CS 586 Introduction to Databases
Graduate Project – Database Implementation
He, Haomin
Ren, Zicheng
11/13/2017

2. ER Diagram and Relational Schema: Produce an ER diagram for your domain, and its translation into a relational schema, including all keys and foreign keys. You should aim for a database with 6 – 10 tables. You should also submit evidence that you have created at least one table from your schema and populated it with at least one row.

Relational Schema:
Primary key (PK), Foreign key (FK)

1.Student_basic table contain all the information about students who want to attend a school overseas.
student_basic (student_id(PK), first, last, email, mobile_phone, alter_phone, wechat, payment, app_status, dob, address)

2.Student_rqst is a weak entity of a student. In reality, a student would come to case manager's office and show his/her interests in studying abroad, which includes basic preference of a student. For instances, what type of school does the student intend to attend? High school? College? In which year does the student want to enroll in the school? 2017 Fall Term? 2018 Summer Term? If the student changes his/her mind and doesn't want to study abroad anymore. We would just delete student information in student_basic table, and associated values in student_rqst table would be gone as well.
student_rqst(student_id(PK&FK), school_type(PK), year, term, region)

3.Submission table indicates all the school applications each student has submitted to the schools. Also this table acts like a checklist (eg. personal statement submitted yet? Yes/no. Transcript submitted yet? Yes/no.) Each student can have more than one submissions because every student can apply multiple schools at the same time.
Submission (appl_id(PK&FK), student_id(PK&FK), deadline, transcript, bank_statement, CV, PS, test_score, recmd_letr)

4.School_applied table shows basic information about schools.
school_applied(appl_id(PK), major, school_name, school_web, state, country, description)

5.Feedback table shows information that if a student gets accepted by a school or not.
feedback(student_id(PK&FK), appl_id(PK&FK), status)

6.Case_manager contains information about company employees.
case_manager(employee_id(PK), first, last, email, phone, division)

7.Each student can consult multiple case managers. Each case manager can help multiple students. This is a many-to-many relationship, we create correspondence table to support this relationship.
correspondence(employee_id(PK&FK), student_id(PK&FK), last_contact_date, contact_method, topic)

8.There are different kinds of tests students need to take before they study abroad, eg. SAT, GMAT, GRE, TOEFL and etc.
test(test_id(PK), test_name)

9.Each student can take multiple tests. Each test can be take by multiple students. This is a many-to-many relationship, we create score_rel table to support this relationship.
score_rel(student_id(PK&FK), test_id(PK&FK), score)

Modifications:
We made some changes of our original design due to some conflicts between the business needs and the design. We also corrected some logical issues between entities.

From the student info entity we pulled out student's request (e.g., when does the student want to get admitted? Which part of the country does the student wish to study in?...) to make a weak entity student_rqst in order to adapt student's studying request. The reason that we made student_rqst a weak entity is because each student might have several requests during the consultation session.

