Step 1: Look at the raw table
Attributes:
customerNumber, customerName, customerType, date, time, employee, partNumber, partName, type, cageCode, quantityOrdered, unitPrice

Step 2 — Define important items
- **Primary Key (PK):**
  customerNumber, date, time, partNumber
- **Functional Dependencies (FD):**
- customerNumber → customerName, customerType
- partNumber → partName, type, cageCode, unitPrice
- {customerNumber, date, time} → employee
- {customerNumber, date, time, partNumber} → quantityOrdered
- **Partial Dependencies:**
- customerNumber → customerName, customerType
- partNumber → partName, type, cageCode, unitPrice
- {customerNumber, date, time} → employee
- **Transitive Dependencies:**
- None in this table.

Happy Supplies
Parts Warehouse

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

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

Step 3: Find candidate keys
Candidate key:
{customerNumber, date, time, partNumber}

Step 4: Check Normal Forms
1NF:
Have PK and each cell has atomic values.

Convert to table

customerName	customerNumber	customerType	date	time	employee	partNumber	name	type	cageCode	quantityOrdered	unitPrice
Jeff Peterson	HG54587	consumer	7/1/2024	10:30am	D.Harrison	10654	Float Control	plumbing	G413	4	12
Jeff Peterson	HG54587	consumer	7/1/2024	10:30am	D.Harrison	10456	Modulator	electrical	H433	3	7
Jeff Peterson	HG54587	consumer	7/1/2024	10:30am	D.Harrison	10776	Hose Assembly	plumbing	G413	7	9
Jeff Peterson	HG54587	consumer	7/1/2024	10:30am	D.Harrison	10657	Float Assembly	plumbing	G413	5	10

Assumption
customerNumber uniquely identifies a customer; it determines customerName and customerType.
partNumber uniquely identifies a part; it determines name, type, cageCode, and unitPrice
One employee will serve the whole order and help one customer at a time (won't be serving both customer same time)
unitPrice comes from Part (no discounts/taxes)

To get 2NF:
Problem: Partial dependency
- **customerNumber** → customerName, customerType
- **partNumber** → name, type, cageCode, unitPrice
- **{customerNumber, date, time}** → employee

Those attributes depend on **part** of the big composite key
Fix: Remove these partial dependencies by creating separate tables for those attributes.

customer

customerNumber (PK)	customerName	customerType
HG54587	Jeff Peterson	consumer

part

partNumber (PK)	name	type	cageCode	unitPrice
10654	Float Control	plumbing	G413	12
10456	Modulator	electrcial	H433	7
10776	House Assembly	plumbing	G413	9
10657	Float Assembly	plumbing	G413	10

order

customerNumber (PK,FK)	date (PK)	time (PK)	employee
HG54587	7/1/2024	10:30am	D. Harisson

order_detail

customerNumber (PK,FK)	date (PK,FK)	time (PK,FK)	partNumber (PK,FK)	quantityOrdered
HG54587	7/1/2024	10:30am	10654	4
HG54587	7/1/2024	10:30am	10456	3
HG54587	7/1/2024	10:30am	10776	7
HG54587	7/1/2024	10:30am	10657	5

It's now in 2NF:
The schema is now in Second Normal Form (2NF).
After the 2NF, there are **no transitive dependencies**—in every relation, each non-key attribute depends **only** on its table's key and not on another non-key.
- **customer:** customerNumber → {customerName, customerType}.
- **part:** partNumber → {name, type, cageCode, unitPrice}.
- **order:** {customerNumber, date, time} → {employee}.
- **order_detail:** {customerNumber, date, time, partNumber} → {quantityOrdered}.

Since no non-primary key field depends on another non-PK (NO transitive dependency), it transfers directly to 3NF.

Step 1: Look at the raw table
Attributes:
staffNo, therapistName, patNo, patName, appointmentDateTime, branchNo

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

Step 2: Define important terms
Primary key (PK):
staffNo, appointmentDateTime)
Candidate key:
(staffNo, appointmentDateTime)
Functional dependency (FD):
staffNo → therapistName, patNo → patName,
(staffNo, appointmentDateTime) → branchNo, patNo

Step 3: Find candidate keys
Candidate key:
(staffNo, appointmentDateTime)

Step 4: Check Normal Forms
Not 1NF:
Have PK but not all columns are atomic. Appointment date and time should be seperated.

To get 1NF:
Problem: Multi-valued attributes; not all columns are atomic
Fix: Separate appointmentDate and appointmentTime, we are able to reach 1NF

appointment

staffNo	therapistName	patNo	patName	appointmentDate	branchNo	appointmentTime
S1011	Fred Smith	P100	Lily White	9/12/2022	M15	10:00
S1011	Fred Smith	P105	Jill Baker	9/12/2022	M15	12:00
S1024	Heidi Pierce	P108	Andy Mckee	9/12/2022	Q10	10:00
S1024	Heidi Pierce	P108	Andy Mckee	9/14/2022	Q10	14:00
S1032	Richard Levin	P105	Jill Baker	9/14/2022	M15	16:30
S1032	Richard Levin	P110	Jimmy Winter	9/15/2022	B13	18:00

To get 2NF:
Problem: Partial dependency — some non-key attributes depend on only part of the composite (staffNo → therapistName, staffNo, appointmentDate) → branchNo)
Fix: Remove these partial dependencies by creating separate tables for those attributes.

