PHASE 2: SUBMISSION CHECKLIST/SIGNOFF SHEET

GROUP #: 7 GROUP NAME: DaBest Team

Deliverables:

✓ Requirements Description

✓ ER Diagram with Min/Max Specification

✓ ER Diagram Uncaptured Constraints

✓ Relational Schema with Referential Integrity, each table should have its primary key underlined

✓ Functional Dependencies and Minimal Universal Key, Universal Schema including all attributes

✓ Transaction Processing Needs: categorized with brief description

□ Forms □ Reports □ Queries

Assessment:

✓ Group Status Report

	Printed Name	*ASUEmail*
Phase Leader	Michelle Zelechowski	mzelecho@asu.edu
Phase Recorder	Kevin Jackson	ktjacks1@asu.edu
Phase Checker	John Shaeffer	jdshaeff@asu.edu
Technical Advisor 1	Andrew Bland	abland4@asu.edu
Technical Advisor 2	Abdulrahman Altuwijry	aaltuwij@asu.edu

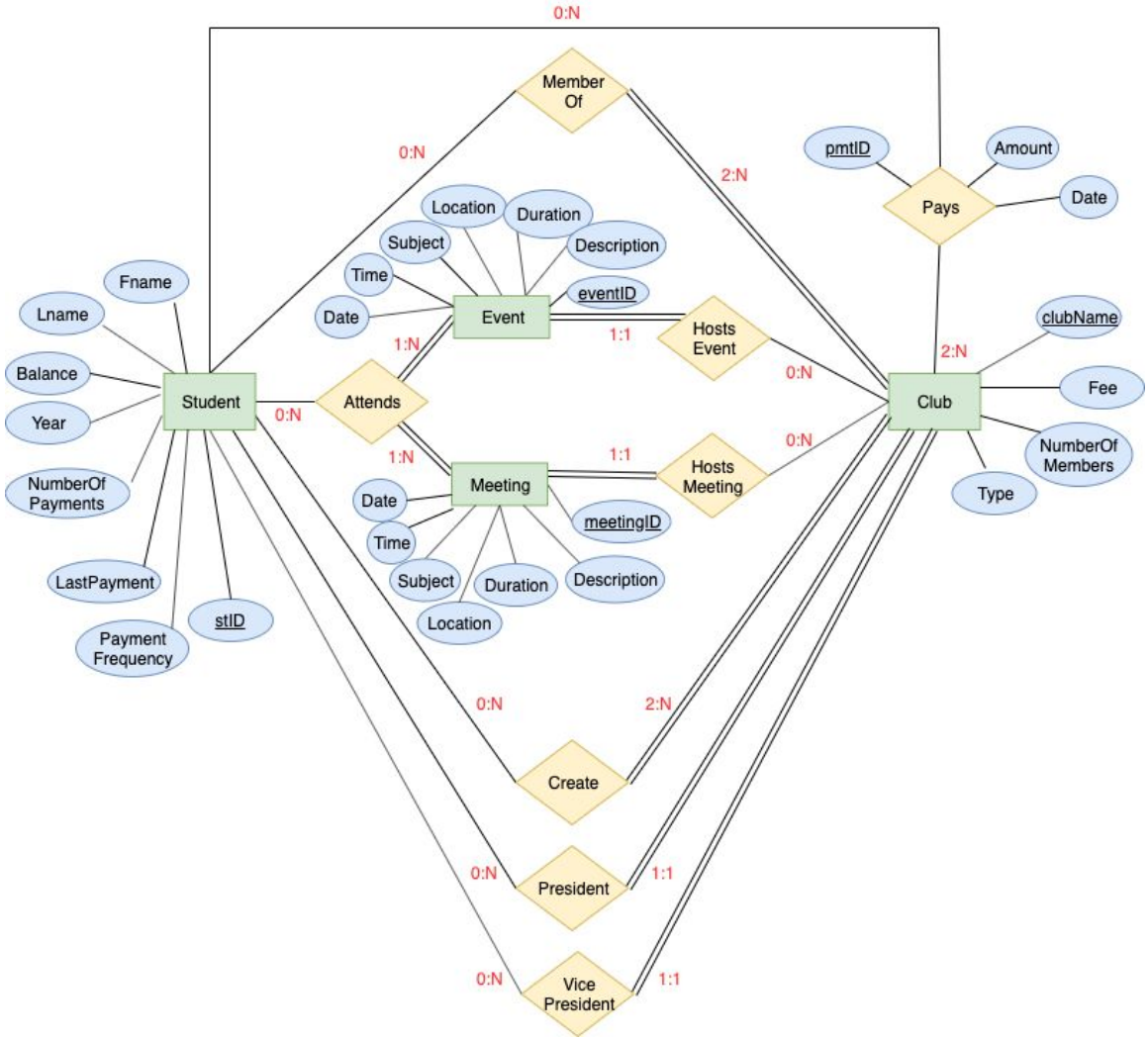
Table Of Contents

Requirements and Description

The college stores information about students, clubs, meetings, events, and student payments. The following data have been identified in the requirements collection and analysis phase and they are to be represented in the enterprise:

- There are many student clubs at the college. Each club has a unique name, club type, student fee, and member count. A club must be founded by at least two students (and subsequently contain at least the two founders as members). A club must also have one Vice President and one President. Not all students belong to a club. A club can host meetings (which only members can attend) and events (which are open to all students). A club does not have to host meetings or events, but can host as many of each as it likes.
- Students do not have to belong to a club, but can belong to as many as they like. If a student does belong to a club, they must pay the student fee. Each student must have a name (consisting of first and last), payment balance, last payment date, payment frequency, count of payments made, year in college, and unique student ID. A student can be a President or Vice President for as many clubs as they like. A student can attend meetings for clubs for which they are a member. They can attend any event they'd like.
- Students pay clubs to compensate for membership. Payments are for a specific amount, the date is recorded, and each payment has a unique ID.
