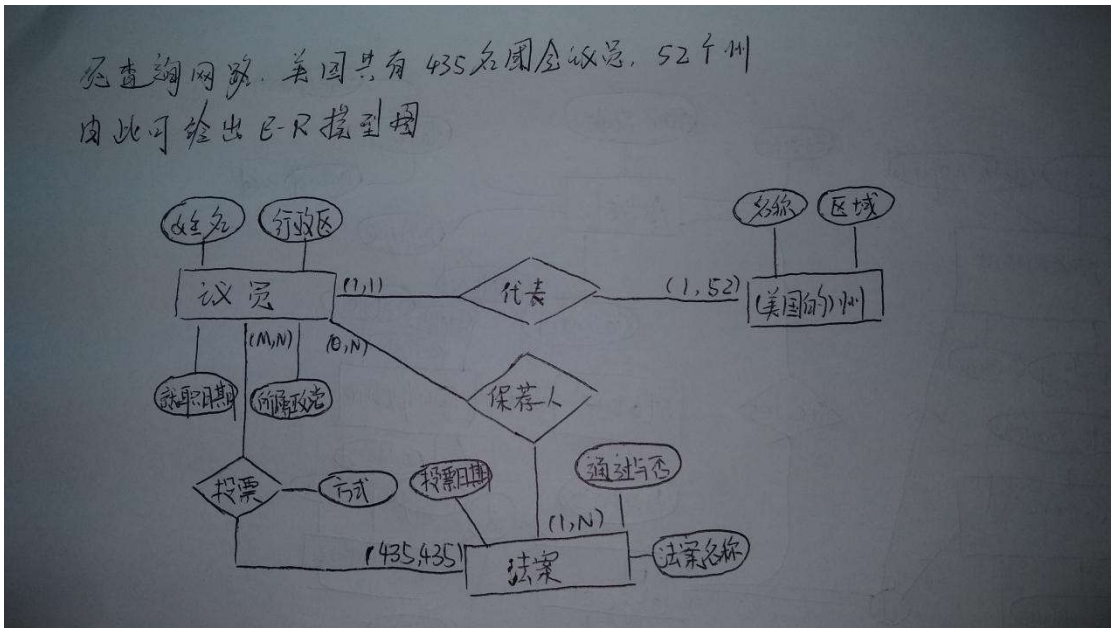
数據庫第一次作業

1.(1)在圖示 1.2 中，每一個 SECTION 記錄都和 COURSE 記錄相關聯。
(2)每一個 GRADE_REPORT 記錄都和 STUDENT 记錄相關聯。
(3)每一個 PREREQUISITE 記錄都和兩個 COURSE 記錄相關聯,分別是課程和它的先修課。

2.如圖所示



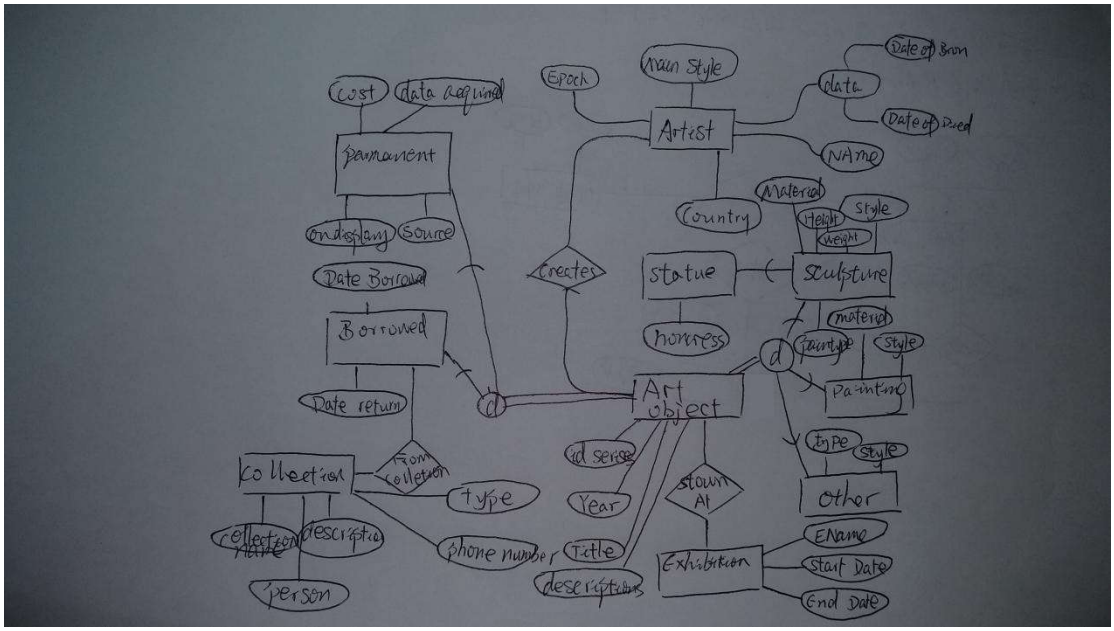
3.a. bank, account, loan, customer
b. weak entity type: bank_branch
partial key: branch_no
identifying relationship: branches
c. bank_branch 不能再缺失 associated bank 的情況下單獨存在。

