

# Assignment4 Normalization

Zien zz5106

## Exercise 1

### (a). Assumptions

- 1: CageCode identifies storage location and depends only on PartNumber
- 2: PartNumber identifies a part uniquely
- 3: An order is uniquely identified by CustomerNumber, Date, Time.
- 4: CustomerNumber identifies customer uniquely
- 5: Unitprice, PartName, PartType depend only on PartNumber
- 6: Employees have unique names, employee name identifies the staff who processed the order.
7. CustomerType depends only on CustomerNumber.
8. QuantityOrdered depends on Customer, Part, date and time.

### (b). Normalization Process

#### UnnormalizedForm:

Customer Name	Customer Number	Customer Type	Date	Time	Employee	Part Number	Part Name	Part Type	Cage Code	Quantity Ordered	Unit Price
Jeff Peterson	HG54587	Consumer	7/1/2024	10:30am	D.Harrison	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

#### Step1-1NF

Table: OrderForm

Customer Name	<u>customer Number</u>	customerType	<u>date</u>	<u>time</u>	employee	<u>partNumber</u>	partName	partType	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

#### Why:

The unnormalized form had a hierarchical structure with repeated groups of parts under one order header.

To achieve 1NF, we need to create a flat table where each row represents one order line item. So there are no repeating groups and every cell has an atomic value.

The primary keys are **customerNumber**, **partNumber**, **date**, **time**, they uniquely identify each order line.

Step2-2NF

Table: Customer

<b><u>customerNumber</u></b>	customerName	customerType
HG54587	Jeff Peterson	Consumer

Table: Order

<b><u>customerNumber</u></b>	<b><u>date</u></b>	<b><u>time</u></b>	employee
HG54587	7/1/2024	10:30am	D.Harrison

Table: Part

<b><u>partNumber</u></b>	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

Table: OrderLine

<b><u>customerNumber</u></b>	<b><u>date</u></b>	<b><u>time</u></b>	<b><u>partNumber</u></b>	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

Why:

1NF contained partial dependencies that

CustomerName and CustomerType depend only on CustomerNumber,

Employee depends on CustomerNumber, Date and Time,

PartName, PartType, CageCode, and UnitPrice depend only on PartNumber

Only QuantityOrdered depends on the full primary key.

To reach 2NF, we need to decompose the 1NF table into four tables based on the functional dependencies and the primary key(s) each table needs to have. So that these data only need to be stored once.

Step3-3NF

Table: Customer

<b><u>customerNumber</u></b>	<b>customerName</b>	<b>customerType</b>
HG54587	Jeff Peterson	Consumer

Table: Order

<b><u>customerNumber</u></b>	<b><u>date</u></b>	<b><u>time</u></b>	<b>employee</b>
HG54587	7/1/2024	10:30am	D.Harrison

Table: Part

<b><u>partNumber</u></b>	<b>partName</b>	<b>partType</b>	<b>cageCode</b>	<b>unitPrice</b>
10654	Float Control	Plumbing	G413	12
10456	Modulator	Electrical	H433	7
10776	Hose Assembly	Plumbing	G413	9
10657	Float Assembly	Plumbing	G413	10

Table: OrderLine

<b><u>customerNumber</u></b>	<b><u>date</u></b>	<b><u>time</u></b>	<b><u>partNumber</u></b>	<b>quantityOrdered</b>
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

Why:

No transitive dependencies exist, so no further decomposition is needed.

(c) Primary and Foreign Keys

1NF:

Table: orderForm

PK: (customerNumber, partNumber, date, time)

2NF:

Table: Customer

PK: customerNumber

Table: Order

PK: (customerNumber, date, time)

Table: Part

PK: partNumber

Table: OrderLine

PK: (customerNumber, partNumber, date, time)

3NF:

Table: Customer

PK: customerNumber

Table: Order

PK: (customerNumber, date, time)

FK: customerNumber

Table: Part

PK: partNumber

Table: OrderLine

PK: (customerNumber, partNumber, date, time)

FK: customerNumber, partNumber, date, time

(d) names of 3NF relations:

Customer, Part, Order, Orderline

## Exercise 2

(a). Assumptions

- 1: staffNo uniquely identifies each therapist, therapist name depends only on staffNo
- 2: branchNo uniquely identifies each branch
- 3: patNo uniquely identifies each patient, patient name depends only on patNo
- 4: A patient can have multiple appointments, possibly with different therapists
- 5: A therapist can only be at one branch on any given days

(b). Normalization Process

Unnormalized Form

<u>staffNo</u>	therapistName	<u>patNo</u>	patName	<u>appointment date time</u>	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

Step1-1NF:

Table: AppointmentAll

<u>staffNo</u>	therapistName	<u>patNo</u>	patName	<u>appointmentDate</u>	<u>appointmentTime</u>	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

Why:

The unnormalized form violated 1NF because the appointment date time column contains non-atomic data, so we need to split it into appointmentDate and appointmentTime to ensure all attributes contain atomic values.

