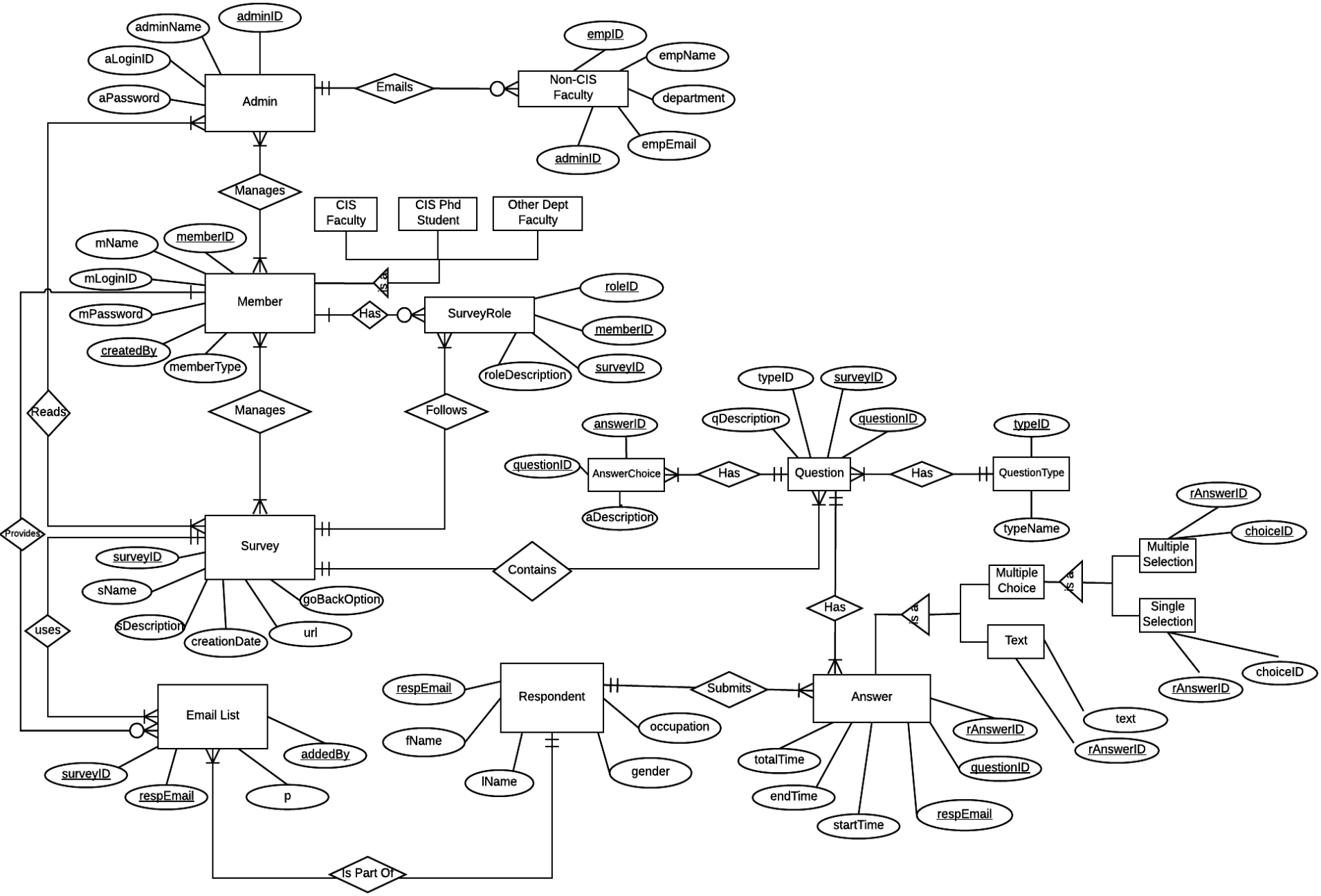
CIS 9340 Team Project
Final Deliverable

Vrushali Akarte

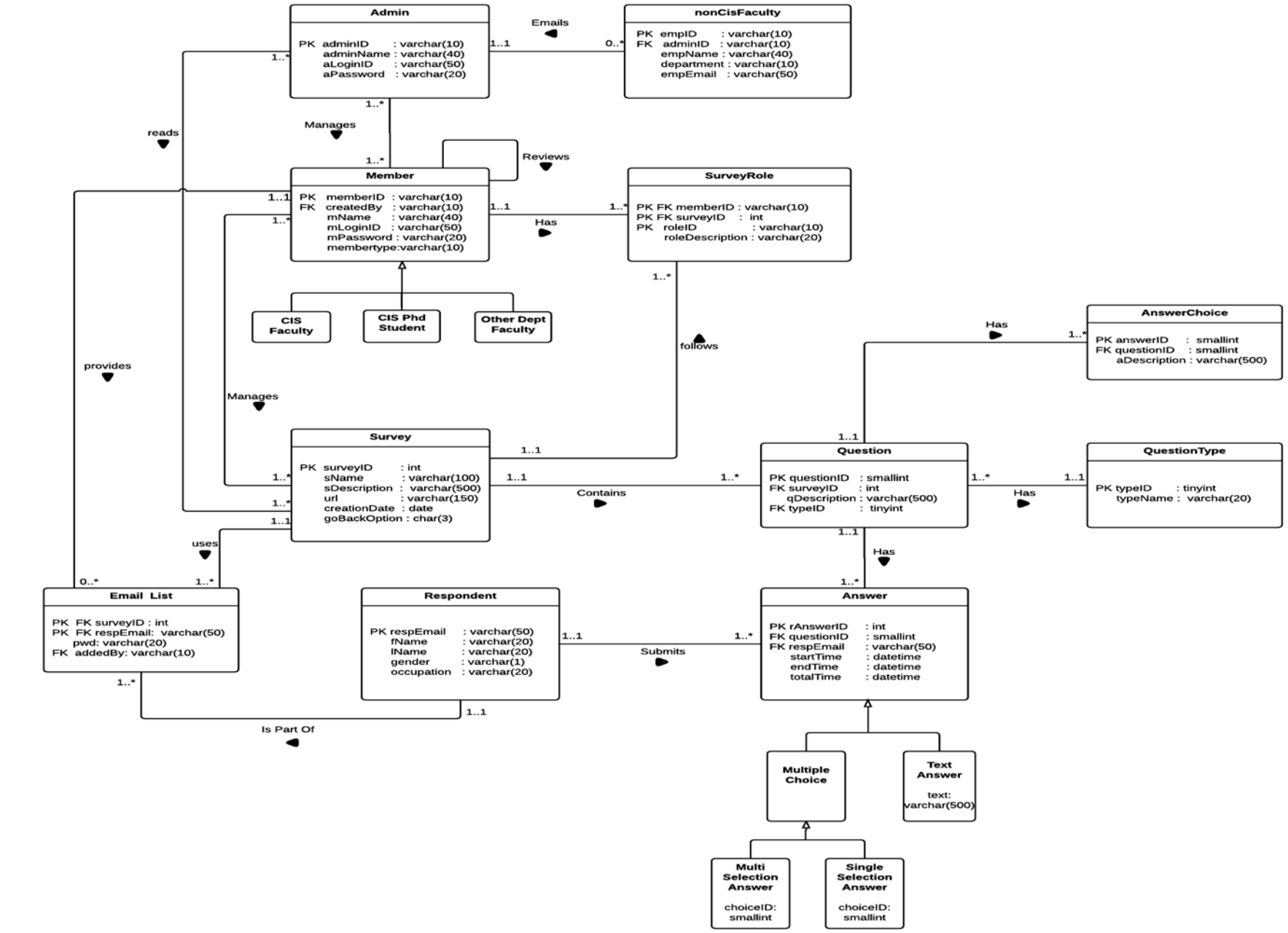
Tanaya Nandanwar

Submitted On: 05/25/2017

1. Systems Analysis - ERD



2. UML Class diagram



3. Normalization

1NF:

1.NON_CIS_FACULTY (empID, empName, department, empEmail, *adminID*(fk))
2.ADMIN (adminID, adminName, aLoginID, aPassword)
3.MEMBER (memberID, mName, mLoginID, mPassword, *createdBy(fk),* memberType)
4.SURVEY_ROLE (*memberID*, *surveyID*, roleID, roleDescription)
5.SURVEY (surveyID, sName, sDescription, url, creationDate, goBackOption)
6.EMAIL_LIST (*surveyID(fk), respEmail(fk),* pwd, *addedby(fk))*
7.QUESTION (questionID, qDescription, *surveyID (fk), typeID(fk)*)
8.QUESTION_TYPE(typeID, typeName)
9.ANSWER_CHOICE (answerID, aDescription, *questionID* (fk))
10.ANSWER (rAnswerID, *questionID(fk), respEmail(fk),* startTime, endTime, totalTime)
11.MULTI_SELECTION_ANSWER(*rAnswerID(fk)*, choiceID)
12.SINGLE_SELECTION_ANSWER(*rAnswerID(fk)*, choiceID)
13.TEXT_ANSWER(*rAnswerID(fk)*, text)
14.RESPONDENT (respEmail, fName, lName, gender, occupation)

