

Problem 1: Happy Supplies Parts Warehouse

Assumptions:

- A part can be ordered by different customers.
- Different employees can help the same customer on the same day if it's at a different time.
- No relationship between specific employees and customers.
- There is no relationship between type and cageCode. Although there appears to be one, I'm going to assume that this is coincidental and this relationship is not always present.

1NF

order										
customerName	customerNumber	customerType	date	time	employee	partNumber	name	type	cageCode	unitPrice
Jeff Peterson	HG54587	consumer	7/1/2024	10:30am	D. Harrison	10654	Float Control	Plumbing	G413	12
Jeff Peterson	HG54587	consumer	7/1/2024	10:30am	D. Harrison	10456	Modulator	Electrical	H433	7
Jeff Peterson	HG54587	consumer	7/1/2024	10:30am	D. Harrison	10776	Hose Assembly	Plumbing	G413	9
Jeff Peterson	HG54587	consumer	7/1/2024	10:30am	D. Harrison	10657	Float Assembly	Plumbing	G413	10

Primary Key: customerNumber, date, time, partNumber

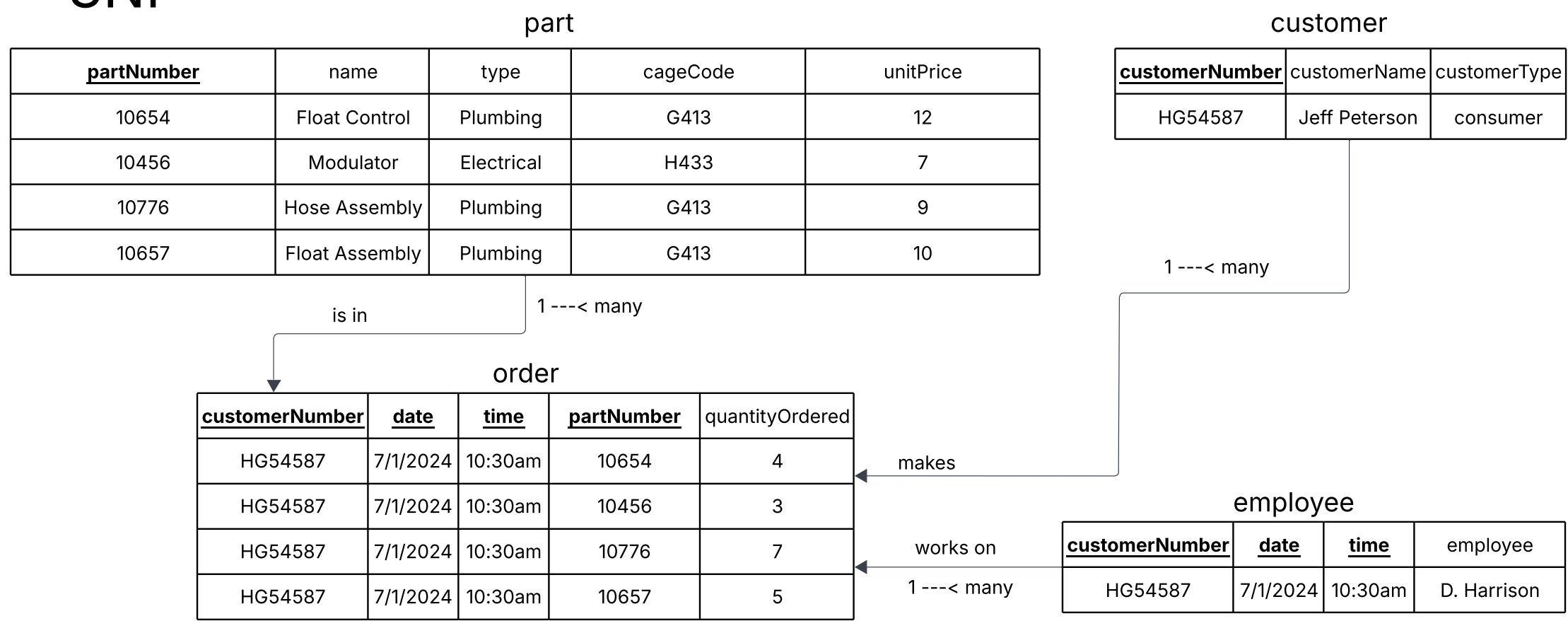
The table is in 1NF because it is comprised of atomic values, has no repeating groups/columns, and has a primary key (enforcing uniqueness).