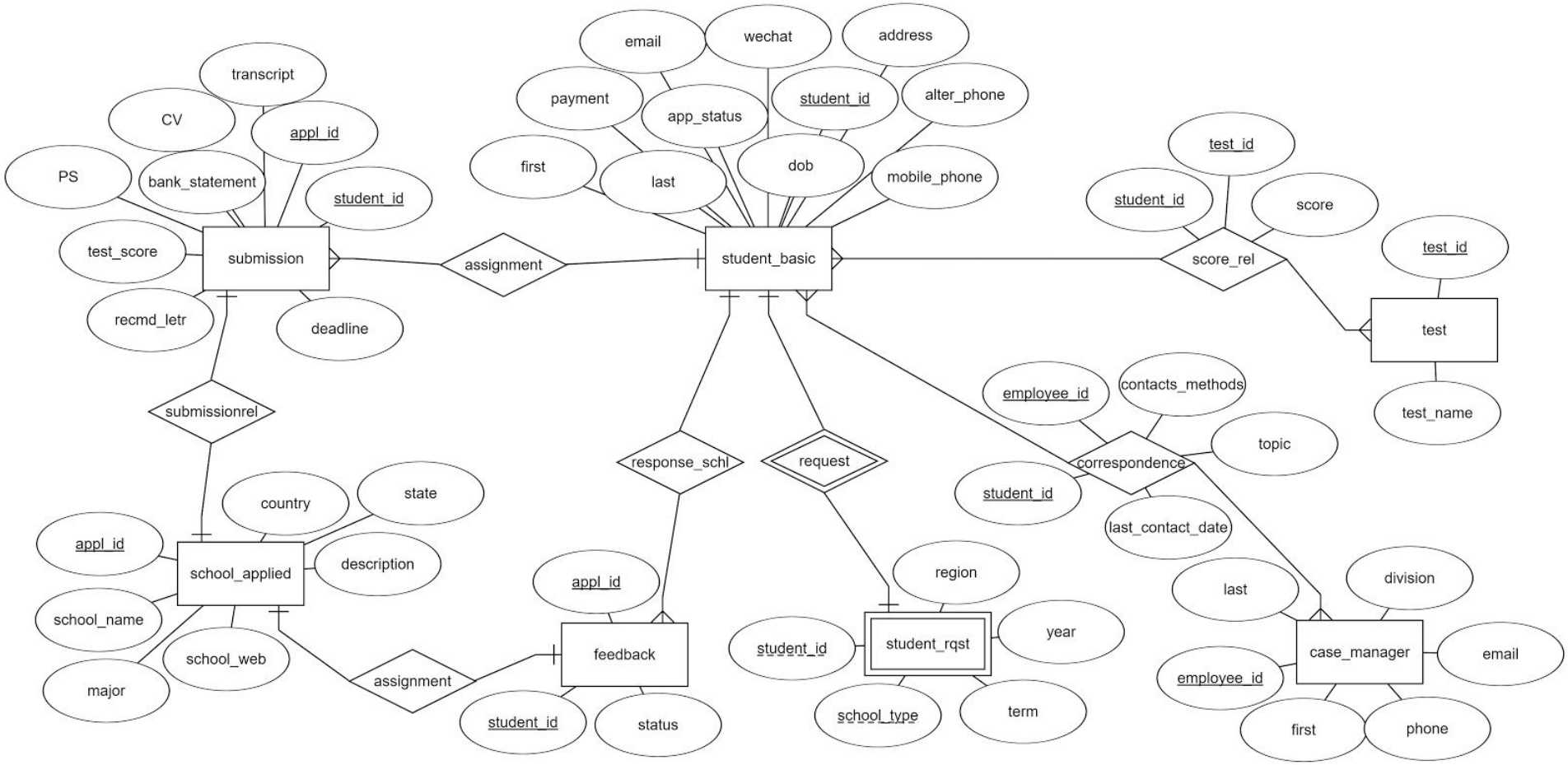
Another change that we made is the entity that hold all the required tests. In order to adapt the changes of test types and each student might be required to take one or several of the tests pool we made a table just for storing all the tests required by all sorts of admission.

We changed the school_id to appl_id in order to uniquely identify the table for the schools that students apply for. Our original design was to use school_id, school_name and major together as primary key.