3. Normalization

2NF:

FD1: empID → empName, department, empEmail
FD2: adminID → adminName, aLoginID, aPassword
FD3: memberID → mName, mLoginID, mPassword, membertype
FD4: roleID → roleDescription
FD5: surveyID → sName, sDescription, url, creationDate, goBackOption
FD6: questionID → qDescription
FD7: typeID → typeName
FD8: surveyID, respEmail → pwd, addedby
FD9: answerID → aDescription
FD10: rAnswerID → startTime, endTime, totalTime
FD11: respondentID → fName, lName, email, gender, occupation
FD12: rAnswerID → choiceID
FD13: rAnswerID → text

FD4 is partial key dependency for SURVEY_ROLE table. Therefore, we split SURVEY_ROLE into following relations:

SURVEY_ROLE (memberID, surveyID, roleID)
ROLE_TYPE (roleID, roleDescription)

All other relations are already in 2NF since they have no partial key dependencies.

3. Normalization

3NF:

We have following partial key dependencies and split the corresponding relations as shown below:

- adminID → aLoginID
  aLoginID → aPassword
  ADMIN (adminID, adminName, *aLoginID(fk)*)
  ADMIN_CREDENTIALS (aLoginID, aPassword)

- memberID → mLoginID
  mLoginID → mPassword
  MEMBER (memberID, mName, *mLoginID*(fk), *createdBy(fk)*)
  MEMBER_CREDENTIALS (mLoginID, mPassword)

All other relations are in 3NF.

BCNF:

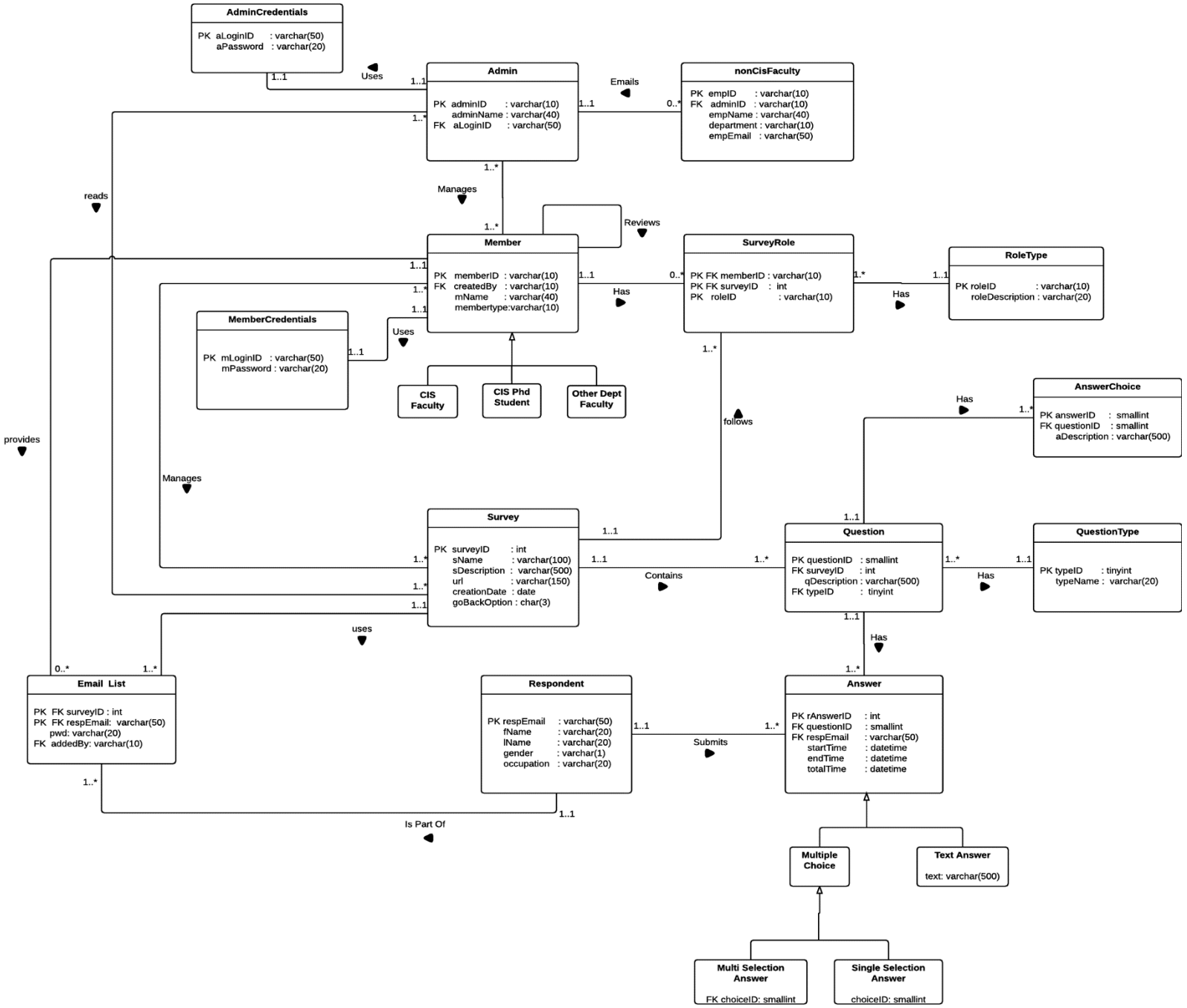
All the determinants are candidate keys and therefore all relations are in BCNF.

