Assignment 1 Key

Total marks: 75 | 2000-1
Due: Feb 2, 2000 by 20:30 | Instructor: G. Louie

1. (4 marks) Explain the difference between logical and physical data independence.

   Logical data independence means that users are shielded from changes in the logical structure of the data (2 marks), while physical data independence insulates users from changes in the physical storage of the data (2 marks).

2. Given the relations shown below, show the results of the following relational operations:

S

S#	city
S1	Paris
S2	Berne
S3	Oslo
S4	.null.
S5	Paris

P

P#	city
P1	Paris
P2	Berne
P3	Rome
P4	.null.
P5	.null.

J

J#	City
J1	Paris
J2	Rome
J3	Oslo
J4	.null.
J5	Rome

a) (2 marks) The projection of S on city
b) (2 marks) The union of result (a) and the projection of P on city
c) (2 marks) The set difference of result (a) and the projection of P on city (in that order)
d) (2 marks) The intersection of result (c) and the projection of J on city
e) (3 marks) The product of P and J
f) (3 marks) The natural join of P and J
g) (4 marks) The outer natural join of S and P.

(a)

city
Paris
Berne
Oslo

(b)

city
Paris
Berne
Oslo
Rome

(c)

City
Oslo

(d)

City
Oslo

(e)

P#	P_city	J#	J_city
P1	Paris	J1	Paris
P2	Berne	J1	Paris
P3	Rome	J1	Paris
P4	.null.	J1	Paris
P5	.null.	J1	Paris
P1	Paris	J2	Rome
P2	Berne	J2	Rome
P3	Rome	J2	Rome
P4	.null.	J2	Rome
P5	.null.	J2	Rome
P1	Paris	J3	Oslo
P2	Berne	J3	Oslo
P3	Rome	J3	Oslo
P4	.null.	J3	Oslo
P5	.null.	J3	Oslo
P1	Paris	J4	.null.
P2	Berne	J4	.null.
P3	Rome	J4	.null.
P4	.null.	J4	.null.
P5	.null.	J4	.null.
P1	Paris	J5	Rome
P2	Berne	J5	Rome
P3	Rome	J5	Rome
P4	.null.	J5	Rome
P5	.null.	J5	Rome

(f)

P#	city	J#
P1	Paris	J1
P3	Rome	J2
P3	Rome	J5

(g)