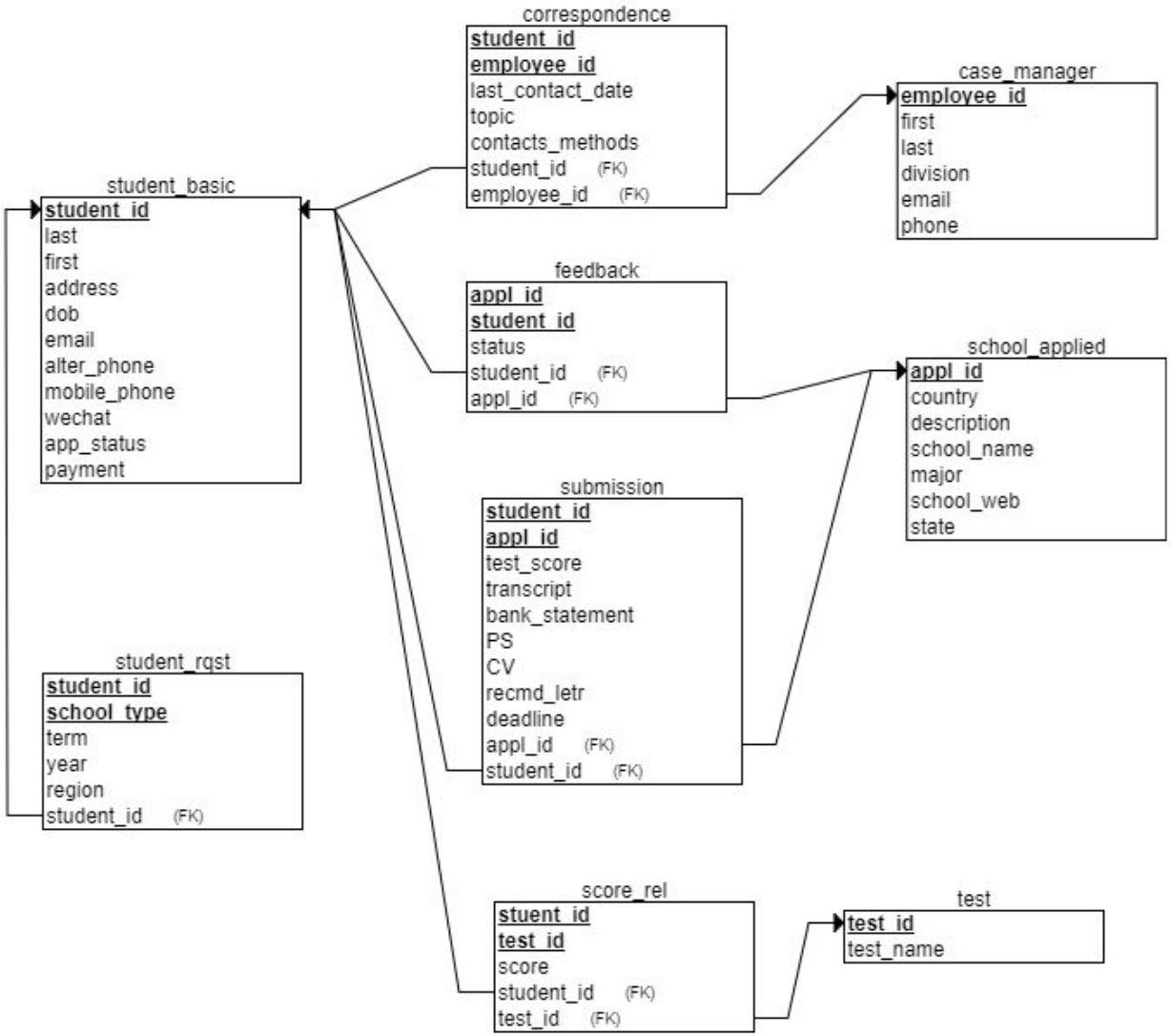
We changed the attributes in the feedback table because it was not necessary to give three individual boolean type of attributes. We replaced them with one attribute status so that we just need to input "accepted", "rejected" or "waiting_list" for inserting the table.

We deleted the application period entity for two reasons: 1) The idea of student's desired admission time might be advanced or pushed off due to all sorts of reasons. It is not crucial information during the consultation period at the agency. We moved the info for application period to the student_rqst entity; 2) Once it's moved onto the application process the deadline is official (and it's defined in the submission entity).

ER Diagram:


email
wechat
address
transcript
payment
app_status
student_id
alter_phone
CV
appl_id
first
last
dob
mobile_phone
PS
bank_statement
student_id
test_id
student_id
score
test_score
submission
assignment
student_basic
score_rel
test_id
recmd_letr
deadline
test
test_name
submissionrel
employee_id
contacts_methods
topic
response_schl
request
correspondence
student_id
last_contact_date
country
state
appl_id
school_applied
description
region
last
division
school_name
appl_id
feedback
student_id
student_rqst
year
case_manager
email
school_web
assignment
employee_id
major
school_type
term
first
phone
student_id
status

Relational Schema:



Tables:

```
CREATE TABLE student_basic (
      student_id int NOT NULL,
      first varchar(40),
      last varchar(40),
      dob date,
      address varchar(255),
      mobile_phone varchar(40),
      alter_phone varchar(40),
      email varchar(40),
      wechat varchar(30),
      appl_status varchar(40),
      payment decimal,
      PRIMARY KEY (student_id)
);
```

