5.12

(a.) 建立一筆資料(EX: <00999,0112,KIX,201709091400,TPE,201709091630>, 並將之丟入 LEG_INSTANCE 當中

(b.) 放入前必須檢查新資料的 flight_number 和 leg_number 不能與已存在的資料相同。

Flight_number 屬性在 FLIGHT 中必須能找到航班資料, leg_number 在 LEG_INSTANCE 中必須找到航程的資訊。

(c.) Entity integrity constraint: 其中 Departure_airport_code, Arrival_airport_code, Scheduled_departure_time 和 Scheduled_arrival_time 不可為 NULL

Referential integrity constraint: Flight_number 屬性在 FLIGHT 中必須能找到航班資料, leg_number 在 LEG_INSTANCE 中必須找到航程的資訊。

(d.) FLIGHT.flight_number←FLIGHT_LEG, LEG_INSTANCE, FARE, SEAT_RESERVATION

FLIGHT_LEG.Leg_number←LEG_INSTANCE, SEAT_RESERVATION

AIRPORT.airport_code←CAN_LAND

LEG_INSTANCE.Date←SEAT_RESERVATION

AIRPLANE.Airplane_id←LEG_INSTANCE

AIRPLANE.Airplane_type_name←CAN_LAND

5.17

(a.) Apartment#在 OPTION 中是 primary key 也是 foreign key

Agent_id 在 BOOKING 中是 primary key 也是 foreign key

(b.)

APARTMENT

Apartment#	Model	Address	Price_perSquareFt
0009	3LDK	Daan Dist.	500k
0010	2LDK	Daan Dist	700k

OPTION

Apartment#	Option_name	Extra_price
0009	Family	0
0009	family	0

BOOKING

Agent_id	Apartment#	Date	Booking_price
19666	0010	20170816	750K
19666	0009	20180502	490K

AGENT

Agent_id	Name	Phone
25333	WangDaMing	0911111111
19666	LeeXiaoHua	0922222222

(c.) 若在 BOOKING 插入一筆 <25652,0010,20100516,700K>的資料，此時找不到 Agent_id 所 refer 的相應資料。插入<25333,0010,20100516,750K>則無問題。
若在 AGENT 中插入資料不會有影響，但刪除資料則會影響所有 refer 到 Agent_id 的資料，違反 referential integrity constriants

8.15

QUERY1：Retrieve the name and address of all employees who work for the 'Research' department

FNAME	LNAME	ADDRESS
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Ramesh	Narayan	975 Fire Oak, Houston, TX
Joyce	English	5631 Rice, Houston, TX

QUERY2：For every project located in 'Stafford', list the project number, the controlling department number, and department manager's last name, address, and birth date

PNUMBER	DNUM	LNAME	ADDRESS	BDATE
10	4	Wallace	291 Berry, Bellaire, TX	1941-06-20
30	4	Wallace	291 Berry, Bellaire, TX	1941-06-20

QUERY3：Find the names of employees who work on all the projects controlled by department number 5.
No a employee who work on all the projects controlled by department number 5.
QUERY4：Make a list of project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.

PNUMBER
1
2

QUERY5：List the names of all employees with two or more dependents

FNAME	LNAME
John	Smith
Franklin	Wong

QUERY6：Retrieve the names of employees who have no dependents.

FNAME	LNAME
Alicia	Zelaya
Ramesh	Narayan
Joyce	English
Ahmad	Jabbar
James	Borg

QUERY7：List the names of managers who have at least one depdent

FNAME	LNAME
Franklin	Wong
Jennifer	Wallace

8.17

a. temp_flight = Π Flight_number, Airline (FLIGHT)

temp_leg = Π Flight_number, Date,Departure_airport_code,Arrival_airport_code

(σ Number_of_available_seats≠0 (LEG_INSTANCE))

result = temp_flight ⋈<temp_flight.Flight_number = temp_leg.Flight_number>temp_leg

b. ~~AIRPLANE_TYPE ∪ CAN_LAND~~

RESULT <-- (AIRPLANE_TYPE * CAN_LAND)

c. temp_seat = Π Flight_number, Seat_number, Customer_name,

Customer_phone(SEAT_RESERVATION)

temp_flight = Π Flight_number (σ (Departure_airport_code = 'iah') AND

(Arrival_airport_code = 'lax') AND

(Date = '2016-03-16') (LEG_INSTANCE))

result = temp_flight ⋈<temp_flight.Flight_number = temp_seat.Flight_number>temp_seat

d. ~~temp = Π Flight_number(σ Airline = 'Delta Airlines' (FLIGHT))~~

~~result = σ (temp = Π Flight_number(FARE))~~

TEMP1 <--ΠFlight_number (σAirline = 'Delta Airline' (FLIGHT))

RESULT <-- (TEMP1 * FARE)

e. temp_flight = Π Flight_number(σ Airline = 'Delta Airline' (FLIGHT))

temp_date = Π Flight_number(σ Date = '2016-04-09' (LEG_INSTANCE))

temp = temp_flight ∩ temp_date

result = Π Number_of_available_seats (σ temp = Flight_number(LEG_INSTANCE))

8.22

(a.)

