

employeeId: Assumed that all employee has an unique id that served for the PK.
customerNumber: Assumed that this number is unique and is assigned to on customer only
partNumber: Assumed that part number is unique to a part and can only refering to one specific part
cageCode: Assumed that this is the code where the part is stored. It is the code for location. Therefore assumed that this is dependent on part number.
partName, partType, cageCode, quantityOrdered, unitPrice: All Assumed to be dependent on partNumber.
order: Assumed that one customer can have many orders but each order can only have one customer. Assumed that one employee can be dealing with many orders but each order can only have one employee doing it. Assumed that one order can has many parts but each part unique part can only be in one order.

ONF Normalization:
All fields are atomic but repetitive data occur in:customerNumber customerName customerType employeeName date time
Currently the primary is the customerNumber. But partial and transitive dependency also exist therefore the table needs to be broken down

warehouse_order											
<u>customerNumber</u>	customerName	customerType	employeeName	date	time	partNumber	partName	partType	cageCode	quantityOrdered	unitPrice
HG54587	Jeff Peterson	consumer	D.Harrison	7/1/2024	10:00am	10654	Float Control	Plumbing	G413	4	12
HG54587	Jeff Peterson	consumer	D.Harrison	7/1/2024	10:00am	10456	Modulator	Electrical	H433	3	7
HG54587	Jeff Peterson	consumer	D.Harrison	7/1/2024	10:00am	10776	Hose Assembly	Plumbing	G413	7	9
HG54587	Jeff Peterson	consumer	D.Harrison	7/1/2024	10:00am	10657	Float Assembly	Plumbing	G413	5	10

Further Normalization:
I broke down the table to: a customer places an order with an employee. This customer table is 3NF. The primary key is customerNumber

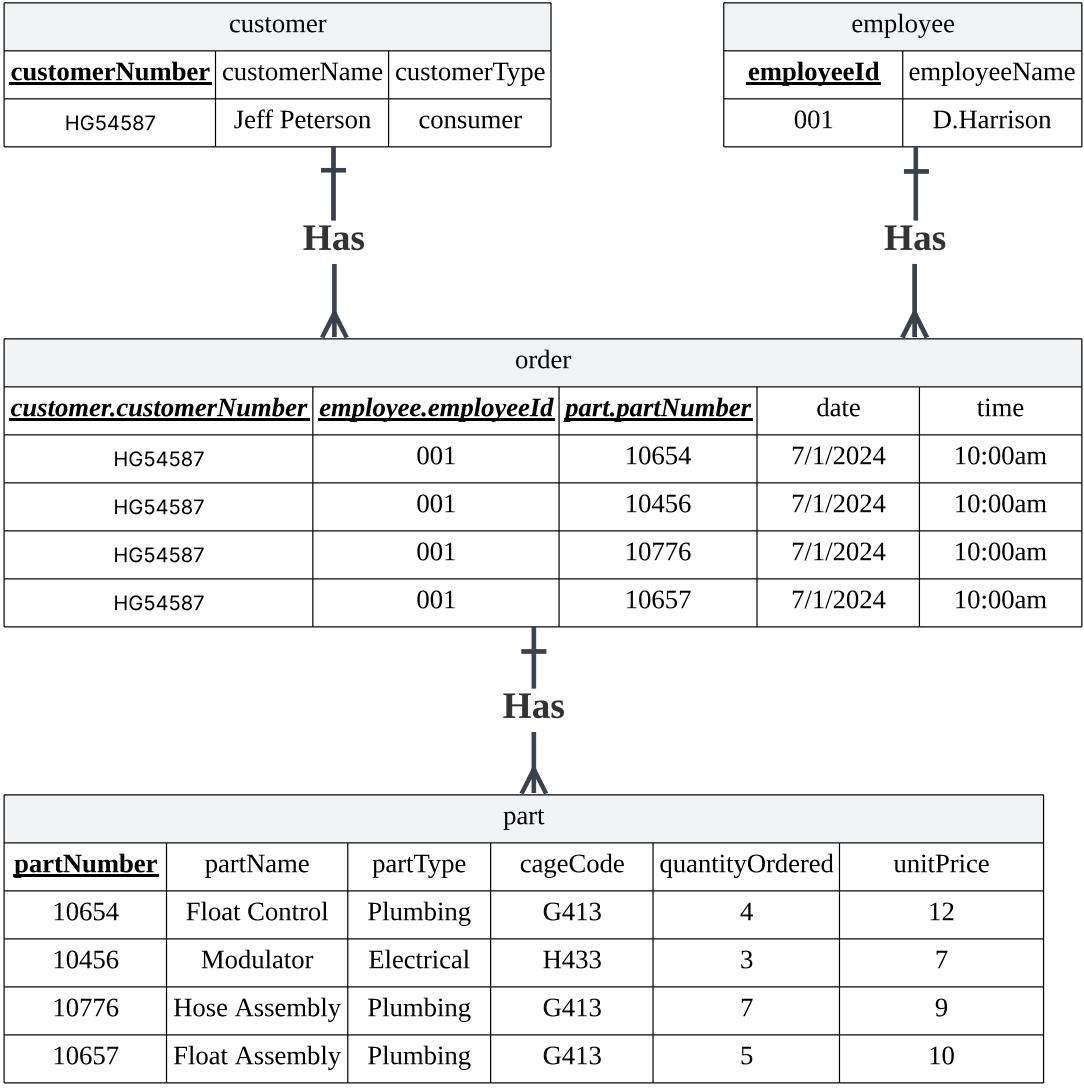
customer		
<u>customerNumber</u>	customerName	customerType
HG54587	Jeff Peterson	consumer