3. Normalization

The final normalized relations are:

1.NON_CIS_FACULTY (empID, empName, department, empEmail, *adminID*(fk))
2.ADMIN (adminID, adminName, *aLoginID*(fk))
3.ADMIN_CREDENTIALS (aLoginID, aPassword)
4.MEMBER (memberID, mName, membertype, *mLoginID*(fk), *createdBy(fk)*)
5.MEMBER_CREDENTIALS (mLoginID, mPassword)
6.SURVEY_ROLE (memberID, surveyID, roleID)
7.ROLE_TYPE (roleID, roleDescription)
8.SURVEY (surveyID, sName, sDescription, url, creationDate, goBackOption)
9.EMAIL_LIST (*surveyID(fk), respEmail(fk)*, pwd, *addedby(fk))*
10.QUESTION (questionID, qDescription, *surveyID (fk), typeID (fk)*)
11.QUESTION_TYPE(typeID, typeName)
12.ANSWER_CHOICE (answerID, aDescription, *questionID* (fk))
13.ANSWER (rAnswerID, questionID, *respEmail(fk)*, startTime, endTime, totalTime)
14.MULTI_SELECTION_ANSWER(*rAnswerID(fk)*, choiceID)
15.SINGLE_SELECTION_ANSWER(*rAnswerID(fk)*, choiceID)
16.TEXT_ANSWER(*rAnswerID(fk)*, text)
17.RESPONDENT (respEmail, fName, lName, gender, occupation)

4. After Normalization



5. Database Implementation & Queries

We have created the database using MS SQL Server. Database file, SQL Query files are is submitted along with this ppt.

Following are the SQL Queries we have run successfully on our database:

6.1 SQL SERVER QUERIES AND RESULTANT TABLES

```
--------------------------------------------------------------------------
--QUERIES ON THE TABLES
--------------------------------------------------------------------------
-- 1. Search the email list for survey: 500000001
select * from EMAIL_LIST where surveyID = '500000001';
```

Results | Messages

	surveyID	respEmail	pwd	addedBy
1	500000001	donald.duck@xyz.com	lol@345	90000001
2	500000001	julia.missy@xyz.cof	lol@456	90000001
3	500000001	mark.job@abc.com	lol@123	90000001

```
-- 2. Find the number of question for each survey
select surveyID, count(*) as Question_Count
from QUESTION
group by surveyID;
```

Results | Messages

	surveyID	Question_Count
1	500000001	4
2	500000002	3

```
-- 3. Search for non-CIS Faculty who do not have access to the system
select ncf.empID, ncf.empName, empEmail, mem.memberID
from NON_CIS_FACULTY as ncf left join MEMBER as mem
on ncf.empEmail = mem.mLoginID
where memberID is NULL;
```

Results | Messages

	empID	empName	empEmail	memberID
1	100001	jim corbett	jim.corbett@xyz.com	NULL
2	100003	rachel black	rachel.black@xyz.com	NULL
3	100004	bill green	bill.green@xyz.com	NULL
4	100005	minoca ed	minoca.ed@xyz.com	NULL
5	100006	edward codd	edward.codd@xyz.com	NULL