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

(b.)

P	Q(B)	R	A	C
15	b	8	10	6
15	b	8	10	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	6	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

8.25

Tuple Relational Calculus :

a. {p.Flight_number, p.Airline, f.Flight_number, f.Date, f.Departure_airport_code, f.Arrival_airport_code | FLIGHT(p) **AND** LEG_INSTANCE(f) **AND** f.Number_of_available_seats≠0 **AND** p.Flight_number = f.Flight_number}

b. {p.Airplane_type_name, p.Max_seats, p.Company, f.Airport_code | AIRPLANE_TYPE(p) **AND** CAN_LAND(f) **AND** p.Airplane_type_name = f.Airplane_type_name}

c. {p.Seat_number, p.Customer_name, p.Customer_phone | SEAT_RESERVATION(p) **AND** LEG_INSTANCE(f) **AND** f.Departure_airport_code = 'IAH' **AND** f.Arrival_airport_code = 'LAX' **AND** f.Date = '2016-03-16' **AND** p.Flight_number = f.Flight_number}

d. {f.Flight_number, f.Fare_code, f.Amount, f.Restrictions|FARE(f) **AND** FLIGHT(p) **AND** p.Airline = 'Delta Airlines' **AND** p.Flight_number = f.Flight_number}

Domain Relational Calculus :

a. ~~{abfik | (∃g)(FLIGHT(abc) **AND** LEG_INSTANCE(defghijkl) **AND** g≠0)}~~
{abfik | (∃c) (∃d) (∃e) (∃g) (∃h) (∃j) (FLIGHT(abc) **AND** LEG_INSTANCE(defghijkl) **AND** g≠0)}

b. ~~{abce| (∃a)(∃d) (AIRPLANE_TYPE(abc) **AND** CAN_LAND(de) **AND** a = d)}~~
{abce| (∃d) (AIRPLANE_TYPE(abc) **AND** CAN_LAND(de) **AND** a = d)}

c. ~~{def | (∃i)(∃l)(∃n) (SEAT_RESERVATION(abcdef) **AND** LEG_INSTANCE(ghijklmno) **AND** i='IAH' **AND** l='LAX' **AND** n='2016-03-16' **AND** a=g)}~~
{def | (∃a) (∃b) (∃c) (∃g) (∃h) (∃i) (∃j) (∃k) (∃l) (∃m) (∃n) (∃o) (SEAT_RESERVATION(abcdef) **AND** LEG_INSTANCE(ghijklmno) **AND** i='IAH' **AND** l='LAX' **AND** n='2016-03-16' **AND** a=g)}

d. ~~{abcd | FARE(abcd) **AND** FLIGHT(efg) **AND** f='Delta Airlines' **AND** a=e}~~
{abcd | (∃e) (∃f) (∃g)FARE(abcd) **AND** FLIGHT(efg) **AND** f='Delta Airlines' **AND** a=e}

8.30

Tuple Relational Calculus:

a. {p | R(p) AND p.A = p.C}

c. {p.A, p.B, p.C, f.D, f.E | R(p) AND S(f) AND p.C = f.C}

d. {p.A, p.B, p.C | R(p) OR S(p)}

f. {p.A, p.B, p.C | R(p) AND S(f) AND (p.A≠f.A OR p.B≠f.B OR p.C≠f.C)}

g. {p.A, p.B, p.C, f.D, f.E, f.F | R(p) AND S(f)}

Domain Relational Calculus:

~~a. {abc | (∃c) (R(abc) AND a = c)}~~

{abc | (R(abc) AND a = c)}

~~b. {abcde | (∃d) (R(abc) AND S(def) AND c = d)}~~

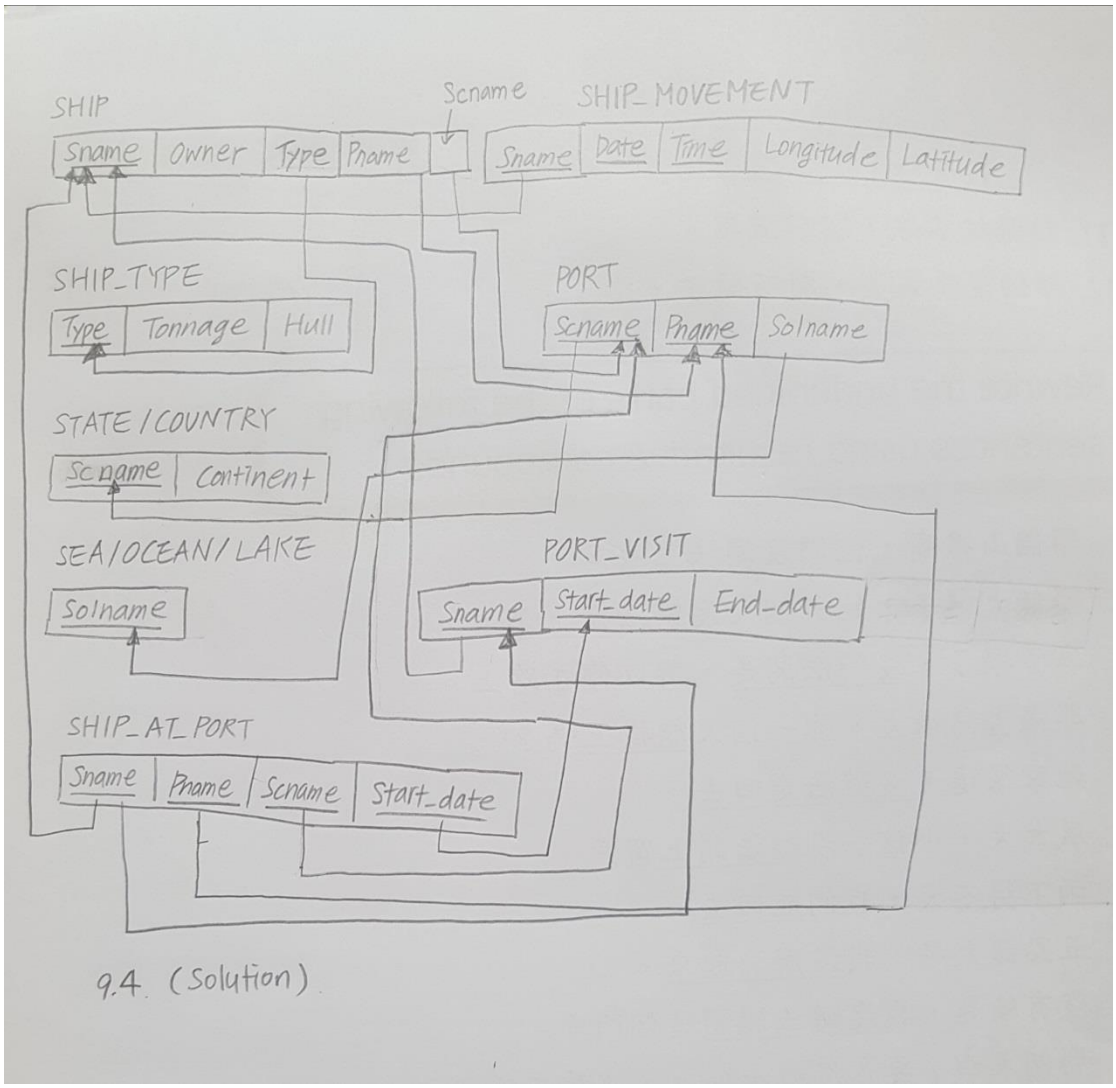
{abcde | (∃f) (R(abc) AND S(def) AND c = d)}

d. {abc | R(abc) OR S(abc)}

f. {abc | (∃d)(∃e)(∃f) (R(abc) AND S(def) AND(a≠d OR b≠e OR c≠f))}

g. {abcdef | R(abc) AND S(def)}

9.4


SHIP
Sname
Owner
Type
Pname
Scname
SHIP_MOVEMENT
Sname
Date
Time
Longitude
Latitude
SHIP_TYPE
Type
Tonnage
Hull
PORT
Scname
Pname
Solname
STATE/COUNTRY
Scname
Continent
SEA/OCEAN/LAKE
Solname
PORT_VISIT
Sname
Start_date
End_date
SHIP_AT_PORT
Sname
Pname
Scname
Start_date

9.4 (Solution)