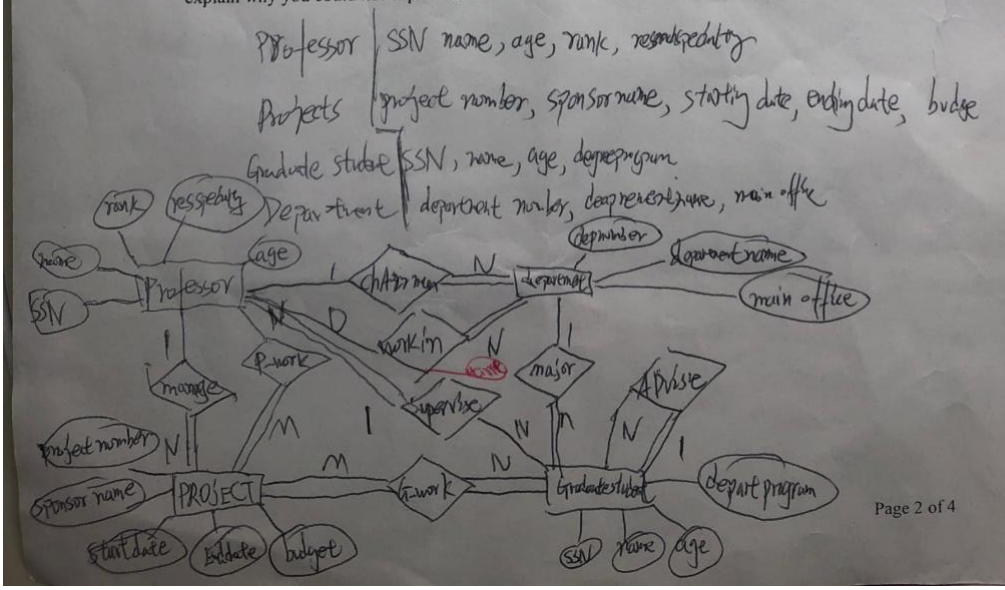
3. (23%) Following are the business descriptions written by domain experts who try to develop a database system for a university.

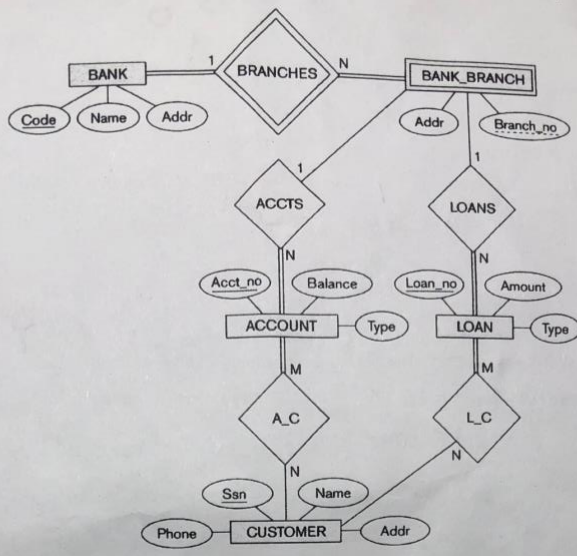
- Professors have an SSN, a name, an age, a rank, and a research specialty.
- Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget.
- Graduate students have an SSN, a name, an age, a degree program (e.g., M.S. or Ph.D.), and one major department in which they are working on their degree.
- Departments have a department number, a department name, and a main office.
- Departments have a professor (known as the chairman) who runs the department.
- Each project is managed by one professor (known as the project's principal investigator).
- Each project is worked on by one or more professors (known as the project's co-investigators).
- Professors can manage and/or work on multiple projects.
- Each project is worked on by one or more graduate students (known as the project's research assistants).
- When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
- Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.

Design a conceptual schema for the university and draw an ER diagram for your schema. Use only the basic ER model here; that is, entities, relationships, and attributes. Be sure to indicate any keys and constraints.

If you need additional assumptions to complete the diagram design, please clearly state your own assumptions here. Identify any constraints you are unable to capture in the ER diagram and briefly explain why you could not express them.



4. (25%) Map the following student lab ER diagram into a relational schema. If you need additional assumptions to complete the mapping design, please clearly state your own assumptions here.



BANK

Code	Name	Addr

Book_BRANCH

Bankcode	Branch_no	Addr

ACCOUNT

Acct_no	Bankcode	Branch_no	Balance	Type

LOAN

Loan_no	Bankcode	branch_no	Amount	Type

CUSTOMER

ssn	name	phone	Addr

A_C

Ssn	Acct_no

L_C

Ssn	Loanno

5. (30%) Consider the following schema:

```
students (sid, name, address, age)
courses (cid, name, profname)
registered (sid, cid, grade)
```

The key fields are underlined. The `registered` relation lists the grades in courses by students.

