Exercise 1 – Happy Supplies Parts Warehouse

a. Assumptions

- 1. **customerNo** uniquely identifies a customer (customerName, customerType).
- 2. An order is uniquely identified by **(customerNo, orderDate, orderTime)** — who placed it and when.
- 3. **partNo** uniquely identifies a part (partName, partType, cageCode, unitPrice). unitPrice is a property of the part, not the order line.
- 4. **employeeName** (e.g. "D. Harrison") is the only employee identifier on the form; no employee number is present.
- 5. **cageCode** is a physical storage location attribute of a part and does not independently determine partType.
- 6. A single order can contain multiple line items. **qtyOrdered** is specific to each order–part combination.
- 7. The same customer can place multiple orders on different dates/times.

Step 1 – Unnormalized Form (UNF)

All data from the form placed in one flat relation. The **repeating group** is the line-item section (partNo, partName, partType, cageCode, qtyOrdered, unitPrice).

customerNo	customerName	customerType	orderDate	orderTime	employeeName	partNo	partName	partType	cageCode	qtyOrdered	unitPrice
HG54587	Jeff Peterson	Consumer	7/1/2024	10:30am	D. Harrison	10654	Float Control	Plumbing	G413	4	12
HG54587	Jeff Peterson	Consumer	7/1/2024	10:30am	D. Harrison	10456	Modulator	Electrical	H433	3	7
HG54587	Jeff Peterson	Consumer	7/1/2024	10:30am	D. Harrison	10776	Hose Assembly	Plumbing	G413	7	9
HG54587	Jeff Peterson	Consumer	7/1/2024	10:30am	D. Harrison	10657	Float Assembly	Plumbing	G413	5	10

Partial dependencies identified (2NF violations):

- customerNo → customerName, customerType *(depends only on customerNo)*
- (customerNo, orderDate, orderTime) → employeeName *(depends on order header subset of PK)*
- partNo → partName, partType, cageCode, unitPrice *(depends only on partNo)*
- (customerNo, orderDate, orderTime, partNo) → qtyOrdered *(full PK dependency ✓)*

Step 3 – Second Normal Form (2NF)

Remove all partial dependencies. Split into **Customer**, **Order**, **Part**, and **OrderLine**.

Customer

customerNo	customerName	customerType
HG54587	Jeff Peterson	Consumer

Order

customerNo	orderDate	orderTime	employeeName
HG54587	7/1/2024	10:30am	D. Harrison

Part

partNo	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

OrderLine

customerNo	orderDate	orderTime	partNo	qtyOrdered
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

No transitive dependencies found. 2NF = 3NF for this exercise.

- cageCode does NOT solely determine partType — the matching types for G413 are coincidental, not a functional dependency.
- No other transitive dependencies exist in Customer, Order, or OrderLine.

Final 3NF Relations

Customer

customerNo, customerName, customerType

Order

customerNo, **orderDate**, **orderTime**, employeeName

FK: customerNo → Customer

Part

partNo, partName, partType, cageCode, unitPrice

OrderLine

customerNo, ***orderDate***, ***orderTime***, ***partNo***, qtyOrdered

FK: (customerNo, orderDate, orderTime) → Order; partNo → Part

Exercise 2 – Panacea Mental Health Corporation

a. Assumptions

- 1. **staffNo** uniquely identifies a therapist (therapistName).
- 2. **patNo** uniquely identifies a patient (patName).
- 3. An appointment is uniquely identified by **(patNo, appointmentDate, appointmentTime)** — a patient can only be in one place at a given date/time slot.
- 4. Each appointment involves exactly one therapist and takes place at exactly one branch.
- 5. On any given date, a therapist works at only one branch: **(staffNo, appointmentDate)** → **branchNo**.
- 6. branchNo uniquely identifies a branch. No additional branch attributes are present in the data.

- 7. The same patient may have appointments with different therapists and on different dates.

b–c. Normalization Process

Step 1 – First Normal Form (1NF)

The table is already flat with no repeating groups — it is already in 1NF. **PK = (patNo, appointmentDate, appointmentTime).**

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

Partial dependencies identified (2NF violations):

- patNo → patName *(depends only on patNo, a subset of the PK)*
- staffNo is determined by the full PK; therapistName and branchNo transitively depend via staffNo — handled at 3NF step.