d.branches- bank:bank_branch, 1:N
accts- bank_branch:account, 1:N
loans- bank_branch:loan, 1:N
a_c- acount:customer, N:M
l_c- loan:customer, N:M

e.一家银行实体，可以与许多银行分行相关，每家分行可以有多个账户和贷款给客户。 每个客户可以拥有多个贷款和账户。 每个贷款和账户都必须与客户相关联。

f. a_c-customer to accounts: (1,N)
l_c-customer to loans: (0,2)
loans-bank_branch to loan: (0,1000)

4.如圖所示



5 結論：(1)(2)正確,(3)錯誤
原因：在圖示表明的三種情況中，E1 和 E2 是有重叠的，如果實體 E2 和實體 E3 之間存在相互關聯，則必須滿足重合的實體與 E3 無關。

資料庫系統第二次作業

5.12
A.我們需要以下若干操作：
INSERT <Flight_number,Leg_number,Date Seat_number,Customer_name,Customer_phone>
INTO SEAT_RESERVATION;
我們可以舉個例子：
INSERT<'FLNO1','1','2018/04/20','2','jack','0999999'>INTO SEAT_RESERVATION;
UPDATE LEG_INSTANCE SET (Number_of_available_seats=Number_of_available_seats-1)
WHERE Flight_number=Flight_number AND Leg_number=Flight_number AND Date=Date
B. 我們需要檢查:
1.在給定日期的航班是否可用（檢查“LEG_INSTANCE”表）
Number_of_available_seats 是否大於 1，即檢查是否有座位空閑。（檢查“AIRPLANE”表）
同時檢查 SEAT_NUMBER 中的座位是否存在。
C. 我們在將數據插入 SEAT_RESERVATION 時，需要檢查每個鍵，實體完整性和參照完整性
同時需要檢查每個 LEG_INSTANCE(航班是否可用)的 NUMBER_OF_AVAILABLE_SEATS 是否大於 1（售票是否允許）
D.在 3.8 中所有的參照完整性如下所示：
FLIGHT_LEG.DEPARARTURE_AIRPORT_CODE 需要參照 AIRPORT
FLIGHT_LEG.ARRIVAL_AIRPORT_CODE 需要參照 AIRPORT
LEG_INSTANCE.(FLIGHT_NUMBER,LEG_NUMBER)需要參照 FLIGHT_LEG
LEG_INSTANCE.DEPARTURE_AIRPORT_CODE 需要參照 AIRPORT
LEG_INSTANCE.ARRIVAL_AIRPORT_CODE 需要參照 AIRPORT

LEG_INSTANCE.AIRPLANE_ID 需要參照 FLIGHT
CAN_LAND.AIRPORT_TYPE_NAME 需要參照 AIRPORT
AIRPLANE.AIRPLANE_TYPE 需要參照 AIRPLANE_TYPE
SEAT_RESERVATION.(FLIGHT_NUMBER,LEG_NUMBER.DATE)需要參照 LEG_INSTANCE

5.17
這個問題的約束遵循以下三個外碼
1，OPTION 中 Serial_no 是 CAR 的外碼。
2，SALE 中的 Salesperson_id 是 SALE 的外碼。
3，SALE 中的 Serial_no 是 OPTION 的外碼。
根據模型擴充元組

