E44065020_何子安_HW4

> This report is written in Notion, and being export as PDF.

HW4, Chapter 6 & 7.

6.16. Specify the following queries on the COMPANY relational database schema shown in Figure
3.5, using the relational operators discussed in this chapter. Also show the result of each query as
it would apply to the database state of Figure 3.6.

(a) Retrieve the names of employees in department 5 who work more than 10 hours per week
on the 'ProductX' project.

```
EMP_W_X <-- ( SELECT PNAME='ProductX' (PROJECT)) EQUIJOIN (PNUMBER),(PNO) (WORKS_ON)
EMP_WORK_10 <-- (EMPLOYEE) EQUIJOIN (SSN),(ESSN) ( SELECT HOURS>10 (EMP_W_X))
RESULT <-- PROJECT LNAME,FNAME ( SELECT DNO=5 (EMP_WORK_10))
```

(b) List the names of employees who have a dependent with the same first name as themselves.

```
E <-- (EMPLOYEE) EQUIJOIN (SSN,FNAME),(ESSN,DEPENDENT_NAME) (DEPENDENT)
R <-- PORJECT LNAME,FNAME (E)
```

(c) Find the names of employees that are directly supervised by 'Franklin Wong'.

```
WONG_SSN <-- PROJECT SSN ( SELECT FNAME='Franklin' AND LNAME='Wong' (EMPLOYEE))
WONG_EMPS <-- (EMPLOYEE) EQUIJOIN (SUPERSSN),(SSN) (WONG_SSN) RESULT <-- PROJECT LNAM
E,FNAME
(WONG_EMPS)
```

(d) For each project, list the project name and the total hours per week (by all employees)
spent on that project.

```
PROJ_HOURS(PNO,TOT_HRS) <-- PNO EQUIJOIN SUM HOURS (WORKS_ON)
RESULT <-- PROJECT PNAME,TOT_HRS ( (PROJ_HOURS) EQUIJOIN (PNO),(PNUMBER) (PROJECT) )
```

(e) Retrieve the names of employees who work on every project.

```
PROJ_EMPS(PNO,SSN) <-- PROJECT PNO,ESSN (WORKS_ON) ALL_PROJS(PNO) <-- PROJECT PNUMBER
(PROJECT) EMPS_ALL_PROJS <-- PROJ_EMPS -:- ALLPROJS (NATURAL JOIN DIVISION operation N
ATURAL JOIN) RESULT <--
PROJECT LNAME,FNAME (EMPLOYEE * EMP_ALL_PROJS)
```

(f) Retrieve the names of employees who do not work on any project.

```
ALL_EMPS <-- PROJECT SSN (EMPLOYEE) WORKING_EMPS(SSN) <-- PROJECT ESSN (WORKS_ON)
NON_WORKING_EMPS <-- ALL_EMPS - WORKING_EMPS (NATURAL JOIN DIFFERENCE) RESULT <-- PROJ
ECT
LNAME,FNAME (EMPLOYEE * NON_WORKING_EMPS)
```

(g) For each department, retrieve the department name, and the average salary of employees working in that department.

```
DEPT_AVG_SALS(DNUMBER,AVG_SAL) <-- DNO EQUIJOIN AVG SALARY (EMPLOYEE) RESULT <-- PROJE
CT
DNUMBER,AVG_SAL ( DEPT_AVG_SALS NATURAL JOIN DEPARTMENT )
```

(h) Retrieve the average salary of all female employees.

```
RESULT(AVG_F_SAL) <-- FUNCTION AVG SALARY ( SELECT SEX='F' (EMPLOYEE) )
```

(i) Find the names and addresses of employees who work on at least one project located in
Houston but whose department has no location in Houston.

```
E_P_HOU(SSN) <-- PROJECT ESSN (WORKS_ON J(PNO),(PNUMBER) ( SELECT PLOCATION='Houston'
(PROJECT))) D_NO_HOU <-- PROJECT DNUMBER (DEPARTMENT) - PROJECT DNUMBER ( SELECT DLOCA
TION='Houston'
(DEPARTMENT)) E_D_NO_HOU <-- PROJECT SSN (EMPLOYEE J(PNO),(DNUMBER) (D_NO_HOU))
RESULT_EMPS <-- E_P_HOU - E_D_NO_HOU (NATURAL JOIN this is set DIFFERENCE NATURAL JOI
N) RESULT <-- PROJECT LNAME,FNAME,ADDRESS (EMPLOYEE * RESULT_EMPS)
```

