

COMP518 Assignment 2

William Cawley Gelling

201077658

1. Create the appropriate relations to represent the entities and relation ships of the E-R diagram

(1) Strong entities (Entity type that is not existence dependent on some other entity type)

Account (SSN, gender, phoneNumber, email, dateOfBirth)

Organization (name, address)

VirtualMachine (id, capability)

Product (id, time)

Here I am going to rename product(id) to be product (Pid), also rename VirtualMachine (id) to be VirtualMachine (VMid) this gives.

Account (SSN, gender, phoneNumber, email, dateOfBirth)

Organization (name, address)

VirtualMachine (VMid, capability)

Product (Pid, time)

(2) Weak entities (Entity type that is existence dependent on some other entity type)

There are no weak entities.

(3) One to many binary relationship types

Maintains: VirtualMachine (id, capability, name)

name references Organization(name)

we leave the owns and affiliated1 to * relationship until we have delt with the subclasses

(4) One to one binary relationship types

There are non

(5) One to one recursive relationship types

There are non

(6) Superclass/subclass relationship types

Firstly, let's look at the {mandatory, and} relation for accounts meaning the account can be developer and customer, as well as just customer or developer, must be at least one. place discriminators in from customer account and developer account, these will be cAccountFlag, dAccountFlag.

Account (SSN, gender, phoneNumber, email, dateOfBirth, officeAddress, creditCard, homeAddress, cAccountFlag, dAccountFlag)

Discriminators (cAccountFlag, dAccountFlag)

Now we look at the {mandatory, or} relationship involving product and music, book, video, image. As this is mandatory or the product must be one of these but can only be one of these. Following the guidelines, we will have four relations music, book, video and images. This gives us the following relations

Music (Pid, time, type, quality)

Book (Pid, time, author, title)

Video (Pid, time, title, rank)

Image (Pid, time, size, category)

Now we can go back and apply relationship types from earlier

1 to many relationships

Owns: VirtualMachine (VMid, capability, name, SSN, Date)

name refers to Organization(name)

SSN refers to Account (SSN)

Affiliated_to: Account (SSN, gender, phoneNumber, email, dateOfBirth, officeAddress, creditCard, homeAddress, cAccountFlag, dAccountFlag, name)

Discriminators (cAccountFlag, dAccountFlag)

name refers to Organization(name)

(7) Many to many binary relationship types

For many to many relationships, create a new relation to represent the relationship and include any attributes that are part of the relationship. Post

Buys: Buys (SSN, Pid, price, discount)

Primary key: SSN, Pid

Foreign Key: SSN references Account(SSN)

Foreign key: Pid references Music(Pid), Book(Pid), Video(Pid), Image(Pid)

Develops: Develops (SSN, Pid)

Primary key: SSN, Pid

Foreign Key: SSN references Account(SSN)

Foreign key: Pid references Music(Pid), Book(Pid), Video(Pid), Image(Pid)

Runs on: Runs (VMid, Pid)

Primary key: VMid, Pid

Foreign Key: VMid references Account(SSN)

Foreign key: Pid references Music(Pid), Book(Pid), Video(Pid), Image(Pid)

Final relations:

Account (SSN, gender, phoneNumber, email, dateOfBirth, officeAddress, creditCard, homeAddress, cAccountFlag, dAccountFlag, name)

Primary key: SSN

Discriminators (cAccountFlag, dAccountFlag)

Foreign key: name refers to Organization(name)

VirtualMachine (VMid, capability, name, SSN, Date)

Primary key: VMid

Foreign key: name refers to Organization(name)

Foreign key: SSN refers to account (SSN)

Buy (SSN, Pid, price, discount)

Primary key: SSN, Pid

Foreign Key: SSN references Account(SSN)

Foreign key: Pid references Music(Pid), Book(Pid), Video(Pid), Image(Pid)

Develops (SSN, Pid)

Primary key: SSN, Pid

Foreign Key: SSN references Account(SSN)

