Abir Mahmood

Database Design and Implementation

Dr. David

Assignment 3

Question 1

a. Assumptions:

- All customers are identified by an individual customer number.
- No specific customers are linked to the employees
- The storage location for every part is shown in the corresponding cage code.
- Every part is stored in a certain type/category and is identified by an individual part number.
- The price per unit is fixed for all parts.
- Many different parts may be part of a single order
- One order is associated with one customer only.

b.Normalization Process:

Unnormalized Form (UNF):

Customer Number	**Customer Name**	**CustomerType**	**PartNumber**	**PartName**	**PartType**	**Cage Code**	**QuantityOrdered**	**UnitPrice**

H654587	Jeff Peterson	Consumer	10654	Float Control	Plumbing	G413	4	12
H654587	Jeff Peterson	Consumer	10456	Modulator	Electrical	H433	3	7
H654587	Jeff Peterson	Consumer	10776	Hose Assembly	Plumbing	G413	7	9
H654587	Jeff Peterson	Consumer	10657	Float Assembly	Plumbing	G413	5	10

First Normal Form (1NF):

- The table is already in 1NF as there are no repeating groups.
- **Primary Key: (CustomerNumber, PartNumber)**

Second Normal Form (2NF):

- Remove partial dependencies.
- Split into separate tables.

Customer Table:

CustomerNumber	**CustomerName**	**CustomerType**

HG54587	Jeff Peterson	Consumer

Primary Key: CustomerNumber

Part Table:

PartNumber	**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

Order Table:

CustomerNo	**PartNo**	**QuantityOrdered**
HG54587	10654	4
HG54587	10456	3
HG54587	10776	7
HG54587	10657	5

Primary Key: (CustomerNo, PartNumber)

Foreign Keys: CustomerNo (references Customer),

Third Normal Form (3NF):

- No transitive dependencies
- Already achieved by separating part details into a separate table.
- Final Relations:
    - Customer(CustomerNumber, CustomerName, CustomerType)
    - Part(PartNumber, PartName, PartType, CageCode, UnitPrice)
    - Order(CustomerNumber, PartNumber, QuantityOrdered)

Primary and Foreign Keys in 3NF:

- Customer Table: **Primary Key: CustomerNumber**
- Part Table: **Primary Key: PartNumber**
- Order Table: **Primary Key: (CustomerNumber, PartNumber)**
    - Foreign Keys: *CustomerNumber, PartNumber*

Question 2

a.Assumptions:

1. Every therapist is assigned a designated staffNo.
2. Every patient is assigned a unique patNo
3. A therapist may work at many branches, but may only see patients in one branch at a time
4. Patients can have several appointments on one day with different therapists.
5. Appointment dates and times are captured within appointmentDateTime.
6. The branchNo uniquely identifies a branch

Step 1: Unnormalized Form (UNF)

The data is currently in a single table:

staffNo	therapistName	patNo	patName	appointment DateTime	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	19/12/2022 10:00	Q10

S1024	Heidi Pierce	P108	Andy McKee	19/12/2022 14:00	Q10
S1032	Richard Levin	P105	Jill Baker	19/12/2022 16:30	M15
51032	Richard Levin	p110	Jimmy Winter	9/15/2022 18:00	B13

Step 2: First Normal Form (1NF)

To achieve 1NF, we must ensure that:

- There are no duplicate rows.
- Each field contains only atomic (indivisible) values.

The current table is already in 1NF because all attributes contain atomic values, and there are no repeating groups.

Primary Key (PK) in 1NF: (staffNo, patNo, appointmentDateTime)

Step 3: Second Normal Form (2NF)

To reach 2NF, we must:

- Ensure that there are no partial dependencies (i.e., every non-key attribute must depend on the whole primary key, not just part of it).

Identified Partial Dependencies:

- therapistName depends only on staffNo.

- patName depends only on patNo.
- branchNo depends only on appointmentDateTime, not on staffNo or patNo.