(j) List the last names of department managers who have no dependents.

```
DEPT_MANAGERS(SSN)<-- PROJECT MGRSSN (DEPARTMENT) EMPS_WITH_DEPENDENTS(SSN) <-- PROJEC
T
ESSN (DEPENDENT) RESULT_EMPS <-- DEPT_MANAGERS - EMPS_WITH_DEPENDENTS RESULT
<-- PROJECT LNAME,FNAME (EMPLOYEE * RESULT_EMPS)
```

6.18. Consider the LIBRARY relational database schema shown in Figure 6.14, which is used to keep track of books, borrowers, and book loans. Referential integrity constraints are shown as directed arcs in Figure 6.14. Write down relational expressions for the following queries:

1. How many copies of the book titles The Lost Tribe are owned by the library branch whose name is 'Sharpstown'?

```
A <-- BOOKCOPIES NATURAL LIBRARY-BRANCH JOIN BOOK
RESULT <-- PROJECT No_Of_Copies ( SELECT BranchName='Sharpstown' and Title='The Lost T
ribe')
```

2. How many copies of the book titles The Lost Tribe are own edbyeachlibrarybranch?

```
PROJECT BranchID,No_Of_Copies ( ( SELECT Title='The Lost Tribe' (BOOK)) NATURAL JOIN B
OOKCOPIES )
```

3. Retrieve the names of all borrowers who do not have any books checked out.

```
NO_CHECKOUT_B <-- PROJECT CardNo (BORROWER) - PROJECT CardNo (BOOK_LOANS)
RESULT <-- PROJECT Name (BORROWER * NO_CHECKOUT_B)
```

4. ForeachbookthatisloanedoutfromtheSharpstownbranchandwhoseDue_dateistoday, retrieve the book title, the borrower's name, and the borrower's address.

```
S <-- P BranchId ( SELECT BranchName='Sharpstown' (LIBRARY-BRANCH) )
B_FROM_S <-- PROJECT BookId,CardNo ( ( SSELECT DueDate='today' (BOOKLOANS) ) NATURAL J
OIN S )
RESULT <-- PROJECT Title,Name,Address ( BOOK * BORROWER * B_FROM_S )
```

5. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.

```
R(BranchId,Total) <-- BranchId FCOUNT(BookId,CardNo) (BOOK_LOANS)
RESULT <-- PROJECT BranchName,Total (R * LIBRARY_BRANCH)
```

6. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.

```
B(CardNo,TotalCheckout) <-- CardNo AGGREGATE COUNT(BookId) (BOOK_LOANS)
B5 <-- SELECT TotalCheckout > 5 (B)
RESULT <-- PROJECT Name,Address,TotalCheckout ( B5 * BORROWER)
```

7. For each book authored (or coauthored) by Stephen King, retrieve the title and the number of copies owned by the library branch whose name is Central.

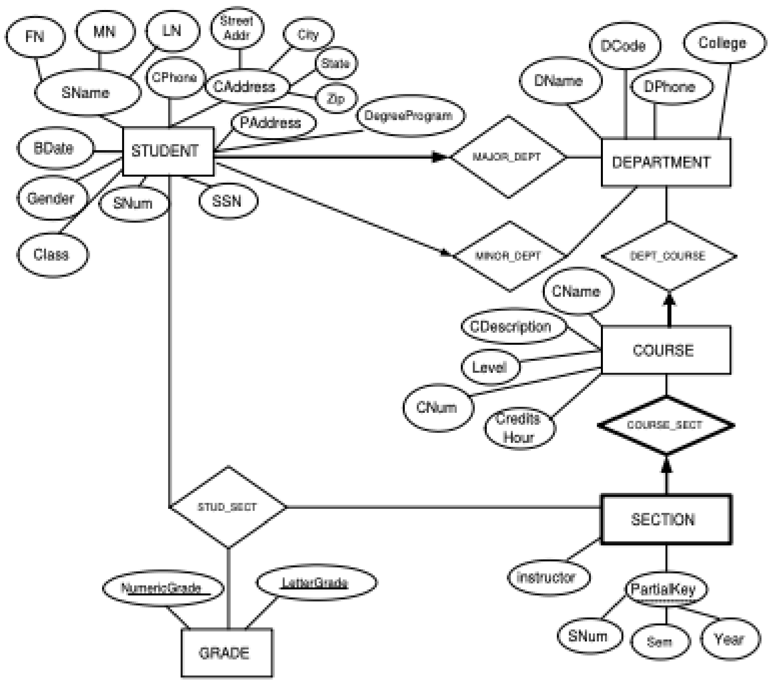
```
SK(BookId,Title) <-- ( sAuthorName='Stephen King' ( BOOK_AUTHORS)) * BOOK
CENTRAL(BranchId) <-- sBranchName='Central' ( LIBRARY_BRANCH )
RESULT <-- PROJECT Title,NoOfCopies ( SK * BOOKCOPIES * CENTRAL )
```

7.16. Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students' transcripts. This is similar but not identical to the database shown in Figure 1.2.