Serial_no	Model	Manufacturer	Price
1	1950	A	3
2	1969	B	9
3	1992	A	18

Serial_no	Option_name	Price
2	EEE	302
3	FFF	688

Saleperson_id	Name	Phone
R1	JACK	123
R2	LUCY	456
R3	TOM	789

Saleperson_id	Serial_no	Date	Sale_price
R1	2	1991	3.1
R2	3	1992	7.2

違背參照完整性約束
INSERT<’R4’,’7’,’1993’,’12’>INTO...
原因是 Serial_id 不存在
INSERT<’R1’,’7’,’1993’,’12’>INTO SALESPERSON
原因是 R1 已經存在
不違背參照完整性約束
INSERT<’R4’,2,1994-01-02,7.3>INTO SALESPERSON
它沒有違背參照完整性約束。

8.15
Query 1:

Fname	Lname	Address
John	Smith	731 Fondren,Houston,TX
Franklin	Wong	638 Voss,Houston,TX
Ramesh	Narayan	975 Fire Oak,Humble,TX
Joyce	English	5631 Rice,Houston,TX

Query 2:

Pnumber	Dnumber	Lname	Address	Bdate
10	4	Wallace	291 Berry,Bellaire,TX	1941-06-20

30 4 Wallace 291 Berry,Bellaire,TX 1941-06-20

Query 3:

[EMPTY]

Query 4:

Pnumber

1

1

Query 5:

Lname Fname

Smith John

Wong

Query 6:

Lname Fname

Zelaya Alicia

Narayan Ramesh

English Joyce

Jabbar Ahmad

Borg James

Query 7:

Wallace Jennifer

Wong

8.17

a.

DEPARTURE = $_{\text{Flight\_number}}\xi_{\text{MIN Leg\_number}}$(FLIGHT_LEG)

Departure_airport = $\pi_{\text{Flight\_number,Departure\_airport\_code}}$(DEPARTURE *FLIGHT_LEG)

ARRIVAL = $_{\text{Flight\_number}}\xi_{\text{MAX Leg\_number}}$(FLIGHT_LEG)

Arrival _airport = $\pi_{\text{Flight\_number, Arrival\_airport\_code}}$(ARRIVAL * FLIGHT_LEG)

RESUL<- Departure_airport * Arrival _airport

DEPARTURE = $_{\text{Flight\_number}}\xi_{\text{MIN Leg\_number}}$(FLIGHT_LEG)

ARRIVAL = $_{\text{Flight\_number}}\xi_{\text{MAX Leg\_number}}$(FLIGHT_LEG)

RESULT<- $\pi_{\text{Flight\_number,Departure\_airport\_code,Arrival\_airport\_code}}$((DEPARTURE ⨯ ARRIVAL)*AIRPORT)

b.

DEP_HOUS <- $\sigma_{\text{Departure\_airport\_code = 'IAH'}}$(FLIGHT_LEG)

ARR_LA <- $\sigma_{\text{Arrival\_airport\_code = 'LAX'}}$(FLIGHT_LEG)

HOUSTOLA <- DEP_HOUS * ARR_LA

RESULT <- $\pi_{\text{Flight\_number, Weekdays}}$(HOUSTOLA* FLIGHT)

c.

DEP_HOUS <- $\sigma_{\text{Departure\_airport\_code = 'IAH'}}$(FLIGHT_LEG)

ARR_LA <- $\sigma_{\text{Arrival\_airport\_code = 'LAX'}}$(FLIGHT_LEG)

HOUSTOLA <- DEP_HOUS * ARR_LA

RESULT <- $\pi_{\text{Flight\_number, Departure\_airport\_code, Scheduled\_departure\_time, Arrival\_airport\_code,Scheduled\_arrival\_time, Weekdays}}$(HOUSTOLA* FLIGHT)

d.

RESULT <- $\sigma_{\text{Flight\_number = 'co197'}}$(FARE)

e.