The table is not in 2NF because there are partial dependencies (something is not dependent on the ENTIRE PK):

- customerName, customerType are dependent on customerNumber
- employee is dependent on customerNumber, date, time
- name, type, cageCode, unitPrice is dependent on partNumber
- quantityOrdered is dependent on customerNumber, date, time, partNumber

These partial dependencies need to be resolved for by creating new tables in order to bring this table into the "2NF" form.

3NF



After resolving the partial dependencies by creating new tables, we find that there are no transitive relationships, meaning that our table is now in 3NF form. Note that because there are no transitive relationships, we don't need any foreign keys.

Problem 2: Panacea Mental Health Corporation

Assumptions:

- A certain staff member and patient can meet more than one time per day (essentially making time part of the primary key).
- Therapists and patients have their own unique number.
- We follow all specification requirements stated in the prompt.
- Patients can only have one appointment at a time.
- A patient is given an appointment at a specific time and date at a particular branch with one therapist (staffNo can be determined by the combination patNo, date, time).

appointment_information					
Q10	therapistName	patNo	patName	appointment date	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

This table is not in 1NF because the appointment part of the table has multiple values in one column. As a result, in order to get this table into 1NF, we need to split this into two columns.

1NF

staffNo	therapistName	patNo	patName	date	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

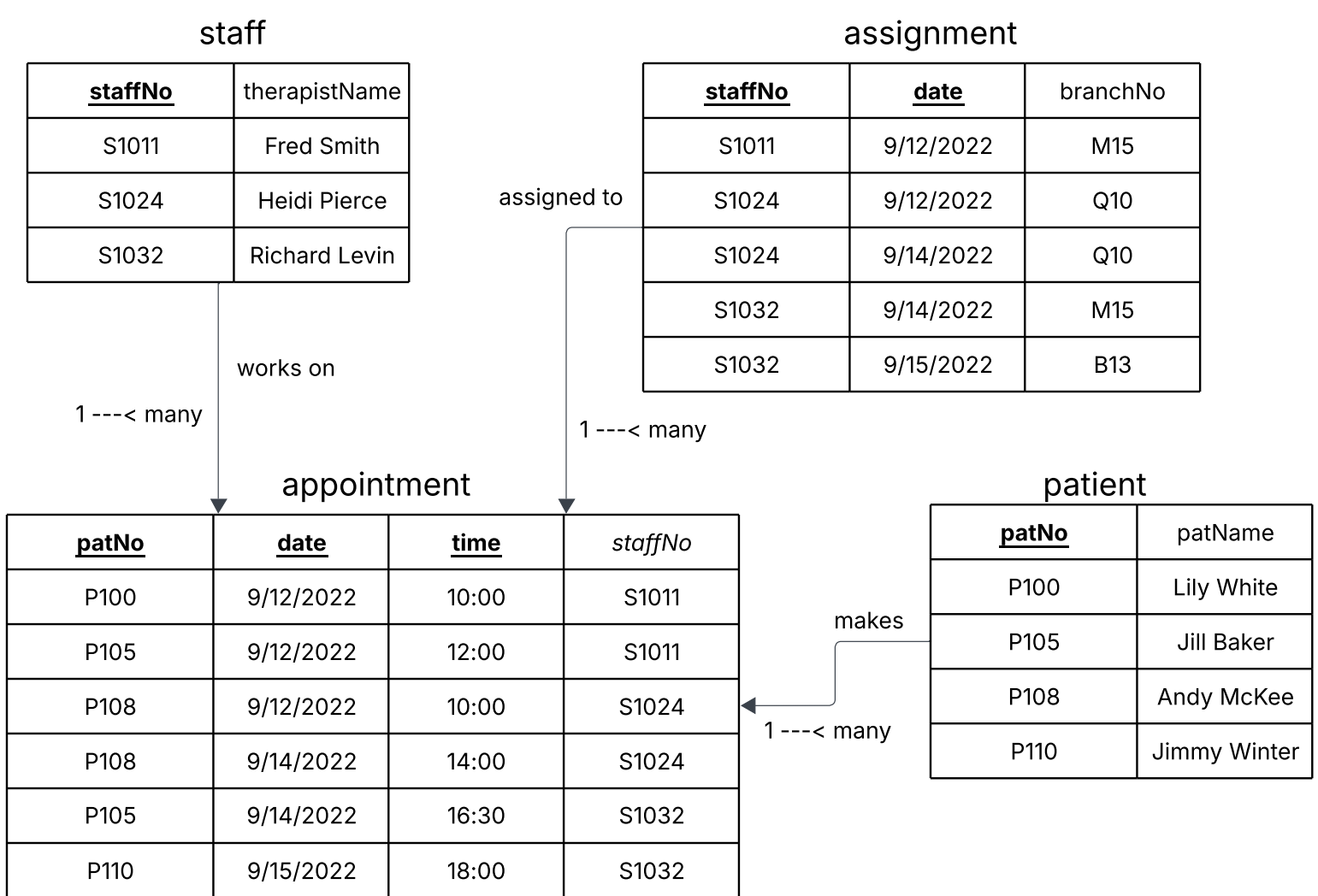
The table is in 1NF because it is comprised of atomic values, has no repeating groups/columns, and has a primary key (enforcing uniqueness).