- Meetings have a date/time, subject, location, description, duration, and unique meeting ID. Only one club hosts a meeting and there is no limit on the number of members that can attend. Only members can attend meetings. At least one member attends each meeting.
- Events have a date/time, subject, location, description, duration, and unique event ID. Only one club hosts an event and there is no limit on the number of members that can attend. At least one student attends each event.

ER Diagram


0:N
Member Of
0:N
2:N
pmtID
Amount
Pays
Date
Location
Duration
Subject
Description
Time
eventID
Date
Event
1:N
1:1
Hosts Event
0:N
2:N
clubName
Fname
Lname
Balance
Year
Student
0:N
Attends
1:N
1:1
Hosts Meeting
0:N
Club
Fee
NumberOf Members
Type
NumberOf Payments
Meeting
Date
Time
Subject
Duration
Description
meetingID
Location
LastPayment
stID
Payment Frequency
0:N
2:N
Create
0:N
President
1:1
0:N
Vice President
1:1

ER Diagram Uncaptured Constraints

The following is a list of constraints that are not captured by the ER diagram:

- You have to be a member to attend a meeting
- Student cannot attend a meeting or event if they have not paid their balance
- STUDENT.year must be a value existing in the following set {Freshman, Sophomore, Junior, Senior}
- STUDENT.numberOfPayments >= 0
- STUDENT.paymentFrequency must be value existing in following set {weekly, monthly, quarterly, annually}
- STUDENT.lastPayment <= current date
- EVENT.duration > 0
- MEETING.duration > 0
- PAYS.date <= current date
- CLUB.fee > 0
- CLUB.type must be a value existing in the following set {athletic, volunteer, academic, hobby, special interest, other}
- PAYS.amount >= 0

Relational Schema

Student (stID, Fname, LName, Balance, Year, NumberOfPayments, LastPayment, Payment Frequency)

Club (clubName, fee, NumberOfMembers, type)

memberOf (stID, clubName)

Pays (pmtID, stID, clubName, Amount, Date)

Create (stID, clubName)

President (stID, clubName)

Vice President (stID, clubName)

Meeting (meetingID, mtgDescription, mtgDuration, mtgLocation, mtgSubject, mtgTime, mtgDate)

Event (eventID, eventDescription, eventDuration, eventLocation, eventSubject, eventTime, eventDate)

HostsEvent (clubName, eventID)

HostsMeeting (clubName, meetingID)

Attends (stID, eventID)