INFORM <- $\sigma_{Flight\_number='CO197' \text{ AND } Date='2009-10-09'}$(LEG_INSTANCE)

RESULT <- $\pi_{Number\_of\_available\_seats}$(INFORM)

8.22

(a)

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	10	b	5
25	a	6	25	c	3

(b)

P	Q	R	A	B	C
15	b	8	10	b	6
15	b	8	10	b	5

(c)

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	10	b	5
15	b	8	null	null	null
25	a	8	25	c	3

(d)

P	Q	R	A	B	C
15	b	8	10	b	6
null	null	null	25	c	3
15	b	8	10	b	5

(e)

P	Q	R
10	a	5
15	b	8
25	a	6
10	b	6
25	c	3
10	b	5

(f)

P	Q	R	A	B	C
10	a	5	10	b	5

資料庫作業 3

1.(10pt) Consider the database shown in Figure 1.2, write appropriate SQL DDL statements to create the database shown in Figure 1.2 in your MySQL account.

```
CREATE TABLE STUDENT
(
Name varchar(255),
```

```
Student_number int,
Class int,
Major varchar(255)，
);
CREATE TABLE COURSE
(
Course_name varchar(255),
Course_number varchar(255),
Credit_hours int,
Department varchar(255)
);
CREATE TABLE SECTION
(
Section_identifier int,
Course_number varchar(255),
Semester varchar(255),
Year varchar(255),
Instructor varchar(255)
);
CREATE TABLE GRADE_REPORT
(
Student_number int,
Section_identifier int,
Grade varchar(255)
);
CREATE TABLE PREREQUISITE
(
Course_number varchar(255),
Prerequisite_number varchar(255)
);
INSERT INTO STUDENT (Name, Student_number,Class,Major) VALUES ('Smith',17,1,'CS');
INSERT INTO STUDENT (Name, Student_number,Class,Major) VALUES ('Brown',8,2,'CS');
INSERT INTO COURSE (Course_name,Course_number,Credit_hours,Department) VALUES ('Intro to Computer Science','CS1310',4,'CS');
INSERT INTO COURSE (Course_name,Course_number,Credit_hours,Department) VALUES ('Data Structures','CS3320',4,'CS');
INSERT INTO COURSE (Course_name,Course_number,Credit_hours,Department) VALUES ('Discrete Mathematics','MATH2410',3,'MATH');
INSERT INTO COURSE (Course_name,Course_number,Credit_hours,Department) VALUES ('Database','CS3380',3,'CS');
INSERT INTO SECTION (Section_identifier,Course_number,Semester,Year,Instructor) VALUES (85,'MATH2410','Fall','07','King');
INSERT INTO SECTION (Section_identifier,Course_number,Semester,Year,Instructor) VALUES (92,'CS1310','Fall','07','Anderson');
```

```
INSERT INTO SECTION (Section_identifier,Course_number,Semester,Year,Instructor) VALUES (102,'CS3320','Spring','08','Knuth');
INSERT INTO SECTION (Section_identifier,Course_number,Semester,Year,Instructor) VALUES (112,'MATH2410','Fall','08','Chang');
INSERT INTO SECTION (Section_identifier,Course_number,Semester,Year,Instructor) VALUES (119,'CS1310','Fall','08','Anderson');
INSERT INTO SECTION (Section_identifier,Course_number,Semester,Year,Instructor) VALUES (135,'CS3380','Fall','08','Stone');
INSERT INTO GRADE_REPORT (Student_number,Section_identifier,Grade) VALUES (17,112,'B');
INSERT INTO GRADE_REPORT (Student_number,Section_identifier,Grade) VALUES (17,119,'C');
INSERT INTO GRADE_REPORT (Student_number,Section_identifier,Grade) VALUES (8,85,'A');
INSERT INTO GRADE_REPORT (Student_number,Section_identifier,Grade) VALUES (8,92,'A');
INSERT INTO GRADE_REPORT (Student_number,Section_identifier,Grade) VALUES (8,102,'B');
INSERT INTO GRADE_REPORT (Student_number,Section_identifier,Grade) VALUES (8,135,'A');
INSERT INTO PREREQUISITE (Course_number,Prerequisite_number) VALUES ('CS3380','CS3320');
INSERT INTO PREREQUISITE (Course_number,Prerequisite_number) VALUES ('CS3380','MATH2410');
INSERT INTO PREREQUISITE (Course_number,Prerequisite_number) VALUES ('CS3320','CS1310');
```

