MIDDLE EAST TECHNICAL UNIVERSITY

NORTHERN CYPRUS CAMPUS

DEPARTMENT
OF COMPUTER ENGINEERING

CNG 351
Data Management and File Structures

Assignment 3

Team Members:
Name: Muhammad Somaan
ID: 2528404
Name: Daniil Belousov
ID: 2491827

PART 1: NORMALISATION

Use case 1:

NF1: as we do not have a table with data, we assume that that there are no nested tables, composite attributes or multivalued information. With this assumption we can say that our table is in NF1.

NF2: at current state the table is not in 2NF because not all non-key attributes are yet fully functional dependent on the primary key. To obtain the NF2 from, we divide the tables based on the functional dependences and obtain this tables:

Event

Event_ID	eventDate	EventTime	eventLocation

Organization

Event_ID	Event Manager SSN	E_O_SSN [FK: EventOrginizer]	Crew SSN [FK: Crew]	Artist Email	Sponsor ID [FK: Sponsor]	Artist Name	Artist Price

EventOrganizer

E_O_SSN	PhoneNo	Name	DOB	Gender	Mail	Address

Crew

Crew_SSN	Name	Role

Sponsor

Sponsor_ID	Name	Amount	Event_ID

With this implementation we ensure that all the attributes are based on the key

The given tables also support NF3 because there are no transition dependancies left in the tables.

Use case 2:

NF1: the table is in NF1 because there are no nested tables, composite attributes or multivalued information.

NF2: the table is not in NF2 because not all non-key attributes are yet fully functional dependent on the primary key.

We split the table based on the functional dependences and obtain this tables:

Player

Username	Dob	Library Connection Token	Library Name [FK: Game]	game ID [FK: Game]	Subscription ID	Session start date time	PC ID

Game

Library Name	Game ID	Game Name

Subscription

Subscription ID	Subscription type [FK: Subscription price]

Subscription price

Subscription type	Subscription price

PC_information

Played_ID	GPU

With this implementation we ensure that all the attributes are based on the key and nothing but the key

NF3: the final tables ensure we do not have any transitive dependancies as well as all of the attributes are based on the super key which implies that the relations are both in NF§ and BCNF.

PART 2: RELATIONAL ALGEBRA

1)

$$\Pi_{\text{id, name, surname,tel}}\ \text{Member}$$

By the use of projection we display the required columns

2)

$$\Pi\ \text{name,creationdate}\ (\ \sigma_{\text{type=Unlisted,createdby=1333}}\ \text{Group})$$

Firstly we perform the selection operation to get the full table of the group with only the required type and information over who created it. Lastly we use the projection to display the required columns

3)

$$T1 = \sigma_{\text{type=Standart,date}\leq\text{1st January 2022}}\ \text{Group}$$

$$T2 = \text{Member} \bowtie_{\text{memberID = created by}} T1$$

$$T3 = \Pi_{\text{name, surname,email}}\ T2$$

Firstly we perform the selection operation to exclude the tuples where the type is not standard and the date is earlier than the 1st of January 2022. Secondly, we combine it with the member table where the member Id is equal to the id of the person who created the group. Thirdly we use projection to display the fields that were required

4)

$$T1 = \sigma_{\text{member=“Yeliz Yesilada”}}\ \text{Connection List}$$

$$T2 = \text{Post} \bowtie_{\text{postedby=conectionmemebrID}} T1$$

$$T3 = \Pi_{\text{content, title}}\ T2$$

Firstly, we need to find all the member connections from the connection list for the given member name. Secondly we combine the output of the first operation to the Post table using the Equijoin where the posted by id is equivalent to the conectionId to find all the post that the connected members of the given user have made. Lastly, we use projection to display the relevant columns.

5)

$$T1 = \sigma_{companyName=\text{“Curiosity”}}\ Company$$

$$T2 = Member \bowtie_{CompWorksfor=compID} T1$$

$$T3 = T2 \bowtie_{member=createdby} Group$$

$$T4 = T3 \bowtie_{manager=memberID} Member$$

$$T5 = \Pi_{name,\ surname,groupname,\ managername,managersurname}\ T4$$

Firstly, we have to find all the information for the company in the company relation. Secondly we combine it with the member relation to find the members that work in that family by the use of equijoin based on the Company Id. Thirdly, we connect the member we found by the use if equijoin with the Group relation to see the groups member of that company have created. Next, we combine the result with the member table agin to get the names and the surnames of the manager by the use of equijoin where managers id is equal to members id. Lastly we do the projection to display the relative columns.

6)

$$T1 = \sigma_{name=\text{“Introduction to Phyton”}}\ Assessment$$

$$T2 = assessmentTaken \bowtie_{assesmentID=AssesmnetID} T1$$

$$T3 = Member \bowtie_{memberId=memberID} T2$$

$$T4 = \Pi_{name,\ surname}\ T3$$

$$T5 = {}_{name,\ surname}J_{count}\ T4$$

$$T6 = \rho_{name,\ surname,\ times\text{-}taken}\ T5$$

Firstly, we find the information over the assessment in the table by the use of selection. Next, we have to combine it by the use of equijoin to the assessment taken to see the member ids that have taken the assessment. Then, we combine the result with the member table to get the full information of the members that have taken the assessment. Next, we do the projection to reduce the amount of data we are currently storing. Next, we have to count the number of the attempts of each member, that is why we use aggregation with grouping by the name and surname. Lastly we name the counted column for the ease of reading.