The table is not in 2NF because there are partial dependencies (something is not dependent on the ENTIRE PK):

- therapistName is dependent on staffNo.
- patName is dependent on patNo
- branchNo is dependent on staffNo, date as stated in the prompt.
- staffNo, branchNo is dependent on patNo, date, time as stated in the prompt.

These partial dependencies need to be resolved for by creating new tables in order to bring this table into the "2NF" form.

3NF



After resolving the partial dependencies by creating new tables, we find that there are no transitive relationships that we need to resolve, meaning that our table is now in 3NF form. Note that we do have a foreign key in the main table. Since we found that staffNo was dependent on patNo, date, time, it does not need to be part of our composite primary key. However, it is a primary key in the "staff" table, meaning it becomes a foreign key in the original table.

Assumptions:

- Employee number is a unique identifier for employee.
- Contract number is a unique identifier contract.
- Each contract only applies to one event.
- Each event is only in one location.
- An event can have multiple different contracts.

1NF

company_data					
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

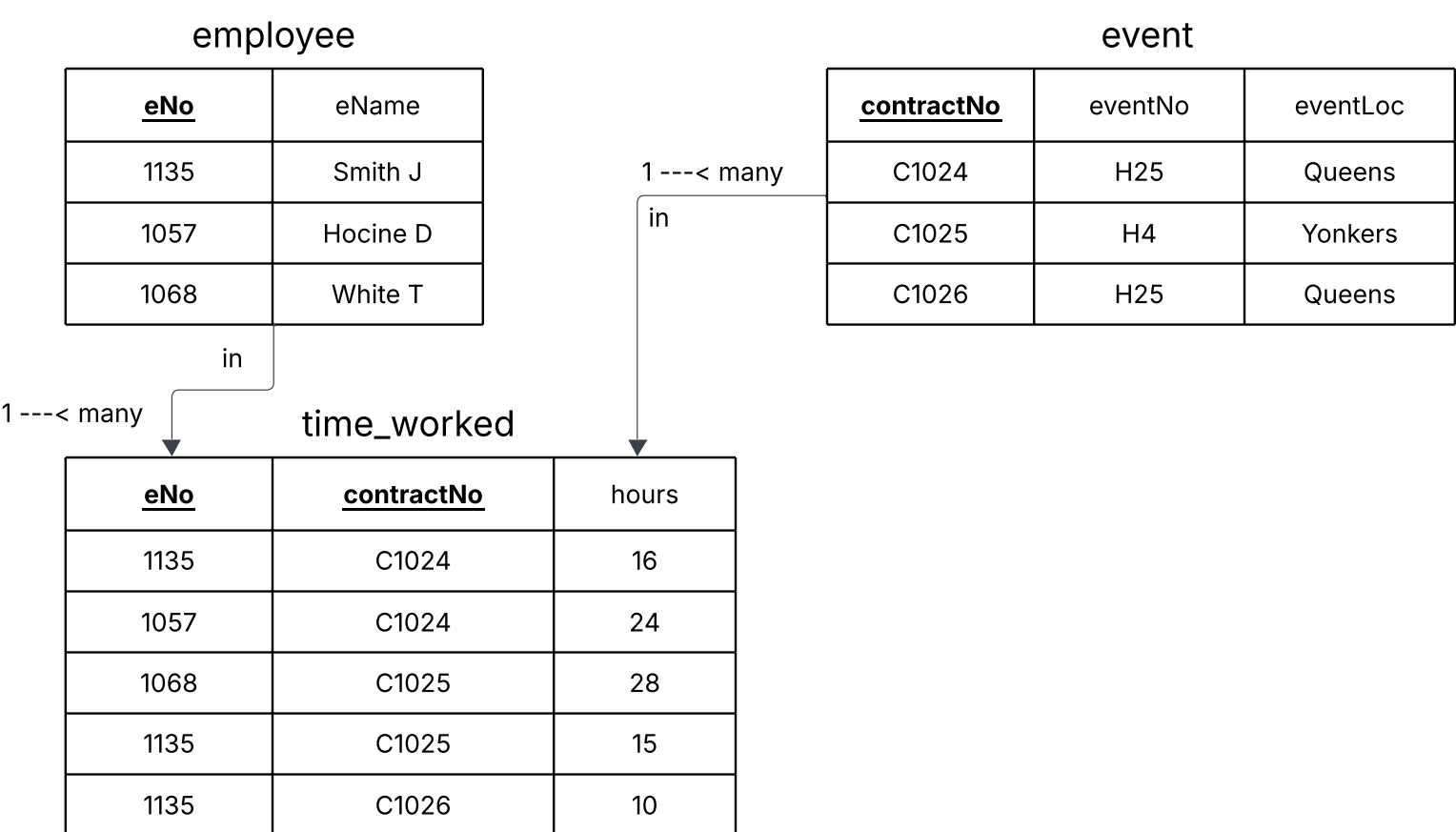
The table is in 1NF because it is comprised of atomic values, has no repeating groups/columns, and has a primary key (enforcing uniqueness).

The table is not in 2NF because there are partial dependencies (something is not dependent on the ENTIRE PK):

- eName is dependent on eNo
- eventNo is dependent on contractNo
- eventLoc is dependent on eventNo, which is dependent on contractNo, thus making it dependent on contractNo
- hours is dependent on eNo, contractNo