2.(15pt) Consider the database shown in Figure 5.6, whose schema is shown in Figure 5.7. Write appropriate SQL DDL statements to create the database shown in Figure 5.6 in your MySQL account.

```
CREATE TABLE EMPLOYEE
( Fname   VARCHAR(15)  NOT NULL,
Minit       CHAR,
Lname     VARCHAR(15)  NOT NULL,
Ssn   CHAR(9)   NOT NULL,
Bdate       DATE,
Address   VARCHAR(30),
Sex   CHAR,
Salary      DECIMAL(10,2),
Super_ssn       CHAR(9),
Dno INT   NOT NULL,
PRIMARY KEY (Ssn));
CREATE TABLE DEPARTMENT
( Dname   VARCHAR(15)  NOT NULL,
Dnumber INT   NOT NULL,
Mgr_ssn   CHAR(9)   NOT NULL,
Mgr_start_date      DATE,
PRIMARY KEY (Dnumber),
UNIQUE (Dname),
FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );
CREATE TABLE DEPT_LOCATIONS
( Dnumber       INT   NOT NULL,
```

```
Dlocation VARCHAR(15)  NOT NULL,
PRIMARY KEY (Dnumber, Dlocation),
FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );
CREATE TABLE PROJECT
(Pname    VARCHAR(15)  NOT NULL,
Pnumber INT   NOT NULL,
Plocation VARCHAR(15),
Dnum      INT   NOT NULL,
PRIMARY KEY (Pnumber),
UNIQUE (Pname),
FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );
CREATE TABLE WORKS_ON
( Essn      CHAR(9)   NOT NULL,
Pno  INT   NOT NULL,
Hours      DECIMAL(3,1)  NOT NULL,
PRIMARY KEY (Essn, Pno),
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );
CREATE TABLE DEPENDENT
( Essn      CHAR(9)   NOT NULL,
Dependent_name   VARCHAR(15)  NOT NULL,
Sex   CHAR,
Bdate      DATE,
Relationship    VARCHAR(8),
PRIMARY KEY (Essn, Dependent_name),
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );
INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`) VALUES ('John', 'B', 'Smith', '123456789', '1965-01-09', '731 Fondren,Houston, TX', 'M', '30000', '333445555', '5');
INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`) VALUES ('Franklin','T','Wong','333445555','1955-12-08','638 Voss,Houston,TX','M','40000','888665555','5');
INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`) VALUES ('Alicia','J','Zetaya','999887777','1968-01-19','3321 Castle,Spring,TX','F','25000','987654321','4');
INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`) VALUES ('Jennifer','S','Wallace','987654321','1941-06-20','291 Bony,Bellaire,TX','F','43000','888665555','4');
INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`) VALUES ('Ramesh','K','Narayan','666884444','1982-09-15','975 Fie,Oalc,Humble,TX','M','38000','333445555','5');
INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`) VALUES ('Joyce','A','English','453453453','1972-07-31','5831 Rice,Houston,TX','F','25000','333445555','5');
```

INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`) VALUES ('Ahmed','V','Jabber','987987987','1969-03-29','980 Dallas,Houston,TX','M','25000','987654321','4');
INSERT INTO `employee` (`Fname`, `Minit`, `Lname`, `Ssn`, `Bdate`, `Address`, `Sex`, `Salary`, `Super_ssn`, `Dno`) VALUES ('James','E','Borg','888665555','1937-11-10','450 Stone,Houston,TX','M','55000',NULL,'1');
INSERT INTO `department` (`Dname`, `Dnumber`, `Mgr_ssn`, `Mgr_start_date`) VALUES ('Research', '5', '333445555', '1988-05-22'), ('Administration', '4', '987654321', '1995-01-01'),('Headquarters', '1', '888665555', '1981-06-19');
INSERT INTO `dept_locations` (`Dnumber`, `Dlocation`) VALUES ('1', 'Houston'), ('4', 'Stafford'),('5', 'Bellaire'), ('5', 'Sugarland') ,('5', 'Houston');
INSERT INTO `works_on` (`Essn`, `Pno`, `Hours`) VALUES ((123456789,1,32.5),(123458789,2,7.5),(666884444,3,40.0),(453453453,1,20.0),(453453453,2,20.0),(333445555,2,10.0),(333445555,3,10.0),(333445555,10,10.0),(333445555,20,10.0),(999887777,30,30.0),(999887777,10,10.0),(987987987,10,35.0),(987987987,30,5.0),(987654321,30,20.0),(987654321,20,15.0),(888665555,20,NULL));
INSERT INTO `project` (`Pname`, `Pnumber`, `Plocation`, `Dnum`) VALUES ('ProductX', '1', 'Bellaire', '5'),('ProducY', '2', 'Sugarland', '5'),('ProductZ', '3', 'Houston', '5'),('Computerization', '10', 'Stafford', '1'),('Reorganization', '20', 'Stafford', '1'),('Newbenefits', '30', 'Bellaire', '4');
INSERT INTO `dependent` (`Essn`, `Dependent_name`, `Sex`, `Bdate`, `Relationship`) VALUES ('333445555', 'Alice', 'F', '1986-04-05', 'Daughter'),('333445555', 'Theodore', 'M', '1983-10-25', 'Son'),('333445555', 'Joy', 'F', '1958-05-03', 'Spouse'),('987654321', 'Abner', 'M', '1942-02-28', 'Spouse'),('123456789', 'Michael', 'M', '1988-01-04', 'Son'),('123456789', 'Alice', 'F', '1988-12-30', 'Daughter'),('123456789', 'Elizabeth', 'F', '1967-05-05', 'Spouse');