6.2 SQL SERVER QUERIES AND RESULTANT TABLES

```
-- 4. Search the types of question for survey: 500000002
select surveyId, qDescription, typeName
from QUESTION as Ques join QUESTION_TYPE as Ques_Type
on Ques.typeID = Ques_Type.typeID
where surveyID = '500000002';
```

Results | Messages

	surveyId	qDescription	typeName
1	500000002	What facilities do you most commonly use at the ...	MultiSelection
2	500000002	Do you think the facilitiea are adequate?	SingleSelection
3	500000002	What more do you expect to be provided as faciliti...	Text

```
-- 5. Find members who created the survey list for each survey

select distinct surveyID, m.mName as Email_List_Creator
from EMAIL_LIST join MEMBER as m
on EMAIL_LIST.addedBy = m.memberID;
```

Results | Messages

	surveyID	Email_List_Creator
1	500000001	martin luther
2	500000002	jeff archer

6.3 SQL SERVER QUERIES AND RESULTANT TABLES

```
-- 6. Displaying Members and names of Admins who created those members. Also displaying the surveys
created by those members
select MEMBER.memberID, MEMBER.mName as Member_Name, SURVEY.sName as Survey_Name, ADMIN.adminName as
Mem_Created_By
from MEMBER join ADMIN
on MEMBER.createdBy = ADMIN.adminID
left join SURVEY_ROLE
on MEMBER.memberID = SURVEY_ROLE.memberID and roleID = '10001'
left join SURVEY
on SURVEY.surveyID = SURVEY_ROLE.surveyID;
```

Results | Messages

	memberID	Member_Name	Survey_Name	Mem_Created_By
1	90000001	martin luther	Course Survey	mary kom
2	90000002	benny dayal	Facilities Survey	mark gates
3	90000003	jeff archer	NULL	mini brown
4	90000004	jim corbett	NULL	mary kom
5	90000005	bill maher	NULL	mark gates
6	90000006	ayn rand	NULL	mini brown

6.4 SQL SERVER QUERIES AND RESULTANT TABLES

```
-- 7. Creating a view for Survey with the description for member roles

create view SURVEY_MEMBER_ROLE as (
select s.surveyId as Survey_Id, s.sName as Survey_Name, s.sDescription as Survey_Description,
m.memberID as Member_ID, m.mName as Member_Name, r.roleDescription as Role_Description
from SURVEY as s join SURVEY_ROLE as sr
on s.surveyID = sr.surveyID
join MEMBER as m
on m.memberID = sr.memberID
join ROLE_TYPE as r
on r.roleID = sr.roleID);

select * from SURVEY_MEMBER_ROLE order by survey_id,role_description;
```

Results | Messages

	Survey_Id	Survey_Name	Survey_Description	Member_ID	Member_Name	Role_Description
1	500000001	Course Survey	Student satisfaction with course offered	90000001	martin luther	create
2	500000001	Course Survey	Student satisfaction with course offered	90000002	benny dayal	delete
3	500000001	Course Survey	Student satisfaction with course offered	90000004	jim corbett	modify
4	500000001	Course Survey	Student satisfaction with course offered	90000003	jeff archer	review
5	500000002	Facilities Survey	To find facilities related student reponse	90000002	benny dayal	create
6	500000002	Facilities Survey	To find facilities related student reponse	90000003	jeff archer	delete
7	500000002	Facilities Survey	To find facilities related student reponse	90000002	benny dayal	modify
8	500000002	Facilities Survey	To find facilities related student reponse	90000001	martin luther	review

6.5 SQL SERVER QUERIES AND RESULTANT TABLES

```
-- 8. Creating a view for Members who are creators and the members who review them

create view MEMBER_REVIEWER as (
select CREATOR.surveyid, CREATOR.Creator, REVIEWER.Reviewer from
(select distinct surveyid, mName as Creator
from MEMBER, SURVEY_ROLE, ROLE_TYPE
where MEMBER.memberID = SURVEY_ROLE.memberID
and SURVEY_ROLE.roleID = '10001') as CREATOR,

(select distinct surveyid, mName as Reviewer
from MEMBER, SURVEY_ROLE, ROLE_TYPE
where MEMBER.memberID = SURVEY_ROLE.memberID
and SURVEY_ROLE.roleID = '10004') as REVIEWER
where CREATOR.surveyId = REVIEWER. surveyId
);

select * from MEMBER_REVIEWER order by surveyid;
```

Results | Messages

	surveyid	Creator	Reviewer
1	500000001	martin luther	jeff archer
2	500000002	benny dayal	martin luther