Foreign key: Pid references Music(Pid), Book(Pid), Video(Pid), Image(Pid)

Runs (VMid, Pid)

Primary key: VMid, Pid

Foreign Key: VMid references Account(SSN)

Foreign key: Pid references Music(Pid), Book(Pid), Video(Pid), Image(Pid)

Music (Pid, time, type, quality)

Book (Pid, time, author, title)

Video (Pid, time, title, rank)

Image (Pid, time, size, category)

We have the schema:

ConferenceData (participantID, participantName, participantAddress, sessionLocation, sessionDate, sessionStartingTime, SessionDuration, topicID, topicName, paperID, paperTitle, SPCID, SPCName)

We can assume this is in first normal form as we are told that without the data the table is flattened and therefore in first normal form. To move from 1NF to 2NF I will, identify the functional dependencies. Identify the primary key for the 1NF relation. And decide if partial dependencies exist on the primary key to remove them by placing them in a new relation along with a copy of their determinant.

participantID → participantName, participantAddress (partial dependency)

sessionDate, topicID → sessionLocation, sessionStartTime, sessionDuration (partial dependency)

topicID → topicName, SPCID, SPCName (partial dependency)

SPCID → SPCName, topicID, topicName (alternate Key/ transitional dependency)

paperID → paperName, topicID, topicName, SPCID, SPCName (partial dependency)

from this we take the primary key to be (topicID, sessionDate, paperID, participantID)

These are the dependencies I have chosen. You could argue the dependencies for paperID to include sessionLocation, sessionDate, sessionStartTime, sessionDuration. But if someone books to see a session, they are not booking to see a paper so therefore a paper could move to a different session, but that person would still be seeing the same session. If paperID were to determine these then the persons session would also switch, which from reading I am assuming is not what happens.

You could argue that sessionDate, PaperID has the dependencies sessionLocation, sessionStartTime, sessionDuration instead of sessionDate and topic. But as with the session above the paper could move to a different session but the person booking would still be going to the original session. If you made the assumption that once a papers session has been set it can't change then these would work but I am assuming papers can change sessions.

If you were to say that once a paperID is assigned to a session it cannot change then you could just have the primary key of paperID and participantID for this relation and this would change the normal forms and the dependencies.

The paperID will have a specific topicID and SPCID but TopicID will have multiple papers and SPCID will have multiple papers. As the SPC has only one topic each SPCID has once topic which is always the same.

You can have a situation where the topicID and topicName has been set but has yet to be assigned a paperID.

So, in first normal form we have

ConferenceData (participantID, sessionDate, topicID, paperID, participantName, participantAddress, sessionLocation, sessionStartingTime, SessionDuration, topicName, paperTitle, SPCID, SPCName)

Moving to second normal form, make new relations so that partial dependencies exist on a primary key

ConferenceData (participantID, sessionDate, topicID, paperID)

Participant (participantID, participantName, participantAddress)

Session (sessionDate, topicID, sessionLocation, sessionStartTime, sessionDuration)

Paper (paperID, paperName, topicID, topicName, SPCID, SPCName)

Topic (topicID, topicName, SPCID, SPCName)

We then take this into third normal form and get create relations for the transitional dependencies giving us

ConferenceData (participantID, sessionDate, topicID, paperID)

Participant (participantID, participantName, participantAddress)

Session (sessionDate, topicID, sessionLocation, sessionStartTime, sessionDuration)

Paper (paperID, paperName, topicID, SPCID)

Topic (topicID, topicName, SPCID)

SPC (SPCID, SPCName)

Here I believe you could remove SPCID from paper as it can be linked to topic as every topicID has the same SPCID so it is a redundant column this gives us

ConferenceData (participantID, sessionDate, topicID, paperID)

Participant (participantID, participantName, participantAddress)

Session (sessionDate, topicID, sessionLocation, sessionStartTime, sessionDuration)

Paper (paperID, paperName, topicID)

Topic (topicID, topicName, SPCID)

SPC (SPCID, SPCName)