3.(15pt) Exercise 6.10.

A） Select emp.Fname,emp.Lname from employee emp,works_on w,project p where emp.Dno=5 and emp.ssn=w.Essn and w.Pno=p.pnumber and p.pname='ProductX' and w.hours>10;

Fname Lname;

John Smith

Joyce English

B） Select emp.Fname, emp.Lname from employee emp, dependent d where emp.ssn= d.essn and emp.Fname = d.Dependent_name;

(NULL)

C) Select emp.Fname, emp.Lname from employee emp, employee emp1 where emp1.Fname= 'Franklin' and emp1.Lname = 'Wong' and emp.superssn = emp1.ssn;

Fname Lname

John Smith

Ramesh Narayan

Joyce English

4.(15pt) Exercise 6.11 (specify the updates of Exercise 5.11 using the SQL update commands).

(A) Insert < 'Robert', 'F', 'Scott', '943775543', '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1 > into EMPLOYEE;

INSERT INTO EMPLOYEE VALUES ('Robert', 'F', 'Scott', '943775543', '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX',M, 58000, '888665555', 1);

(B) Insert < 'ProductA', 4, 'Bellaire', 2 > into PROJECT;

INSERT INTO PROJECT VALUES ('ProductA', 4, 'Bellaire', 2);

(C) Insert < 'Production', 4, '943775543', '01-OCT-88' > into DEPARTMENT;

INSERT INTO DEPARTMENT VALUES ('Production', 4, '943775543', '01-OCT-88');

(D) Insert < '677678989', null, '40.0' > into WORKS_ON;

INSERT INTO WORKS_ON VALUES ('677678989', NULL, '40.0');

(E) Insert < '453453453', 'John', M, '12-DEC-60', 'SPOUSE' > into DEPENDENT;

INSERT INTO DEPENDENT VALUES ('453453453', 'John', M, '12-DEC-60', 'SPOUSE');

(F) Delete the WORKS_ON tuples with ESSN= '333445555';

DELETE FROM WORKS_ON WHERE ESSN= '333445555';

(G) Delete the EMPLOYEE tuple with SSN= '987654321';

DELETE FROM EMPLOYEE WHERE SSN= '987654321';

(H) Delete the PROJECT tuple with PNAME= 'ProductX';

DELETE FROM PROJECT WHERE PNAME= 'ProductX';

(I) Modify the MGRSSN and MGRSTARTDATE of the DEPARTMENT tuple with DNUMBER= 5 to '123456789' and '01-OCT-88';UPDATE DEPARTMENT SET MGRSSN = '123456789', MGRSTARTDATE = '01-OCT-88' WHERE DNUMBER= 5;

(J) Modify the SUPERSSN attribute of the EMPLOYEE tuple with SSN= '999887777' to '943775543';UPDATE EMPLOYEE SET SUPERSSN = '943775543' WHERE SSN= '999887777';

(K) Modify the HOURS attribute of the WORKS_ON tuple with ESSN= '999887777' and PNO= 10 to '5.0';UPDATE WORKS_ON SET HOURS = '5.0' WHERE ESSN= '999887777' AND PNO= 10;

5.(15pt) Exercise 6.12.