1. The university keeps track of each student's name, student number, Social Security number,
current address and phone number, permanent address and phone number, birth date, sex,
class (freshman, sophomore, ..., graduate), major department, minor department (if any),
and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city,
state, and ZIP Code of the student's permanent address and to the student's last name. Both
Social Security number and student number have unique values for each student.
2. Each department is described by a name, department code, office number, office

phone
number, and college. Both name and code have unique values for each department.
3. Each course has a course name, description, course number, number of semester hours,
level, and offering department. The value of the course number is unique for each course.
4. Each section has an instructor, semester, year, course, and section number. The section
number distinguishes sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each
semester.
5. A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, or 4).

Design an ER schema for this application, and draw an ER diagram for the schema. Specify key attributes of each entity type, and structural constraints on each relationship type. Note any unspecified requirements, and make appropriate assumptions to make the specification complete.



7.23. Consider the ER diagram shown in Figure 7.21 for part of a BANK database. Each bank can have multiple branches, and each branch can have multiple accounts and loans.

1. List the strong (nonweak) entity types in the ER diagram.

- Customer, Account, Loan, Bank

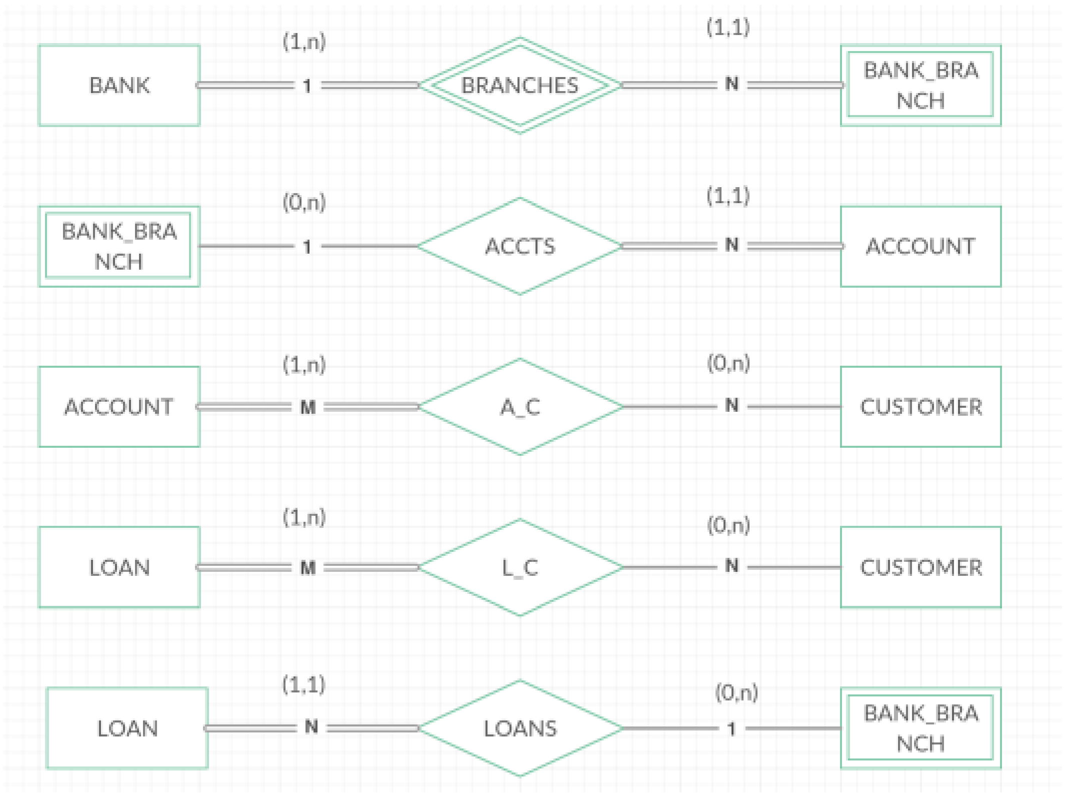
2. Is there a weak entity type? If so, give its name, partial key, and identifying relationship.