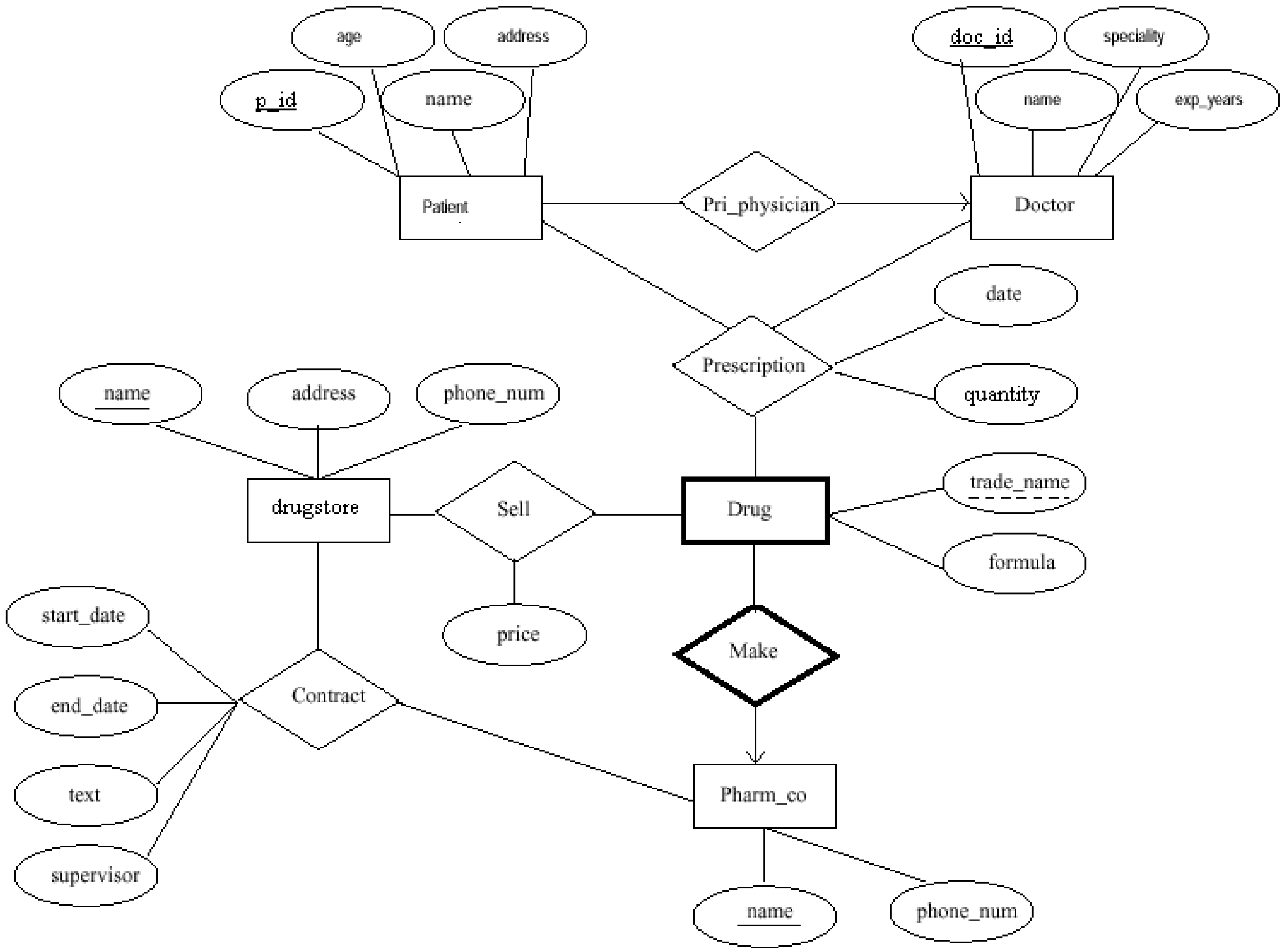
S#	city	P#
S1	Paris	P1
S2	Berne	P2
S3	Oslo	.null.
S5	Paris	P1
.null.	Rome	P3

3. Suppose you are given the following information about a database for a chain of drug stores:
   - a drugstore sells drugs prescribed by doctors to patients,
   - each drugstore in the chain is identified by a store name, address, and a phone number,
   - patients are identified by a patient id, and their names, addresses, and ages must be recorded,

- doctors are identified by a doctor id. Each doctor's name, specialty, and years of experience must be recorded,
- each drug is made by a pharmaceutical company and sold to the drugstore. The drug's trade name identifies the drug uniquely from among the products of that company. For each drug, the trade name and formula must be recorded
- each pharmaceutical company is identified by name and has a phone number,
- every patient has a primary doctor,
- every doctor has at least one patient,
- each drugstore sells several drugs and has a price for each. A drug could be sold at several drugstores, and the price could vary from one drugstore to another,
- doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors,
- each prescription has a date and a quantity associated with it. You can assume that if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored,
- pharmaceutical companies have long-term contracts with drugstores. A pharmaceutical company can contract with several drugstores, and a drugstore can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract,
- drugstores appoint a supervisor for each contract. There must always be a supervisor for each contract, but the contract supervisor can change over the lifetime of the contract,
- if a pharmaceutical company is deleted, you need not keep track of its products any longer.

a) (8 marks) Draw an ER diagram that captures the above information and identify any constraints that are not captured by the diagram.