6.6 SQL SERVER QUERIES AND RESULTANT TABLES

```
-- 9. a sub query of 3 tables showing SurveyName, Question description, Respondent's email and what were
the respondent's answers.
select T2.respEmail, s.sName, T2.qDescription, T2.aDescription from Survey as s,
            (select q.surveyID, q.qDescription, T1.aDescription, T1.respEmail from Question as q,
                        (select msa.choiceId, msa.rAnswerID, a.questionID, a.respEmail, ac.aDescription
from MULTI_SELECTION_ANSWER as msa, ANSWER as a,
                        ANSWER_CHOICE as ac
                        where msa.rAnswerID = a.rAnswerID and msa.choiceId = ac.answerId and a.questionId
= ac.questionId
                        UNION
                        select ssa.choiceId, ssa.rAnswerID, a.questionID, a.respEmail, ac.aDescription
from SINGLE_SELECTION_ANSWER as ssa, ANSWER as a,
                        ANSWER_CHOICE as ac
                        where ssa.rAnswerID = a.rAnswerID and ssa.choiceId = ac.answerId and a.questionId
= ac.questionId) T1
            where q.questionID = T1.questionID) T2
where s.surveyID = T2.surveyID;
```

Results | Messages

	respEmail	sName	qDescription	aDescription
1	mark.job@abc.com	Course Survey	Which courses did you complete this semester?	programming
2	mark.job@abc.com	Course Survey	Which courses did you complete this semester?	statistics
3	mark.job@abc.com	Course Survey	Which courses did you complete this semester?	dbms
4	mark.job@abc.com	Course Survey	Would you like to take more such courses?	yes
5	steve.banon@abc.com	Facilities Survey	What facilities do you most commonly use at the ...	library
6	steve.banon@abc.com	Facilities Survey	What facilities do you most commonly use at the ...	computer lab
7	steve.banon@abc.com	Facilities Survey	What facilities do you most commonly use at the ...	printers and scanners
8	steve.banon@abc.com	Facilities Survey	What facilities do you most commonly use at the ...	study rooms
9	steve.banon@abc.com	Facilities Survey	Do you think the facilitiea are adequate?	no

6.7 SQL SERVER QUERIES AND RESULTANT TABLES

```
 -- 10. A join and subquery of 4 tables giving the total time required to complete a survey for each
repondent with survey and respondent names.

select T3.Survey_Name, m.mName as Survey_Created_By,
       T3.Respondent_Name, T3.Respondent_Email, T3.Required_Time
from SURVEY_ROLE as sr, MEMBER as m,
(select T2.surveyID, T2.sName as Survey_Name,  r.fName + '' + r.lName as Respondent_Name, T2.respEmail as
Respondent_Email,T2.Required_Time from Respondent as r,
(select T1.surveyID, s.sName, T1.respEmail, T1.Required_Time from Survey as s,
(select Ques.surveyID, Resp.respEmail,
RIGHT('0' + CAST(SUM(DATEDIFF(SECOND, StartTime, EndTime)) / 3600 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST((SUM(DATEDIFF(SECOND, StartTime, EndTime)) / 60) % 60 AS VARCHAR),2) + ':' +
RIGHT('0' + CAST(SUM(DATEDIFF(SECOND, StartTime, EndTime)) % 60 AS VARCHAR),2) as Required_Time
from Question as Ques
join Answer as Ans on Ques.questionID = Ans.questionID
join Respondent as Respon Resp.respEmail = Ans.respEmail
group by Ques.surveyID, Resp.respEmail) as T1
where s. surveyID = T1.surveyID) as T2
where r.respEmail = T2.respEmail) as T3
where sr.surveyID = T3.surveyID and m.memberID = sr.memberID;
```

Results | Messages

	Survey_Name	Survey_Created_By	Respondent_Name	Respondent_Email	Required_Time
1	Course Survey	martin luther	markjob	mark.job@abc.com	00:03:52
2	Facilities Survey	benny dayal	stevebanon	steve.banon@abc.com	00:03:43
3	Facilities Survey	martin luther	stevebanon	steve.banon@abc.com	00:03:43
4	Course Survey	jim corbett	markjob	mark.job@abc.com	00:03:52
5	Course Survey	benny dayal	markjob	mark.job@abc.com	00:03:52
6	Facilities Survey	benny dayal	stevebanon	steve.banon@abc.com	00:03:43
7	Facilities Survey	jeff archer	stevebanon	steve.banon@abc.com	00:03:43
8	Course Survey	jeff archer	markjob	mark.job@abc.com	00:03:52