Step 2 – Second Normal Form (2NF)

Remove the partial dependency: patName depends only on patNo. Extract **Patient** table.

Patient

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

Appointment

patNo	appointmentDate	appointmentTime	staffNo	therapistName	branchNo
P100	9/12/2022	10:00	S1011	Fred Smith	M15
P105	9/12/2022	12:00	S1011	Fred Smith	M15
P108	9/12/2022	10:00	S1024	Heidi Pierce	Q10
P108	9/14/2022	14:00	S1024	Heidi Pierce	Q10
P105	9/14/2022	16:30	S1032	Richard Levin	M15
P110	9/15/2022	18:00	S1032	Richard Levin	B13

Transitive dependencies identified in Appointment (3NF violations):

- staffNo → therapistName (therapistName transitively depends on PK via staffNo)
- (staffNo, appointmentDate) → branchNo (branchNo transitively depends on PK via staffNo + appointmentDate)

Step 3 – Third Normal Form (3NF)
Remove transitive dependencies. Extract **Staff** (staffNo → therapistName) and **StaffSchedule** (staffNo, appointmentDate → branchNo).

Patient

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

Staff

staffNo	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levin

StaffSchedule

staffNo	appointmentDate	branchNo
S1011	9/12/2022	M15
S1024	9/12/2022	Q10
S1024	9/14/2022	Q10
S1032	9/14/2022	M15
S1032	9/15/2022	B13

Appointment

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

d. Final 3NF Relations

Patient
patNo, patName

Staff
staffNo, therapistName

StaffSchedule
staffNo, **appointmentDate**, branchNo
FK: staffNo → Staff

Appointment
patNo, ***appointmentDate***, **appointmentTime**, *staffNo*
FK: patNo → Patient; (staffNo, appointmentDate) → StaffSchedule

Exercise 3 – Maid Better Temp Agency

a. Assumptions

- 1. **eNo** uniquely identifies an employee (eName).
- 2. **contractNo** uniquely identifies a contract; each contract applies to exactly one event.

- 3. **(eNo, contractNo)** uniquely identifies a row — an employee can work on multiple contracts, and a contract can involve multiple employees.
- 4. **hours** represents the total hours worked by a specific employee on a specific contract.
- 5. **eventNo** uniquely identifies an event. **eventLoc** is a property of the event (not the contract).
- 6. eventLoc depends on eventNo, not directly on contractNo — evidenced by H25 always being in Queens regardless of contract.

b–c. Normalization Process

Step 1 – First Normal Form (1NF)

The table is already flat with no repeating groups — it is already in 1NF. **PK = (eNo, contractNo).**

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

Partial dependencies identified (2NF violations):

- eNo → eName (depends only on eNo, not the full PK)
- contractNo → eventNo, eventLoc (depend only on contractNo, not the full PK)
- (eNo, contractNo) → hours (full PK dependency)

Step 2 – Second Normal Form (2NF)

Remove partial dependencies. Split into **Employee**, **Contract**, and **WorkAssignment**.

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

WorkAssignment

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

Transitive dependency identified in Contract (3NF violation):

- contractNo → eventNo → eventLoc *(eventLoc depends on eventNo, not directly on contractNo)*

- Evidence: H25 always maps to Queens; H4 always maps to Yonkers — regardless of contract.

Step 3 – Third Normal Form (3NF)
Remove the transitive dependency by extracting eventLoc into a separate **Event** table. Contract retains eventNo as a FK.

Employee

eNo	eName
1135	Smith J
1057	Hocine D
1068	White T

Event

eventNo	eventLoc
H25	Queens
H4	Yonkers

Contract

contractNo	eventNo
C1024	H25
C1025	H4
C1026	H25

WorkAssignment

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

d. Final 3NF Relations

Employee
eNo, eName

Event
eventNo, eventLoc

Contract
contractNo, *eventNo*
FK: eventNo → Event

WorkAssignment
eNo, ***contractNo***, hours
FK: eNo → Employee; contractNo → Contract