therapist

staffNo (PK)	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levin

branch

branchNo (PK)
M15
Q10
B13

staff_assignment

staffNo (PK,FK)	branchNo (FK)	appointmentDate (PK,FK)
S1011	M15	9/12/2022
S1011	M15	9/12/2022
S1024	Q10	9/12/2022
S1024	Q10	9/14/2022
S1032	M15	9/14/2022
S1032	B13	9/15/2022

appointment

staffNo (PK)	patNo	appointmentDate (PK)	appointmentTime (PK)	patName
S1011	P100	9/12/2022	10:00	Lily White
S1011	P105	9/12/2022	12:00	Jill Baker
S1024	P108	9/12/2022	10:00	Andy Mckee
S1024	P108	9/14/2022	14:00	Andy Mckee
S1032	P105	9/14/2022	16:30	Jill Baker
S1032	P110	9/15/2022	18:00	Jimmy Winter

It's now in 2NF:
The schema is now in Second Normal Form (2NF). Every non-key attribute in **Appointment** depends on the *entire* composite identifier. Attributes that depended on only part of that identifier were split out—therapistName under **Therapist(staffNo)**, and the day-level branchNo under **Staff_Assignment(staffNo, appointmentDate)**.

To get 3NF
Problem: A **transitive dependency** exists. Factor out the dependency
Fix: patNo → patName into a separate **Patient(patNo, patName)** relation and keep only patNo in **Appointment**. At this point, non-key attributes are fully and non-transitively dependent on their keys.

therapist

staffNo (PK)	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levin

patient

patNo (PK)	patName
P100	Lily White
P105	Jill Baker
P108	Andy Mckee
P110	Jimmy Winter

branch

branchNo (PK)
M15
Q10
B13

appointment

staffNo (PK)	patNo	appointmentDate (PK)	appointmentTime (PK)
S1011	P100	9/12/2022	10:00
S1011	P105	9/12/2022	12:00
S1024	P108	9/12/2022	10:00
S1024	P108	9/14/2022	14:00
S1032	P105	9/14/2022	16:30
S1032	P110	9/15/2022	18:00

staff_assignment

staffNo (PK,FK)	branchNo (FK)	appointmentDate (PK,FK)
S1011	M15	9/12/2022
S1011	M15	9/12/2022
S1024	Q10	9/12/2022
S1024	Q10	9/14/2022
S1032	M15	9/14/2022
S1032	B13	9/15/2022

It's in 3NF
- No **partial** dependencies (every non-key depends on the whole key).
- No **transitive** dependencies (e.g., patName is no longer in Appointment; it lives in Patient).
- Every attribute is fully dependent on **the key, the whole key, and nothing but the key**.

Step 1: Look at the raw table
Attributes:
eNo, contractNo, hours, eName, eventNo, eventLoc

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

Step 2: Define important terms
Primary key (PK):
eNo, contractNo
Partial dependency:
eNo → eName , contractNo → eventNo
Functional dependency (FD):
- eNo → eName
- contractNo → eventNo
- eventNo → eventLoc
- {eNo, contractNo} → hours

Transitive dependency:
contractNo → eventNo → eventLoc

Step 3: Find candidate keys
Candidate key:
({eNo, contactNo)

Step 4: Check Normal Forms
1NF (Eliminate repeating groups)
Already satisfied. Each cell has atomic values.

To get 2NF:
Problem: Partial dependency — some non-key attributes depend on only part of the composite (eNo → eName, contractNo → eventNo)
Fix: Split out attributes that depend on only part of the key. Create employee table (eNo, eName), create contract table (contractNo, eventNo), keep work (eNo, contractNo, hours)

employee

eNo	eName
1135	Smith J
1057	Hocine D
1068	White T

contract

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

work

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

It's now in **2NF**:
The primary key for **Employee** is **eNo**, which uniquely identifies each employee; the primary key for **Contract** is **contractNo**, which uniquely identifies each contract; and the primary key for **Work** is **(eNo, contractNo)**, which together uniquely identify each record of hours worked by an employee on a specific contract.

To get 3NF:
- **Problem: A transitive dependency** exists — eventLoc is determined by eventNo, which in turn depends on contractNo, instead of depending directly on the full primary key (eNo, contractNo). This means eventLoc is indirectly related to the key.
- **Fix:** Eliminate the transitive dependency by creating a separate **Event** table to store eventNo and eventLoc, and let the **Contract** table reference eventNo as a foreign key.

employee

eNo (PK)	eName
1135	Smith J
1057	Hocine D
1068	White T

contract

contractNo (PK)	eventNo (FK)
C1024	H25
C1025	H4
C1026	H25

work

eNo (PK,FK)	contractNo(PK, FK)	hours
1135	C1024	16
1057	C1024	24
1068	C1025	28
1135	C1025	15
1135	C1026	10

event

eventNo (PK)	eventLoc
H25	Queens
H4	Yonkers

It's now in 3NF
All PK:
- **Employee:** eNo
- **Event:** eventNo
- **Contract:** contractNo
- **Work:** (eNo, contractNo)

Rule (3NF): Every non-key attribute now depends directly on the primary key and nothing else. There are no partial or transitive dependencies.
- **Employee(eNo, eName):**
  eName depends only on eNo, the primary key.
- **Event(eventNo, eventLoc):**
  eventLoc depends only on eventNo, the primary key.
- **Contract(contractNo, eventNo):**
  eventNo depends only on contractNo, the primary key.
- **Work(eNo, contractNo, hours):**
  hours depends on the whole composite key (eNo, contractNo) and nothing else.