Formulate the following queries. **If the query cannot be formulated, briefly explain why.**

a. (6%) Write the query "*Find the* `sid` *of students who received grades higher than 90 in a course named* `'Computer Network'`" in **domain relational calculus**.

$\{a \mid (\exists b)(\exists c)(\exists d)(\text{students}(abcd)$ and
$(\exists X)(\exists Y)(\exists Z)(\text{registered}(XYZ)$ and $Z > 90$ and $a = X)$
$((\exists q)(\exists r)(\exists t)(\text{Courses}(qrt)$ and $r = \text{"Computer Network"}$ and $q = Y)\}$

b. (6%) Write the query "*Find the* `name` *of students who register in* `'Computer graphic'` *course* ***OR*** *whose name is* `'F. Wong'`." in **tuple relational calculus.**

$\{t.name \mid student(t)$ and $(t.name = \text{'F. wong'}$ or $((\forall X)(\text{courses}(X)$
and $X.name = \text{'computer graphic'}$ and $(\exists Y)(\text{registered}(Y)$
and $Y.cid = X.cid$ and $Y.sid = t.sid))))\}$

c. (6%) Write the query "*Find the* `sid` *of students who have taken all the courses given by* `'S. Navathe'`" in **relational algebra.**

$TEMP = \pi_{cid}(\sigma_{profname = \text{'S. Navathe'}}(\text{courses}))$

~~$TEMP = \pi_{sid, cid}(Temp_1 \div \pi_{cid}(courses))$~~

$Result = Temp * registered \div TEMP$

-3

d. (6%) Write the query "*Find pairs of student ids such that the student with the first sid got lower grade in some classes than the student with the second sid*" in **tuple relational calculus.**

$T_1(sid_1, cid, grade) = registered$

$T_2(sid_2, cid, grade) = registered$

$Result = \pi_{sid_1, sid_2}(T_1 \bowtie_{T_1.cid = T_2.cid \text{ and } T_1.grade < T_2.grade} T_2)$

-3

e. (6%) Write the query "*For each course name, retrieve the average grade of all students taken that course*" in **relational algebra**. You may use grouping and/or aggregate functions for the query, if necessary.

$Average = {}_{name}\mathcal{F}_{average(grade)}(registered * courses)$

$Result = \pi_{name, avggrade}(students * Average)$

-3+1

CS3010301 Database Systems

Midterm

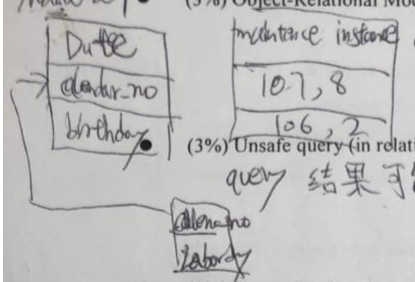
Spring 2019

SID# & Name: A1011005 任意

1. (12%) Explain the following terms briefly. Use an example if necessary.

- (3%) Object-Relational Models (in database systems)..

approach is combining that of relational database. database manage system using model template classes and inheritance directly support database schema

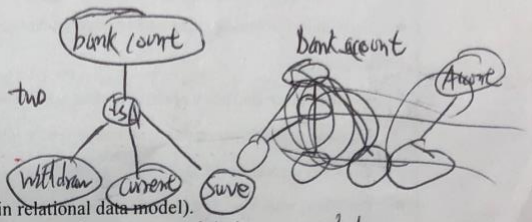


- (3%) Unsafe query (in relational calculus).

query 结果可能是无限集合 $\{e \mid \text{not employee}\}$

- (3%) Generalization (in EER model).

bottom-up approach in which two lower level entities combine to form higher level entity.



- (3%) Referential integrity constraint (in relational data model).

specialize between two relations, maintains consistency among tuples in two relation

2. (10%) Let `R (a, b, c)` be a relation schema. Assume that the domains of the `a`, `b`, and `c` attributes are all integers. Let `r1` and `r2` be two instances of `R`. For each pair of expressions below, justify if the result is **always** the same? You can use an example if necessary.

- (5%) $\mathtt{r1} * \mathtt{r2} \equiv \mathtt{r2} * \mathtt{r1}$

Yes

R	A	B	C
	1	3	
	1	4	

$F_{count\ r_1 * r_2} = 2$

$F_{count\ r_2 * r_1} = 2$

- (5%) $\mathcal{F}_{\text{count}} \Pi_{a,c}(\mathtt{r1} - \mathtt{r2}) \equiv \mathcal{F}_{\text{count}}(\Pi_{c,a}(\mathtt{r1}) - \Pi_{c,a}(\mathtt{r2}))$

No

R	A	B	C
	1	2	3
	1	5	6

S	a	b	c
	1	2	3
	1	7	8

$\Pi_{ac}(r \cap s)$ $\Pi_{ca}(r_1) \cap \Pi_{ca}(r_2)$

X	Z
1	3

X	Z
1	3
1	6