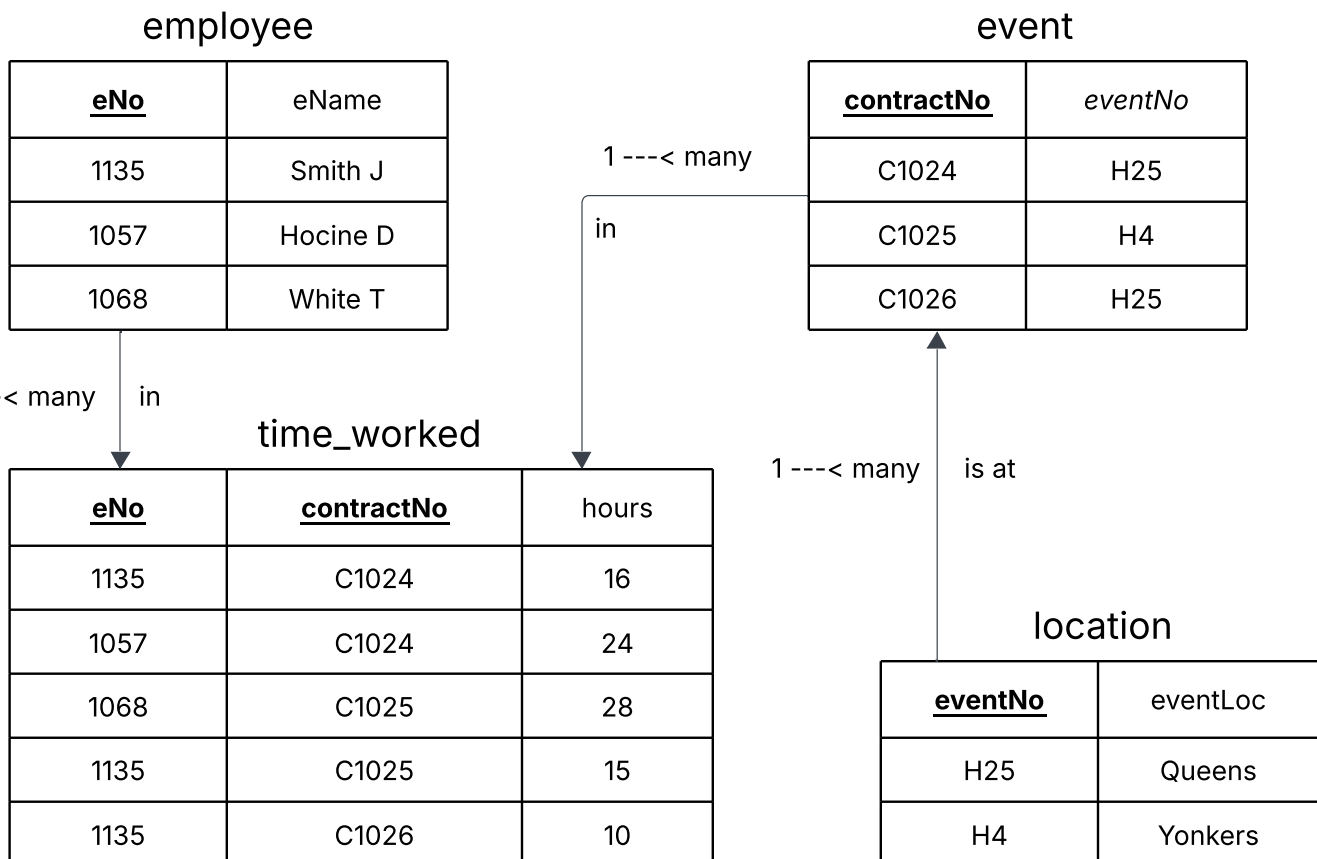
These partial dependencies need to be resolved for by creating new tables in order to bring this table into the "2NF" form.

2NF



After resolving the partial dependencies, we have now normalized our data into 2NF. However, there exists a transitive dependency in our tables that has not been resolved. eventLoc is dependent on eventNo, but eventNo is not a primary key in a table. As a result, we need to create a new table that illustrates this relationship.

3NF



We have now resolved our transitive dependencies, meaning that we have normalized the data to the form 3NF. The foreign keys and relationships have been labeled properly.