Functional Dependencies

F = {
FD1: stID → Fname, Lname, Balance, Year, Number of Payments, LastPayment, PaymentFrequency, stID
FD2: clubName → clubName, fee, NumberOfMembers, type
FD3: stID, clubName → stID, clubName
FD4: pmtID → pmtID, stID, clubName, Amount, Date
FD5: clubName → stID, clubName
FD6: meetingID → meetingID, mtgDescription, mtgDuration, mtgLocation, mtgSubject, mtgTime, mtgDate
FD7: eventID → eventID, evntDescription, evntDuration, evntLocation, evntSubject, evntTime, evntDate
FD8: eventID → clubName, eventID
FD9: meetingID → clubName, meetingID
FD10: stID, eventID → stID, eventID
}

Universal Schema:

College(ID/STID, FNAME, LNAME, BALANCE, YEAR, NUMBEROFPAYMENTS, LASTPAYMENT, PAYMENTFREQUENCY, CLUBNAME, FEE, NUMBEROFMEMBERS, TYPE, PMTID, AMOUNT, DATE, MEETINGID, MTGDESCRIPTION, MTGDURATION, MTGLOCATION, MTGSUBJECT, MTGTIME, MTGDATE, EVENTID, EVNTDESCRIPTION, EVNTDURATION, EVNTLOCATION, EVNTSUBJECT, EVNTTIME, EVNTDATE)

The minimal universal key is {STID, CLUBNAME, PMTID, EVENTID, MEETINGID}.

The attribute closure for the minimum universal key shown below shows that the minimal universal key can be used to derive all of the attributes in our universal schema:

Given	STID, CLUBNAME, PMTID, EVENTID, MEETINGID
FD1	FNAME, LNAME, BALANCE, YEAR, NUMBEROFPAYMENTS, LASTPAYMENT, PAYMENTFREQUENCY
FD2	FEE, NUMBEROFMEMBERS, TYPE
FD4	AMOUNT, DATE
FD6	MTGDESCRIPTION, MTGDURATION, MTGLOCATION, MTGSUBJECT, MTGTIME, MTGDATE
FD7	EVNTDESCRIPTION, EVNTDURATION, EVNTLOCATION, EVNTSUBJECT, EVNTTIME, EVNTDATE

Since no subset of the universal key, {STID, CLUBNAME, PMTID, EVENTID, MEETINGID}, forms a key for the universal schema, it can be considered minimal:

- STID is needed to determine the attributes of the student table (FD1)
- CLUBNAME is needed to determine the attributes of the club table (FD2)
- PMTID is necessary to determine the attributes of the payment relation (FD4)
- MEETINGID is necessary to determine the attributes of the meeting table (FD6)
- EVENTID is necessary to determine the attributes of the event table (FD7)

Status Report

GROUP #: 7 **GROUP NAME:** DaBest Team **PHASE #:** 2

We have each reviewed the contents of the following group status report:

	Printed Name	*Signature*
Phase Leader	Michelle Zelechowski	_____
Phase Recorder	Kevin Jackson	
_____		
Phase Checker	John Shaeffer	_____
Technical Advisor 1	Andrew Bland	_____
Technical Advisor 2	Abdulrahman Altuwijry	_____

Dates & attendance at group meetings in this phase:

Tuesday, October 15th 2019 4:30-5:45 All present

Overview of progress on project as of October 17th:

The Phase 2 deliverable has been completed with all below work assigned & accomplished jointly by the team. We are prepared to move forward to the next deliverable, and the team will decide the work division at a later team meeting.

CONTRIBUTIONS OF GROUP MEMBERS

(• indicates completed contribution o indicates contribution to be completed)

Leader:

- Refined & edited ER Diagram to reflect design corrections
- Identified functional dependencies and super key
- o Define department and dept_locations tables in Access

Recorder:

- Identified universal key
- Assisted in design & edit of relational schema
- o Define dependent table in Access

Phase Checker:

- Drafted Group Status report with Technical Advisor 1 for deliverable
- Aided in preparation of final deliverable document
  - Define employee table in Access

Technical Advisor 1:

- Drafted group status report with Phase Checker for deliverable
- Aided in preparation of final deliverable document
  - Define project table in Access

Technical Advisor 2:

- Documented the relational schema with referential integrity
- Aided in preparation of final deliverable document
  - Define Relationships Window in Access