- Bank_Branch is a weak entity. Its partial key is Branch-noand its identifying relationship is BRANCHES with Bank

3. What constraints do the partial key and the identifying relationship of the weak entity typespecify in this diagram?

- The constraint of the partial key Branch-no is that we need to combine Branch-nowith Code, the key from its owner entitysetBank, to uniquely identify a Bank_Branch.The constraints of the identifying relationship are:
    - Theidentifying relationship between the owner entity set, Bank,and the weak entity set, Bank_Branchmust be one to many and Bank_Branch could only have one Bank as its owner.
    - The weak entity set, Bank_Branch,must have total participation in the identifying relationshipset, BRANCHES

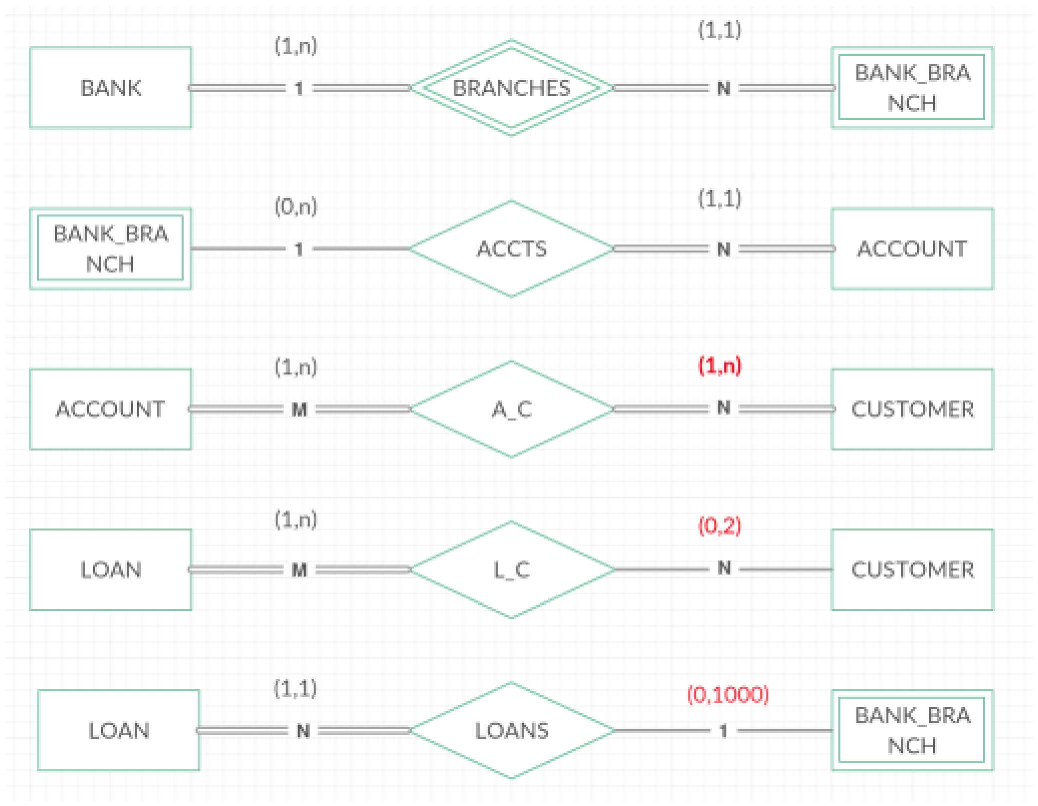
4. List the names of all relationship types, and specify the (min, max) constraint on each participation of an entity type in a relationship type. Justify your choices.



5. List concisely the user requirements that led to this ER schema design.

- The requirements may be stated as follows: Each BANK has a unique Code, as well as a Name and Address. Each BANK is related to one or more BANK-BRANCHes, and the BranhNo is unique among
  each set of BANK-BRANCHes that are related to the same BANK. Each BANK-BRANCH has an Address. Each BANK-BRANCH has zero or more LOANS and zero or more ACCTS. Each ACCOUNT
  has an AcctNo (unique), Balance, and Type and is related to exactly one BANK-BRANCH and to at
  least one CUSTOMER. Each LOAN has a LoanNo (unique), Amount, and Type and is related to exactly
  one BANK-BRANCH and to at least one CUSTOMER. Each CUSTOMER has an SSN (unique), Name,
  Phone, and Address, and is related to zero or more ACCOUNTs and to zero or more LOANs.

6. Suppose that every customer must have at least one account but is restricted to at most two loans at a time, and that a bank branch cannot have more than 1,000 loans. How does this show up on the (min, max) constraints?


BANK
(1,n)
1
BRANCHES
(1,1)
N
BANK_BRANCH
BANK_BRANCH
(0,n)
1
ACCTS
(1,1)
N
ACCOUNT
ACCOUNT
(1,n)
M
A_C
(1,n)
N
CUSTOMER
LOAN
(1,n)
M
L_C
(0,2)
N
CUSTOMER
LOAN
(1,1)
N
LOANS
(0,1000)
1
BANK_BRANCH