Step2-2NF:

Table: Therapist

<u>staffNo</u>	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levin

Table: Patient

<u>patNo</u>	patName
P100	Lily White
P105	Jill Baker
P108	Andy McKee
P110	Jimmy Winter

Table: Appointment

<u>staffNo</u>	<u>patNo</u>	<u>appointmentDate</u>	<u>appointmentTime</u>	branchNo
S1011	P100	9/12/2022	10:00	M15
S1011	P105	9/12/2022	12:00	M15
S1024	P108	9/12/2022	10:00	Q10
S1024	P108	9/14/2022	14:00	Q10
S1032	P105	9/14/2022	16:30	M15
S1032	P110	9/15/2022	18:00	B13

Why:

1NF table contained partial dependencies, which will cause redundancy that therapists and patient names are repeated for each appointment. By creating separate tables, we can eliminate redundancy and store these data only once.

Step3-3NF:

Table: Therapist

<u>staffNo</u>	therapistName
S1011	Fred Smith
S1024	Heidi Pierce
S1032	Richard Levin

Table: Patient

<u>patNo</u>	patName
P100	Lily White
P105	Jill Baker
P108	Andy McKee
P110	Jimmy Winter

Table: TherapistAssignment

<u>staffNo</u>	<u>appointmentDate</u>	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

Table: Appointment

<u>staffNo</u>	<u>patNo</u>	<u>appointmentDate</u>	<u>appointmentTime</u>
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

Why:

2NF's appointment table contained a transitive dependency.

Because branchNo depends on staffNo and appointmentDate, branchNo is transitively dependent on the primary key through (staffNo,appointmentDate).

So we create a therapistAssignment table to store each therapist's branch assignment per day, so the appointment table can delete the branchNo column, and references the assignment through (staffNo, appointmentDate).

(c) Primary and Foreign Keys

1NF:

Table: AppointmentAll

PK: (staffNo, patNo, appointmentDate, appointmentTime)

2NF:

Table: Therapist

PK: staffNo

Table: Patient

PK: patNo

Table: Appointment

PK: (staffNo, patNo, appointmentDate, appointmentTime)

3NF:

Table: Therapist

PK: staffNo

Table: Patient

PK: patNo

Table: TherapistAssignment

PK: (staffNo, appointmentDate)

FK: staffNo

Table: Appointment)

PK: (staffNo, patNo, appointmentDate, appointmentTime)

FK: (staffNo, appointmentDate), patNo, StaffNo

(d) names of 3NF relations:

Therapist, Patient, TherapistAssignment, Appointment

### Exercise 3

(a). Assumptions

- 1: Different contracts can exist for the same event
- 2: Each contract applies to only one event, contractNo uniquely identifies each contract
- 4: Employee name depends only on eNo, eNo uniquely identifies each employee
- 5: Event location depends only on eventNo
6. EventNo uniquely identifies each event
7. Hours worked depends on both employee and contract

(b). Normalization Process

Unnormalized Form

<u>eNo</u>	<u>contractNo</u>	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

Step1-1NF:

Table: Appointment

The Unnormalized Form already satisfies 1NF because all attributes contain atomic values and there are no repeating groups.

Step2-2NF:

Table: Employee

<u>eNo</u>	eName
1135	Smith J
1057	Hocine D
1068	White T

Table: Contract

<u>contractNo</u>	eventNo	eventLoc
C1024	H25	Queens
C1025	H4	Yonkers
C1026	H25	Queens

Table: Assignment:

<u>eNo</u>	<u>contractNo</u>	hours
1135	C1024	16
1057	C1024	24
1068	C1025	28
1135	C1025	15
1135	C1026	10

Why:

The 1NF table contained partial dependencies causing redundancy that the same employee's name might be repeatedly mentioned for each of his work, the location queens and yonkers are also repeated. We can separate Employee and Contract tables to eliminate this redundancy.

Step3-3NF:

Table: Employee

<u>eNo</u>	eName
1135	Smith J
1057	Hocine D
1068	White T

Table: Event

<u>eventNo</u>	eventLoc
H25	Queens
H4	Yonkers

Table: Contract

<u>contractNo</u>	eventNo
C1024	H25
C1025	H4
C1026	H25

Table: Assignment:

<u>eNo</u>	<u>contractNo</u>	hours
1135	C1024	16
1057	C1024	24
1068	C1025	28
1135	C1025	15
1135	C1026	10

Why:

The 2NF Contract table contained a transitive dependency that eventLoc depends on eventNo, but not directly on contractNo.

This means if multiple contracts exist for the same event, the event location will be repeated several times in different records. So we need to create an event table to store each event's location once.

#### (c) Primary and Foreign Keys

1NF:

Table: Unnormalized Form

PK: eNO, contractNo

2NF:

Table: Employee

PK: eNo

Table: Contract

PK: contractNo

Table: Assignment

PK: (eNO, contractNo)

3NF:

Table: Employee

PK: eNo

Table: Event

PK: eventNo

Table: Contract

PK: contractNo

FK: eventNo

Table: Assignment

PK: (eNO, contractNo)

FK: eNO, contractNo

#### (d) names of 3NF relations:

Employee, Event, Contract, Assignment