Excercise 1

- **Primary Key (PK):** minimal attribute set that uniquely identifies a row.
- **Candidate Key:** any minimal attribute set that could serve as PK.
- **Functional Dependency (FD):** X → Y means X determines Y.
- **Partial Dependency:** non-key attribute depends on part of a composite key.
- **Transitive Dependency:** PK → A and A → B implies PK → B indirectly.

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

Assumption
- CustomerNumber uniquely identifies a Customer.
- PartNumber uniquely identifies a Part.
- A specific order is uniquely identified by (CustomerNumber, OrderDate, OrderTime).
- An order may contain multiple parts.
- CageCode identifies the storage location of a part.
- UnitPrice is determined by PartNumber.
- Employees may serve any customer.

Attributes:
- CustomerNumber
- CustomerName
- CustomerType
- OrderDate
- OrderTime
- EmployeeName
- PartNumber
- PartName
- PartType
- CageCode
- QuantityOrdered
- UnitPrice

OrderLine_1NF

customerNumber PK	customerName	customerType	**orderDate PK**	**orderTime PK**	employeeName	**partNumber PK**	partName	partType	cageCode	quantityOrdered	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

Step 1 (1NF)
All attributes are atomic and there are no repeating groups.
Therefore, the table is in 1NF.

Primary Key (Composite):
(CustomerNumber, OrderDate, OrderTime, PartNumber)
Because:
- One order can contain multiple parts.
- PartNumber is required to uniquely identify each row.

Step 2 (Identify Functional Dependencies)
- CustomerNumber → CustomerName, CustomerType
- PartNumber → PartName, PartType, CageCode, UnitPrice
- (CustomerNumber, OrderDate, OrderTime) → EmployeeName
- (CustomerNumber, OrderDate, OrderTime, PartNumber) → QuantityOrdered

Step 3 (Check 2NF)
2NF Rule:
No non-key attribute may depend on part of a composite primary key.
The primary key is composite, so we check for partial dependencies.
Violations:
- CustomerName and CustomerType depend only on CustomerNumber (part of the PK) → Partial dependency.
- PartName, PartType, CageCode, UnitPrice depend only on PartNumber (part of the PK) → Partial dependency.
- EmployeeName depends only on (CustomerNumber, OrderDate, OrderTime), not the entire PK → Partial dependency.

Part_2NF

partNumber PK	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

PartsOrderLine_2NF

customerNumber PK	**orderDate PK**	**orderTime PK**	**partNumber PK**	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

Customer_2NF

customerNumber PK	customerName	customerType
HG54587	Jeff Peterson	Consumer

PartsOrder_2NF

customerNumber PK	**orderDate PK**	**orderTime PK**	employeeName
HG54587	7/1/2024	10:30am	D. Harrison

Step 5 (3NF)
There are no additional splits needed from 2NF → 3NF for Exercise 1, because there are no transitive dependencies left. So the relations are the same

Step 4 (2NF Fix))
Split by determinants:
- customerNumber → customerName, customerType (Create Customer)
- partNumber → partName, partType, cageCode, unitPrice (Create Part)
- (customerNumber, orderDate, orderTime) → employeeName (Create PartsOrder)

Part_3NF

partNumber PK	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

A part can appear in 0: order lines. Each order line refers to exactly 1 part.
Cardinality:
Part (1) —— (0:*) OrderLine

appears in

PartsOrderLine_3NF

customerNumber PK FK	**orderDate PK FK**	**orderTime 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

An order can contain 1: order lines. Each order line belongs to exactly 1 order.
Cardinality:
Order (1) —— (1:*) OrderLine

contains

PartsOrder_3NF

customerNumber PK FK	**orderDate PK**	**orderTime PK**	employeeName
HG54587	7/1/2024	10:30am	D. Harrison

Customer_3NF

customerNumber PK	customerName	customerType
HG54587	Jeff Peterson	Consumer

One customer may place many orders.

A customer can have 0: orders, and each order belongs to exactly 1 customer.
Cardinality:
Customer (1) —— (0:*) Order

Excercise 2

Panacea Mental Health Corporation

staffNo	theraplistName	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

Assumption
- staffNo uniquely identifies a therapist.
- patNo uniquely identifies a patient.
- An appointment occurs at exactly one date + time, with exactly one therapist and one patient, at exactly one branch.
- A patient may have multiple appointments per day and may see multiple therapists.
- A therapist may work at many branches over time, but on any given day a therapist works at only one branch. So (staffNo, appointmentDate) → branchNo
- A therapist cannot have two different patients at the same appointment date+time. Therefore (staffNo, appointmentDate, appointmentTime) identifies the appointment slot for that therapist.

TherapistAppointment_1NF

staffNo PK	therapistName	patNo	patName	**appointmentDate PK**	**appointmentTime PK**	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

Attributes:
- staffNo
- therapistName
- patNo
- patName
- appointmentDate
- appointmentTime
- branchNo

Step 1 (1NF)
All attributes are atomic and there are no repeating groups.
Therefore, the table is in 1NF.