A)Select Name from Student where Major = 'CS';

Name

Smith

Brown

B)Select Course_name from Course, Section where Course.Course_number = Section.Course_number and instructor = "king" AND (Year='07' or Year='08');

Course_name

Discrete Mathematics

C)Select Course_number, Semester, Year, Count(g.Student_number) As "# Students" from Sections, Grade_Report g where s.instructor= "king" and s.Section_identifier = g.Section_identifier Group by Course_number, Semester, Year;

MATH2410 Fail 07 8

D)Select St.Name, C.Course_name, C.Course_number, C.Credit_hours, S.Semester, S.Year, G.Grade from Student St, Course C, Section S, Grade_Report G where class = 4 AND Major='CS' AND St.Student_number= G.Student_number AND G.Section identifier= S.Section identifier AND S.Course number= C.Course_number;

(EMPTY)

6.(15pt) Exercise 6.13.

A)INSERT INTO STUDENT VALUES ('Johnson', 25, 1, 'MATH');
B)UPDATE STUDENT SET CLASS = 2 WHERE Name='Smith';
C)INSERT INTO COURSE VALUES ('Knowledge Engineering','COSC4390', 3,'COSC') ;
D)DELETE FROM STUDENT WHERE Name='Smith' AND StudentNumber=17;

7.(15pt) Exercise 7.5.

SELECT DNAME, COUNT (*) FROM DEPARTMENT, EMPLOYEE WHERE DNUMBER=DNO GROUP BY DNAME HAVING AVG (SALARY) > 30000;

DNAME	DNUMBER	COUNT(*)
Research	5	4
Administrator	4	3
Headquarters	1	1

SELECT DNAME, COUNT (*) FROM DEPARTMENT, EMPLOYEE WHERE DNUMBER=DNO AND SEX='M' AND DNO IN (SELECT DNO FROM EMPLOYEE GROUP BY DNO HAVING AVG (SALARY) > 30000) GROUP BY DNAME;

DNAME	DNUMBER	COUNT(*)
Research	5	3
Administrator	4	1
Headquarters	1	1

資料庫作業 4

14.24

Consider the universal relation $R = \{A, B, C, D, E, F, G, H, I, J\}$ and the set of functional dependencies $F = \{\{A, B\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B\} \rightarrow \{F\}, \{F\} \rightarrow \{G, H\}, \{D\} \rightarrow \{I, J\}\}$. What is the key for R? Decompose R into 2NF and then 3NF relations.

(1)如果某一個集合可以固定出一個唯一的結果且這個集合是最小的，我們就稱這個集合為這個關係的鍵。在關係 R 和依賴 F 中，由 A 可以約束 D&E，我們可以定義一個 R2={A,D,E}, 又知道 D 可以約束 I&J，所以 R2={A,D,E,I,J}同時 B 可以約束 F，則 R3={B,F},F 可以約束 G&H, 則 R3={B,F,G,H},所以聯合 R2 和 R3 我們可以得到{A,B}必然是{D,E,F,G,H,I,J}的鍵，又知道 {A,B}->{C},所以確定了 A 和 B 就確定了唯一行，**R 的鍵（key）是{A,B}**。

(2)**由上面的稱述可知 R2={A,D,E,I,J}，R3={B,F,G,H}，剩餘元素組成 R4={A,B,C}。**
由於{A}->{D}->{I,J},所以 IJ 對 AB 形成的是不完全依賴，所以我們拆分 R2 得到：
R21={D,I,J},R22={A,D,E},同理，由於{B}->{F}->{G,H},拆分 R3 為：
R31={F,G,H},r32={B,F},

所以 **3NF 是{R21,R22,R31,R32,R4}**

14.27

Consider a relation $R(A, B, C, D, E)$ with the following dependencies:

$AB \rightarrow C, CD \rightarrow E, DE \rightarrow B$. Is AB a candidate key of this relation? If not, is ABD?

Explain your answer.

由題意，根據 AB->C 可形成集合{A,B,C},不足以約束全部元素。AB 不是 R 的候選鍵。
原集合{A,B,D}經過 AB->C 的擴展后可得到集合{A,B,C,D},再經過 CD->E 的擴展，可得到集合{A,B,C,D,E}，所以**{A,B,D}是 R 的鍵**。

14.30

Consider the following relation:

CAR_SALE(Car#, Date_sold, Salesperson#, Commission%, Discount_amt)

Assume that a car may be sold by multiple salespeople, and hence {Car#, Salesperson#} is the primary key. Additional dependencies are

Date_sold → Discount_amt and

Salesperson# → Commission%

Based on the given primary key, is this relation in 1NF, 2NF, or 3NF? Why or why not? How would you successively normalize it completely?

