non-key attribute depends only on the key of its own table, we further separate eventNo and eventLoc into the Event table and link it to the Contract table. This eliminates the transitive dependency and brings the design into full Third Normal Form.

convert to 3NF:



Data:

employee

eNo	eName
1135	Smith J
1057	Hocine D
1068	White T

employee_hour

eNo	contractNo	hours
1135	C1024	16
1057	C1024	24
1068	C1025	28
1135	C1025	15
1135	C1026	10

contract

contractNo	eventNo
C1024	H25
C1025	H4
C1026	H25

event

eventNo	eventLoc
H25	Queens
H4	Yonkers