1. Happy Supplies Parts Warehouse

raw table

Customer Name:	Jeff Peterson	**Date:**	7/1/2024
Customer Number:	HG54587	**Time:**	10:30am
		Employee:	D. Harrison
Customer Type:	Consumer		

Part Number	Name	Type	Cage Code	Quantity Ordered	Unit Price
10654	Float Control	Plumbing	G413	4	12
10456	Modulator	Electrical	H433	3	7
10776	Hose Assembly	Plumbing	G413	7	9
10657	Float Assembly	Plumbing	G413	5	10

List of assumptions:
- A customer cannot place multiple orders on the same date for the same part (in 1NF): parts_order
- A customer cannot place multiple orders on the same date at the same time (in 2NF, 3NF): order_log
- A customer cannot place multiple orders on the same date at the same time for the same part (2NF, 3NF): order_info
- A employee can handle any and all customers; they are not linked to one customer
- Employee names are unique
- Unit price adjusts in real time per part ; theres no historical record of price
- unitPrice and quantityOrdered will always be non-negative
- customerName is atomic

parts_order

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

This table is in 1NF. First, we added a composite PK of customerNumber, orderDate, partNumber. This PK operates under the assumption that a customer will not a place separate order for the same part on the same date. If that was the case, we could add the orderTime field to ensure unique records. Second, we made sure there are no repeating groups, assuming customerName is atomic (and shouldn't be split into firstName, lastName) since we cannot introduce new fields.

customer

customerNumber	customerName	customerType
HG54587	Jeff Peterson	Consumer

order_log

orderDate	**orderTime**	employee	***customerNumber***
07-01-2024	10:30 AM	D. Harrison	HG54587

Customer places an order with an employee

Customer places an order for a part at a certain amount, they can place many orders

order_info

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

Part is in the order, many parts can be in one order

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

Now this database is in 2NF and 3NF. It is in 2NF because we created 1NF and there are no partial dependencies. We create customer table so constumerName and costumerType both depend on the whole PK (customerNumber). we create order_log so employee depends on the whole composite PK; here, we have added orderTime as a PK component since it is plausible a customer places more than one order on the same date (we originally assumed it to not repeat for the same part). In orde_info, quantityOrdered depends on the large PK and it is obvious in part that all fields depend on PartNumber. Thus, the database is in 2NF. Furthermore, the database is in 3NF since there are no transitive dependenices, meaning every non key attribute depends only on the key in every table.

Here are the FKs if the italics is not clear: customerNumber and partNumber in order_info, customerNumber in order_log. Idea: if we were to add a seconds part on the orderTime, that would eliminate a couple assumptions we need to make.

2. Mental Health Corp

raw table

staffNo	therapistName	patNo	patName	appointment date	appointment 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

List of assumptions:
- An appointment has only one staff
- An appointment has only one patient
- Therapist can work out of any branch
- Patient can book appointment at any branch
- Patients may have many appointments in one day with the same or different therapists
- Assumes the name are atomic seperated from firstName and lastName

appointment

staffNo	therapistName	**patNo**	patName	**date**	**time**	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 Levin	P105	Jill Baker	09-14-2022	16:30	M15
S1032	Richard Levin	P110	Jimmy Winter	09-15-2022	18:00	B13

The table is in 1NF. The composite PK uniquely already identifies each appointment, and we removed the repeating group under the original appointment column to create atomic valued columns of date and time. Also, we assume the names of patients and therapists are atomic and not seperated as firstName and lastName The records are unique since a therapist cannot see a patient more than once at the same time on the same day.

appointment

date	**time**	branchNo	***patNo***	***staffNo***
09-12-2022	10:00	M15	P100	S1011
09-12-2022	12:00	M15	P105	S1011
09-12-2022	10:00	Q10	P108	S1024
09-14-2022	14:00	Q10	P108	S1024
09-14-2022	16:30	M15	P105	S1032
09-15-2022	18:00	B13	P110	S1032

Therapists work many appointments; they can work many over time and in the same day

Patients make appointments; they can make many

therapist

staffNo	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levin

patient

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

This database is in 2NF and 3NF. It is in 2NF because it is in 1NF and there are no partial dependencies. In patient, patName depends entirely on patNo. In therapist, therapistName depends entirely on staffNo. In appointment, branchNo depends on the entire composite PK as it shows the location of the appointment. For 3NF, we assume no transitive dependencies so that a therapist can operate out of any branch and a patient can schedule an appointment at any branch. Else, a therapist would be tied to one branch and a transitive dependency would exist, the branch on the therapist; this would require further normalization and an added surrogate key for branchId, which we are not allowed to create. Also, since it explicitly says "therapiists only see patients at one specific branch on any given day," we knot that branch is also dependent on date for certain

If italics is not clear, patNo and staffNo are FK in appointment table.

3. Maid Better temp agency

raw 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

List of assumptions:
- Assumes eName is atomic seperated from firstName and lastInitial
- Each contract only applies to one event
- An event can have many contracts
- Each event has one location
- Hours worked are per employee per contract so (eNo, contractNo) is unique

contract

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

The table is in 1NF. There are no repeating groups assuming that eName is an atomic value. Also, the composite key of eNo and contractNo uniquely identifies every record assuming hours worked are per employee per contract.

employee

eNo	eName
1135	Smith J
1057	Hocine D
1068	White T

assignment

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

An employee is assigned to a contract and is given hours worked at that event; they can work on multiple events/contracts

An event or assignment has a contract; they can have many

contract

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

Now we are in 2NF. First, the tables are in 1NF. Also, eName depends on eNo, hours depends on both eNo and contractNo as they vary by contract per employee, and eventNo and eventLoc both depend on the contract; so there are no partial dependenices.

The FKs are eNo and contractNo in assignment table.

employee

eNo	eName
1135	Smith J
1057	Hocine D
1068	White T

event

eventNo	eventLoc
H25	Queens
H4	Yonkers

An employee gets an assingment; they can have multiple

An event has at least one contract

assignment

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

A contract has assignments for emloyeess

contract

contractNo	*eventNo*
C1024	H25
C1025	H4
C1026	H25

Now we are in 3NF. First, the talbes are in 1NF and 2NF. Also, we removed a transitive dependency where eventLoc depends on eventNo which are both non key attributes. We split this transitive dependency by movign EventLoc to its own event table and link it to the eventNo, using eventNo as an FK in contract table.

The FKs are eventNo in contract table, contractNo and eNo in assignment table.