Column	Type	Not Null	Default	Constraints	Actions				Comment
student_id	integer	NOT NULL			Browse	Alter	Privileges	Drop	
first	character varying(40)				Browse	Alter	Privileges	Drop	
last	character varying(40)				Browse	Alter	Privileges	Drop	
dob	date				Browse	Alter	Privileges	Drop	
address	character varying(255)				Browse	Alter	Privileges	Drop	
mobile_phone	character varying(40)				Browse	Alter	Privileges	Drop	
alter_phone	character varying(40)				Browse	Alter	Privileges	Drop	
email	character varying(40)				Browse	Alter	Privileges	Drop	
wechat	character varying(30)				Browse	Alter	Privileges	Drop	
appl_status	character varying(40)				Browse	Alter	Privileges	Drop	
payment	numeric				Browse	Alter	Privileges	Drop	

```
INSERT INTO student_basic (student_id, first,      last, dob, address, mobile_phone,
alter_phone, email, wechat, appl_status, payment)
VALUES
(1, 'Zicheng', 'Ren', '1990-11-25', '123 Anhui China', '5236985963', Null,
'renzicheng3@gmail.com', 'hjkiu7896', 'in progress', 2700),
(2, 'Zhao', 'Qian', '1987-11-22', '456 Tokyo Japan', '1257896543', Null, 'sdf5@hdm.com',
'kjgfd4523', 'start', 2700),
(3, 'Sun', 'Li', '1992-5-3', '684 Beijing China', '3658745963', Null, 'shdd@idh.com',
'mgfu5522', 'finish', 2500),
(4, 'Zhou', 'Wu', '1993-5-9', '856 Shanghai China', '3652140596', Null, 's5d@hdk.com',
'nhfs2589', 'in progress', 1600),
(5, 'Zheng', 'Wang', '1994-8-16', '367 Chongqing China', '3003652586', Null,
'sdf3@hdk.com', 'nycs2596', 'start', 2500)
;
```

Actions	student_id	first	last	dob	address	mobile_phone	alter_phone	email	wechat	appl_status	payment
Edit Delete	1	Zicheng	Ren	1990-11-25	123 Anhui China	5236985963	*NULL*	renzicheng3@gmail.com	hjkiu7896	in progress	2700
Edit Delete	2	Zhao	Qian	1987-11-22	456 Tokyo Japan	1257896543	*NULL*	sdf5@hdm.com	kjgfd4523	start	2700
Edit Delete	3	Sun	Li	1992-05-03	684 Beijing China	3658745963	*NULL*	shdd@idh.com	mgfu5522	finish	2500
Edit Delete	4	Zhou	Wu	1993-05-09	856 Shanghai China	3652140596	*NULL*	s5d@hdk.com	nhfs2589	in progress	1600
Edit Delete	5	Zheng	Wang	1994-08-16	367 Chongqing China	3003652586	*NULL*	sdf3@hdk.com	nycs2596	start	2500

5 row(s)

```
CREATE TABLE test (
     test_id int NOT NULL,
     test_name varchar(20),
     PRIMARY KEY (test_id)
);
```

Column	Type	Not Null	Default	Constraints	Actions				Comment
test_id	integer	NOT NULL			Browse	Alter	Privileges	Drop	
test_name	character varying(20)				Browse	Alter	Privileges	Drop	

```
INSERT INTO test (test_id, test_name)
VALUES
(1, 'GRE'),
(2, 'GMAT'),
(3, 'SAT'),
(4, 'IELTS'),
(5, 'TOEFL')
;
```

Actions		test_id	test_name
Edit	Delete	1	GRE
Edit	Delete	2	GMAT
Edit	Delete	3	SAT
Edit	Delete	4	IELTS
Edit	Delete	5	TOEFL

5 row(s)

```
CREATE TABLE score_rel(
     student_id int NOT NULL,
     test_id int NOT NULL,
     score decimal,
```

```
    PRIMARY KEY (student_id, test_id),
    FOREIGN KEY (student_id) REFERENCES student_basic (student_id),
    FOREIGN KEY (test_id) REFERENCES test(test_id)
);
```

Column	Type	Not Null	Default	Constraints	Actions				Comment
student_id	integer	NOT NULL			Browse	Alter	Privileges	Drop	
test_id	integer	NOT NULL			Browse	Alter	Privileges	Drop	
score	numeric				Browse	Alter	Privileges	Drop	

```
INSERT INTO score_rel (student_id, test_id, score)
VALUES
(1, 2, 500),
(1, 3, 1000),
(2, 3, 1200),
(2, 4, 7),
(3, 5, 100),
(4, 1, 300),
(5, 3, 1400)
;
```

Actions		student_id	test_id	score
Edit	Delete	1	2	500
Edit	Delete	1	3	1000
Edit	Delete	2	3	1200
Edit	Delete	2	4	7
Edit	Delete	3	5	100
Edit	Delete	4	1	300
Edit	Delete	5	3	1400

7 row(s)