Further Normalization:
I broke down the table to: a customer places an order with an employee. This employee table is 3NF. The table needs a primary key so I added employeeId as a primary key

employee	
<u>employeeId</u>	employeeName
001	D.Harrison

Further Normalization:
I broke down the table to: a customer places an order with an employee. This "part" table is 1NF. There is a composite key made of customerNumber and employeeId. Date and time depends on this composite key. However, I found that "partName partType cageCode quantityOrdered unitPrice" all depend on partNumber only. There is a partial dependency therefore 2NF is broken.

part									
<u>customer.customerNumber</u>	<u>employee.employeeId</u>	<u>partNumber</u>	date	time	partName	partType	cageCode	quantityOrdered	unitPrice
HG54587	001	10654	7/1/2024	10:00am	Float Control	Plumbing	G413	4	12
HG54587	001	10456	7/1/2024	10:00am	Modulator	Electrical	H433	3	7
HG54587	001	10776	7/1/2024	10:00am	Hose Assembly	Plumbing	G413	7	9
HG54587	001	10657	7/1/2024	10:00am	Float Assembly	Plumbing	G413	5	10



Final Normalization:
The "customer" table is in 3NF and has PK: customerNumber

Final Normalization:
The "employee" table is in 3NF and has PK: employeeId

Final Normalization:
The "order" table was made as a bridging table that connects customer and employee and part. Its PKs came from the other three tables. It has composite PKs which are also FK from other tables. The table is in 3NF

Final Normalization:
The "part" table is in 3NF and has PK: partNumber

contractNo: Assumed that multiple eNo will have one contractNo.
hours: Assumed that the working hour is decided by both the employee and the contract
contractNo: Assuemd one contract can have many events but one event can only have one contract.
eventLoc: Assumed directly depends on eventNo. One eventNo can only have on eventLoc

1NF:
All fields are atomic and no repetitive data exists.
However there is a patial dependency where eName depends on eNo while eventLoc depends only on eventNo depends only on contractNo. The current PK is eNo

event_management					
<u>eNo</u>	contractNo	hours	eName	eventNo	eventrLoc
1135	C1024	16	Smith J	H25	Queen
1057	C1024	24	Hocine D	H25	Queen
1068	C1025	28	White T	H4	Yonkers
1135	C1025	15	Smith J	H4	Yonkers
1135	C1026	10	Smith J	H25	Queen

Further Normalization:
This is a 3NF table where eName directly depends on the PK eNo

employee	
<u>eNo</u>	eName
1135	Smith J
1057	Hocine D
1068	White T

Further Normalization:
This is a 2NF table where eventNo directly depends on contractNo but evenLoc depends on eventNo. A transitive dependency exists

contract		
<u>contractNo</u>	eventNo	eventLoc
C1024	H25	Queen
C1025	H4	Yonkers
C1026	H25	Queen

Further Normalization:
This is a 3NF table where FK and PK eNo(employee) and contractNo(contract) forms a composite key that decides how many hours are worked