關係 CAR_SALE 由於元素不可分必然是滿足 1NF.它的 key 是{Car#,Salesperson#}，但是由於 Salesperson# → Commission%，所以前者對後者產生不完全依賴，不滿足 2NF，我們

修改原表滿足 2NF：

CAR_SALE1(Car#, Date_sold, Discount_amt)

CAR_SALE2(Car#, Salesperson#)

CAR_SALE3(Salesperson#, Commission%)

修改原表滿足 3NF：(消除傳遞依賴)

CAR_SALE1-1(Car#,Date_sold)

CAR_SALE1-2(Date_sold, Discount_amt)

CAR_SALE2(Car#,Salesperson#)

CAR_SALE3(Salesperson#,Commission%)

Suppose you are given a relation R with four attributes ABCD. For each of the following sets of FDs, assuming those are the only dependencies that hold for R, do the following: (a) Identify the candidate key(s) for R. (b) Identify the best normal form that R satisfies (1NF, 2NF, 3NF, or BCNF). (c) If R is not in BCNF, decompose it into a set of BCNF relations that preserve the dependencies.

(a)C → D, C → A, B → C

(b)AB → C, AB → D, C → A, D → B

1.

(a) 候選碼: B

(b) 關係 R **屬於 2NF 但不屬於 3NF**
(c) C → D 和 C → A 使得 R 違反了 BCNF 準則.可以**將 R 分解為 AC, BC 和 CD.**
2.
(a) 候選碼： **AB, BC, CD, AD**
(b) 關係 **R 是 3NF 但不是 BCNF**
(c) C → A 和 D → B 使得 R 違反了 BCNF 準則. 我們可以將其分解為: AC, BCD
但因爲 D → B，BCD 仍然違反了 BCNF 規則。假設我們進一步分解: AC, BD, CD.
仍然不滿足 BCNF，所以 **b 無法分解為 BCNF**。

22.27

What implications would a no-steal/force buffer management policy have on check pointing and recovery?

steal/force 表示在用戶正式提交之前，更改就已經被寫入磁碟 no-steal/force 緩存管理策略表示**緩存頁被更新后，不會立刻寫出磁碟，除非用戶手動提交**。但是有些例如更新（updata）的操作仍然會立刻寫入磁碟，此時，如果我們需要撤回該操作，依然需要 undo 操作。

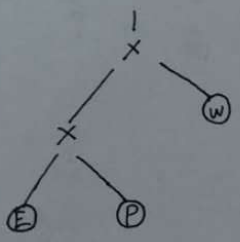
Consider the following SQL query:
SELECT P.Pname, E.Fname, E.Lname
FROM EMPLOYEE E, PROJECT P, WORK_ON W
WHERE W.Hours < 40 AND E.Ssn = W.Essn AND W.Pno = P.Pnumber AND E.Sex = 'F';
(a)Draw the query graph.
(b)Draw the initial query tree.
(c)Draw the optimized query tree using heuristic optimization.

(b)

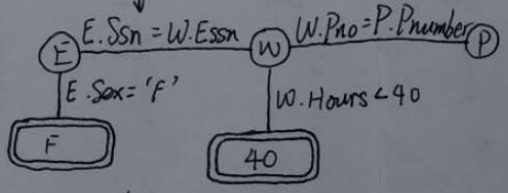
π P.Pname, E.Fname, E.Lname

σ W.Hours < 40 AND E.Ssn = W.Essn AND W.Pno = P.Pnumber AND E.Sex = 'F'

← Initial query tree



(a) Query graph

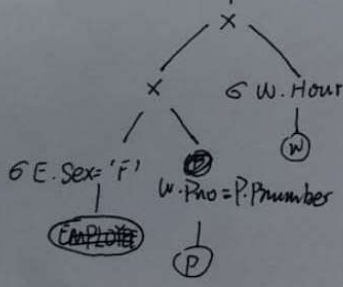


(c)

① Moving SELECT operations down the query tree.

π P.Pname, E.Fname, E.Lname

σ E.Ssn = W.Essn ~~AND σ W.Pno = P.Pnumber~~



② ~~Apply~~ the more restrictive SELECT operations first

[NOTHING CHANGE].

③ Replacing CARTESIAN PRODUCT and SELECT with JOIN operations.