Decomposed Tables for 2NF:

Therapist Table

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

Patient Table

patNo	**patName**
P100	Lilv White
P105	Jill Baker
p108	Andv Mckee
p110	Jimmy Winter

Branch Table

branchNo

M15
010
R13

Appointment Tabl

staffNo	**patNo**	**appointmentDateTime**	**branchNo**
S1011	P100	9/12/2022 10:00	M15
S1011	P105	9/12/2022 12:00	M15
S1024	P108	19/12/2022 10:00	Q10
S1024	P108	19/12/2022 14:00	Q10
S1032	P105	19/12/2022 16:30	M15
S1032	P110	9/15/2022 18:00	B13

Primary Keys in 2NF:

- **Therapist Table: staffNo**
- **Patient Table: patNo**
- **Branch Table: branchNo**
- **Appointment Table: (staffNo, patNo, appointmentDateTime)**

Step 4: Third Normal Form (3NF)

To reach 3NF, we must:

- Ensure that there are no transitive dependencies (i.e., non-key attributes must depend only on the primary key and not on other non-key attributes).

Identified Transitive Dependencies:

- therapistName depends on staffNo (which is already in its own table, so this is fine).
- patName depends on patNo (which is also handled in the Patient table).
- branchNo is in the Appointment table but does not depend on staffNo or patNo—it depends on appointmentDateTime.

Thus, no further decomposition is necessary, and the relations are in 3NF.

And final 3NF Relations with Primary and Foreign Keys:

Therapist Table

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

Patient Table

patNo	**patName**

P100	Lily White
P105	Jill Baker
P108	Andv Mckee
p110	Jimmv Winter

Branch Table

branchNo
M15
Q10
B13

Appointment Table

staffNo	**patNo**	**appointmentDateTime**	**branchNo**
S1011	P100	9/12/2022 10:00	M15
S1011	P105	9/12/2022 12:00	M15
S1074	P108	19/12/2022 10:00	Q10
S1024	P108	19/12/2022 14:00	Q10

S1032	P105	19/12/2022 16:30	M15
S1032	P110	9/15/2022 18:00	B13

Primary Keys in 3NF:

- **Therapist Table: staffNo**
- **Patient Table: patNo**
- **Branch Table: branchNo**
- **Appointment Table: (staffNo, patNo, appointmentDateTime)**

Foreign Keys in 3NF:

- *Appointment Table: _branchNo_ (FK referencing Branch)*

Question 3

a. Assumptions:

- Every employee (eNo) can be issued multiple contracts and is uniquely identifiable.
- A contract (contractNo) is linked to a single event, however, it is possible to have multiple contracts for the same event.
- An event (eventNo) is associated with a single venue (eventLoc).
- Aliases of employees (eName) are not unique and therefore, cannot be utilized for identification.
- Hours worked (hours) are kept for each employee and per contract.
- Every contract is issued a distinct number (contractNo).
- Locations stem from eventNo, which means they are functionally reliant on eventNo.

b. Normalization to 3NF:

Unnormalized Form (UNF):

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	C1075	15	Smith J	H4	Yonkers

1135	C1026	10	Smith J	H25	Queens

First Normal Form (1NF):

- The table is already in 1NF since all attributes contain atomic values.

Second Normal Form (2NF):

- Identify the composite primary key: (eNo, contractNo).
- Identify partial dependencies:
    - eName depends only on eNo.
    - eventLoc depends only on eventNo.
- Decompose into separate tables:

1. Employee(eNo, eName)

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

2) Event(eventNo, eventLoc)

eventNo	**eventLoc**
H25	Queens

H4	Yonkers

3) Contract(contractNo, eventNo)

contractNo	**eventNo**
C1024	H25
C1025	H4
C1075	H4
C1026	H25

4) EmployeeContract(eNo, contractNo, hours)

eNo	**contractNo**	**hours**
1135	C1024	16
1057	C1024	24
1068	C1025	28
1135	C1075	15
1135	C1026	10

Third Normal Form (3NF):

- The tables are now in 2NF and also in 3NF as:
    - There is no transitive dependencies.
    - Every non-key attribute is functionally dependent only on the primary key

c. Primary and Foreign Keys:

- Employee Table:
    - **Primary Key: eNo**
- Event Table:
    - **Primary Key: eventNo**
- Contract Table:
    - **Primary Key: contractNo**
    - *Foreign Key: eventNo : Event(eventNo)*
- EmployeeContract Table:
    - **Primary Key: (eNo, contractNo)**
    - *Foreign Keys: eNo : Employee(eNo), contractNo : Contract(contractNo)*

d. Final 3NF Relations:

- Employee(eNo, eName)
- Event(eventNo, eventLoc)
- Contract(contractNo, eventNo)
- EmployeeContract(eNo, contractNo, hours)