Primary Key (Composite):
(staffNo, appointmentDate, appointmentTime)
- staffNo alone is not unique (therapist has many appointments)
- patNo alone is not unique (patient has many appointments)
- (appointmentDate, appointmentTime) not unique (many appointments occur then)
- For a given therapist, a time slot is unique:

Step 2 (Identify Functional Dependencies)
- staffNo → therapistName
- patNo → patName
- (staffNo, appointmentDate, appointmentTime) → patNo
- (staffNo, appointmentDate) → branchNo
- Transitive pattern later: staffNo, appointmentDate, appointmentTime) → patNo → patName

Step 3 (Check 2NF)
2NF requires:
- Already in 1NF
- No non-key attribute depends on only part of a composite PK.

PK = (staffNo, appointmentDate, appointmentTime)
Violations:
- therapistName depends only on staffNo
  staffNo is part of the PK → partial dependency
- branchNo depends on (staffNo, appointmentDate)
  missing appointmentTime → depends on a proper subset of PK → partial dependency

Appointment_2NF

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

Step 4 (2NF Fix)
Split by determinants:
- staffNo → therapistName (Create Therapist)
- (staffNo, appointmentDate) → branchNo (Create TherapistDay.)

Keep appointment facts in Appointment_2NF.

Therapist_2NF

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

ThearpistDayBranch_2NF

staffNo PK	**appointmentDate PK**	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

Step 5 (3NF)
3NF Check (Transitive Dependencies)
3NF requires:
- Already in 2NF
- No transitive dependency (non-key depends on non-key).

Violation of 3NF for Appointment_2NF:
- patNo → patName
- (staffNo, appointmentDate, appointmentTime) → patName transitively
- patName depends on the PK indirectly through patNo.
- This is a transitive dependency

Step 6 (3NF Fix)
Transitive issue in Appointment_2NF: patNo → patName, so move Patient out.

Patient_3NF

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

A patient can have 0: appointments. Each appointment belongs to exactly 1 patient.
Cardinality:
Patient (1) —— (0:*) Appointment

attends

Therapist_3NF

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

A therapist can work 0: days. Each therapist-day record belongs to exactly 1 therapist.
Cardinality:
Therapist (1) —— (0:*) ThearpistDayBranch

works on

Appointment_3NF

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

contains

ThearpistDayBranch_3NF

staffNo PK FK	**appointmentDate PK**	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

A therapist-day can have 0: appointments. Each appointment belongs to exactly 1 therapist-day.
Cardinality:
ThearpistDayBranch (1) —— (0:*) Appointment

Excercise 3

Maid Better temp agency

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

Assumption
- eNo uniquely identifies each employee.
- Each contractNo applies to exactly one event.
- Each eventNo occurs at exactly one location.
- An employee may work on multiple contracts
- A contract may involve multiple employees
- The number of hours worked depends on both the employee and the contract.

Attributes:
- eNo
- contractNo
- hours
- eName
- eventNo
- eventLoc

Step 1 (1NF)
All attributes are atomic and there are no repeating groups.
Therefore, the table is in 1NF.

Primary Key (Composite): (eNo, contractNo)
- eNo alone is not unique (employee works on many contracts)
- contractNo alone is not unique (contract involves multiple employees)
- For a given employee and contract combination, the record is unique

Record_1NF

eNo PK	**contractNo PK**	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 (Identify Functional Dependencies)
- eNo → eName
- contractNo → eventNo
- eventNo → eventLoc
- (eNo, contractNo) → hours
- Transitive pattern later: contractNo → eventNo → eventLoc

Step 3 (Check 2NF)
2NF requires:
- Already in 1NF
- No partial dependency (non-key depends on part of composite PK)

PK = (eNo, contractNo)
Violations:
- eNo → eName → depends only on part of PK → Partial dependency
- contractNo → eventNo → depends only on part of PK → Partial dependency

Therefore, table is not in 2NF.

Employee_2NF

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

Contract_2NF

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

WorkAssignment_2NF

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

Step 4 (2NF Fix)
Create
- Employee(eNo PK, eName)
- Contract(contractNo PK, eventNo, eventLoc)
- WorkAssignment(eNo PK/FK, contractNo PK/FK, hours)

Step 5 (3NF)
3NF Check (Transitive Dependencies)
3NF requires:
- Already in 2NF
- No transitive dependency (non-key depends on non-key).

Violation of 3NF for Contract_2NF:
- contractNo → eventNo. eventNo → eventLoc
- contractNo → eventLoc (transitively)

Step 6 (3NF Fix)
Split Contract into:
- Event(eventNo PK, eventLoc
- Contract(contractNo PK, eventNo FK)

Final 3NF Relations:
- Employee(eNo PK, eName)
- Event(eventNo PK, eventLoc)
- Contract(contractNo PK, eventNo FK)
- WorkAssignment(eNo PK/FK, contractNo PK/FK, hours))

A employee can have 0: work assignments. Each work assignment belongs to exactly 1 employee.
Cardinality:
Employee (1) —— (0:*) WorkAssignment

works on

Employee_3NF

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

WorkAssignment_2NF

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

includes

A contract can have 0: work assignments. Each work assignment belongs to exactly 1 contract.
Cardinality:
Contract (1) —— (0:*) WorkAssignment

Event_3NF

eventNo PK	eventLoc
H25	Queens
H4	Yonkers

A event can have 0: contracts. Each contract belongs to exactly 1 event.
Cardinality:
Event (1) —— (0:*) Contract

has

Contract_3NF

contractNo PK	eventNo FK
C1024	H25
C1025	H4
C1026	H25