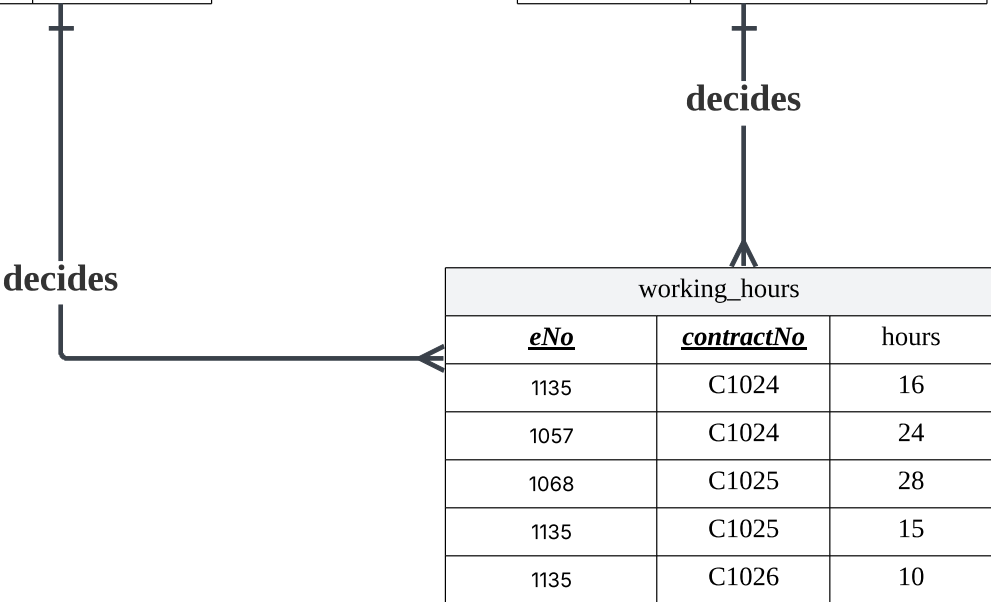
working_hours		
<u>eNo</u>	<u>contractNo</u>	hours
1135	C1024	16
1057	C1024	24
1068	C1025	28
1135	C1025	15
1135	C1026	10

Final Normalization:
This is a 3NF table now where eventLoc directly depends on contractNo. eventNo as PK

event	
<u>eventNo</u>	eventLoc
H25	Queen
H4	Yonkers

employee	
<u>eNo</u>	eName
1135	Smith J
1057	Hocine D
1068	White T

contract	
<u>contractNo</u>	eventNo
C1024	H25
C1025	H4
C1026	H25



Final Normalization:
Employee has PK eNo. eName directly depends on that.
Contract has PK contractNo. eventNo directly depends on that.
Working hours has PK FK eNo and contractNo. These two fields decided the working hour. Becuase the company tracks employee hours againsts contracts. So hours are decided by these two fields.
Event has PK eventNo. eventLoc directly depends on that.

staffNo: Assumed that this number is unique and is assigned to one staff only
patNo: Assumed that this number is unique and is assigned to one patient only
branchNo: Assumed that this is dependent on the one patient and one staff together because one therapist can be at different branches based on the need of the patients.
appointment: Assumed that one therapist can take many patients at different times and one patient can have have many therapists at different times. Assumed that one appointment can only have one unique time with one staff and one patient.

ONF:
staffNo and therapist names happens to be repeated data and appointmentDateAndTime is not atomic.
Currently the primary is the staffNo. But partial and transitive dependency also exist therefore the table needs to be broken down

mental_health_cooperation_track					
<u>staffNo</u>	therapistName	patNo	patName	appointmentDateAndTime	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	winter	9/15/2022 18:00	B13

Further Normalization:

I broke down the table to: a staff treats a patient at a certain time in a branch. This staff table is 3NF. The PK is: staffNo.

staff	
<u>staffNo</u>	therapistName
s1011	Fred Smith
s1024	Heidi Pierce
s1032	Richard Levin

Further Normalization:

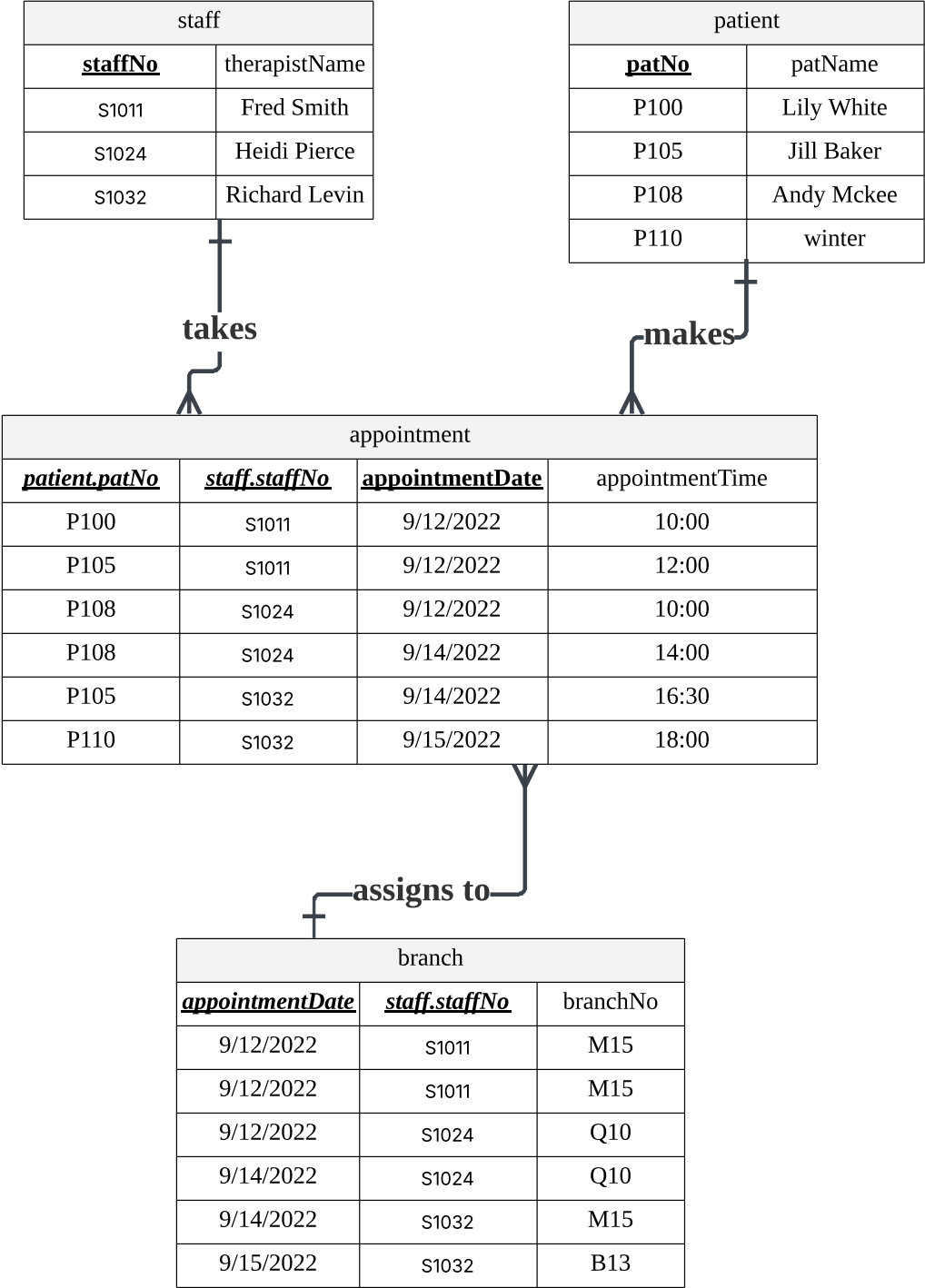
I broke down the table to: a staff treats a patient at a certain time in a branch. This appointment table is 2NF. The PK is: patNo and staff.staffNo. The staff.staffNo is also a FK. Two PKs formed a composite key. However, here I have a transitive dependency where appointmentDate is dependent on the composite key but appointmentTime depends on appointmentDate. A bridging table is needed.

appointment				
<u>patient.patNo</u>	<u>staff.staffNo</u>	appointmentDate	appointmentTime	branchNo
P100	s1011	9/12/2022	10:00	M15
P105	s1011	9/12/2022	12:00	M15
P108	s1024	9/12/2022	10:00	Q10
P108	s1024	9/14/2022	14:00	Q10
P105	s1032	9/14/2022	16:30	M15
P110	s1032	9/15/2022	18:00	B13

Further Normalization:

I broke down the table to: a staff treats a patient at a certain time in a branch. This patient table is 3NF. The PK is: patNo.

patient	
<u>patNo</u>	patName
P100	Lily White
P105	Jill Baker
P108	Andy Mckee
P110	winter



Final Normalization:

I broke down the table and now they are all 3NF. One staff and a patient can together set up an appointment. PK of staffNo in staff and PK of patNo in patient together forms a composite key in appointment and branch. The Appointment has the PK of the composite PK and the appointmentDate. Since one staff can only be working at one branch in one day. The composite PK of FK appointmentDate and staffNo will assign the branch