MIDDLE EAST TECHNICAL UNIVERSITY

NORTHERN CYPRUS CAMPUS

DEPARTMENT
OF COMPUTER ENGINEERING

CNG 351
Data Management and File Structures
Assignment 3

Team Details

Member 1

Name: Mert Can
Surname: Bilgin
Student ID: 2453025

Member 2

Name: Hami
Surname: Karslı
Student ID: 2453322

PART 1: NORMALISATION

Use Case 1 – MENTOR (Metu EveNT OrganiseR)

There is no multivalued attributes or nested relations, so this table is already 1NF. However, there is a composite primary key FD7, so we can normalize to 2NF.

Event(eventID, eventDate, eventTime, eventLocation, eventManagerSSN, eventOrganizerSSN[FK: Organizer: eventOrganizerSSN], crewSSN[FK: Crew: crewSSN], mainSponsorID[FK: Sponsor: mainSponsorID])

Organizer(eventOrganizerSSN, eventOrganizerPhoneNo, eventOrganizerName, eventOrganizerDOB, eventOrganizerGender, eventOrganizerMail, eventOrganizerAddress)

Sponsor(mainSponsorID, sponsorName, sponsorAmount)

Crew(crewSSN, crewName, crewRole)

Artist(artistEmail, artistName, crewSSN[FK: Crew: crewSSN])

ArtistPrice(eventID[FK: Event: eventID], artistEmail[FK: Artist: artistEmail], artistPrice)

With this schema, the relation is now in 3NF because all non-primary-key attributes are non-transitively dependent on the primary key.

Use Case 2 – NCCCloud

First of all, the table can be represented like this:

User(username, dob, library_connection_token, library_name, game_id, game_name, subscription_id, subscription_type, subscription_price, Session_start_date_time, played_computer_id, played_computer_GPU)

The functional dependencies are:

FD_1 : {username} -> {dob, subscription_id}
FD_2 : {library_connection_token} -> {username, library_name}
FD_3 : {game_id, library_name} -> {game_name}
FD_4 : {subscription_id} -> {subscription_type, subscription_price}
FD_5 : {subscription_type} -> {subscription_price}
FD_6 : {username, Session_start_date_time} -> {game_id, library_name, computer_id}
FD_7 : {computer_id} -> {GPU}

There is no multivalued attributes or nested relations, so this table is already 1NF. However, there is a composite primary key FD3 and FD6, so we can normalize to 2NF.

User(username, subscription_id[FK: Subscription: subscription_id], dob)

Connection(library_connection_token, username[FK: User: username], library_name)

Game(game_id, library_name, game_name)

Subscription(subscription_id, subscription_type[FK: Subscription_Price: subscription_type])

Subscription_Price(subscription_type, subscription_price)

Computer(username[FK: User: username], Session_start_date_time, game_id[FK: Game: game_id], library_name, played_computer_id[FK: GPU: played_computer_id])

GPU(played_computer_id, played_computer_GPU)

Then, we need to ensure that each relation is in 3NF. This means that all the attributes in the relation must depend on the primary key and there should be no transitive dependencies. For example, in the User relation, username is the primary key and dob and subscription_id depend on it directly, so it is in 3NF.

Finally, we need to ensure that each relation is in BCNF. This means that all the attributes in the relation must depend on the candidate key, which is a subset of the primary key that uniquely identifies each record in the relation. For example, in the User relation, the candidate key could be either (username, dob) or (username, subscription ID), and all other attributes depend on this candidate key directly. Therefore, the relations are also in BCNF.

PART 2 – RELATIONAL ALGEBRA

FindJob

Queries

1. List the id, name, surname and telephone number of all members in the system.

$$\prod_{\text{member\_id, name, surname, tel}}(\text{Member})$$

2. List the name and the date of creation of the groups that are "Unlisted" (type) and created by the member with the ID of 1333.

$$\prod_{\text{name, creation\_date}}(\sigma_{\text{type = "Unlisted" and created\_by = 1333}}(\text{Group}))$$

3. List the name, surname, and email address of the Members who created Standard groups after the 1st of January 2022.

$$\prod_{\text{Member.name, surname, email}}(\text{Member} \bowtie_{\text{type = "Standard" and creation\_date > 01/01/2022 and member.id = created\_by}} \text{Group})$$

4. We would like to display all the posts (listing their title and content) that are made by all the connections of the member called "Yeliz Yesilada".

$\prod_{title,\ content}((ConnectionList \bowtie_{ConnectionList.member\_id\ =\ Member.member\_id} (\sigma_{name\ =\ "Yeliz"\ and\ surname\ =\ "Yesilada"}(Member))) \bowtie_{posted\_by\ =\ connection\_member\_id} Post)$

5. There is a company called"Curiosity" and this company would like to receive a report that shows the list of employee names (name/surname) that has created a group, the name of that group and also the manager name/surname of that group.

$\prod_{employeeName,\ employeeSurname,\ groupName,\ managerName,\ managerSurname}((\rho_{employee(employeeName,\ employeeSurname,\ groupName)} \prod_{Member.name,\ surname,\ Group.name}((\sigma_{companyname\ =\ "Curiosity"}(Company)) \bowtie_{works\_for\_company\ =\ company\_id} Member) \bowtie_{created\_by\ =\ member\_id} Group) \bowtie \rho_{(managerName,\ managerSurname)} \prod_{Member.name,\ surname}(Group \bowtie_{member\_id\ =\ manager\_Id} Member))$

6. There is an assessment in the system named as "Introduction to Phyton". We would like to list the name/surname of the members and how many times they took this assessment.

$\prod_{name,\ surname,\ assessment\_taken\_times}((\rho_{(assessment\_taken\_times)}\ AssessmentTaken.member\_id\ \Im_{COUNT\ date}(((\sigma_{name\ =\ "Introduction\ to\ Python"}(Assessment)) \bowtie_{Assessment.assessment\_id\ =\ AssessmentTaken.assessment\_id} AssessmentTaken) \bowtie_{Member.member\_id\ =\ AssessmentTaken.member\_id} Member)) \bowtie (((\sigma_{name\ =\ "Introduction\ to\ Python"}(Assessment)) \bowtie_{Assessment.assessment\_id\ =\ AssessmentTaken.assessment\_id} AssessmentTaken) \bowtie_{Member.member\_id\ =\ AssessmentTaken.member\_id} Member))$