Here is a sample ER diagram. Depending on the assumptions made (which should be clearly stated), other diagrams are possible! Start w. 8 marks and deduct one for each missing entity, attribute, role indicator, cardinality indicator, etc. Note that the attribute names do not have to match exactly. Note that the entity 'drug' is a weak entity set and the relationship set 'make' is also weak, as indicated by the thicker lines in this diagram as opposed to the double lines used in the textbook.



b) (8 marks) Define the relational schema corresponding to the entity sets and relationship sets. Underline the primary keys.

drugstore(<u>name</u>, address, phone_num)
doctor(<u>doc_id</u>, name, specialty, exp_years)
patient(<u>p_id</u>, name, address, age)
pri_physician(<u>p_id</u>, doc_id)
prescription(<u>p_id</u>, <u>doc_id</u>, date, quantity, <u>trade_name</u>, <u>pharm_name</u>)
drug(<u>pharm_name</u>, <u>trade_name</u>, formula)
sell(<u>store_name</u>, <u>pharm_name</u>, <u>trade_name</u>, price)
contract(<u>store_name</u>, <u>pharm_name</u>, start_date, end_date, text, supervisor)

4. Suppose you are given the following schema:

employee(emp_id, name, salary)
flights(flight_no, from, to, distance, depart_time, arrival_time)
aircraft(aircraft_id, manufacturer, model, range)
certified(emp_id, aircraft_id)

The *certified* relation indicates which employee(s) is/are certified to fly which aircraft. For each of the following queries, give an expression in

i) the relational algebra,
ii) the tuple relational calculus,
iii) the domain relational calculus.

For example, the following expressions would be used to find the names of employees who are certified to fly aircraft manufactured by 'Boeing':

i) $\Pi_{name}(\sigma_{manufacturer = \text{'Boeing'}}(aircraft \bowtie certified \bowtie employee))$

ii) $\{t \mid \exists\, e \in employee\ \exists\, c \in certified\ \exists\, a \in aircraft\ (t[name] = e[name]$
$\wedge\ a[aircraft\_id] = c[aircraft\_id] \wedge e[emp\_id] = c[emp\_id]$
$\wedge\ a[manufacturer] = \text{"Boeing"})\}$

iii) $\{\langle n\rangle \mid \exists\, a, e, m\ (\langle a, e\rangle \in certified \wedge \langle e, n\rangle \in employee \wedge \langle a, m\rangle \in aircraft$
$\wedge\ m = \text{"Boeing"})\}$

a) (3 marks) Find the flight numbers of all the flights originating from Vancouver which depart after "13:00".

i) $\Pi_{flight\_no}(\sigma_{from = \text{'Vancouver'} \wedge\ depart\_time > \text{'13:00'}}\ (flights))$

ii) $\{t \mid \exists\, f \in flights\ (t[flight\_no] = f[flight\_no]$
$\wedge\ f[from] = \text{"Vancouver"} \wedge f[depart\_time] > \text{"13:00"})\}$

iii) $\{\langle fn\rangle \mid \exists\, fr, d\ (\langle fn, fr, d\rangle \in flights \wedge fr = \text{"Vancouver"} \wedge d > \text{"13:00"})\}$

b) (6 marks) Find the employee id's of the pilots certified to fly aircraft manufactured by "Boeing".

i) $\Pi_{emp\_id}(\sigma_{manufacturer\ =\ \text{'Boeing'}}(aircraft \bowtie certified))$

ii) $\{t \mid \exists\, c \in certified\ \exists\, a \in aircraft\ (t[emp\_id] = c[emp\_id]$
$\wedge\ a[aircraft\_id] = c[aircraft\_id] \wedge a[manufacturer] = \text{"Boeing"})\}$

iii) $\{<e> \mid \exists\, a, m\ (<a, e> \in certified \wedge <a, m> \in aircraft \wedge m = \text{"Boeing"})\}$

c) (6 marks) Find the aircraft_ids of all aircraft that can be used on non-stop flights (*i.e.* where the aircraft.range > flights.distance) from "Vancouver" to "Tokyo".

i) $\Pi_{aircraft\_id}(\sigma_{range\ >\ distance}\ (aircraft \times (\sigma_{from\ =\ \text{'Vancouver'}\ \wedge\ to\ =\ \text{'Tokyo'}}\ (flights))))$

ii) $\{t \mid \exists\, a \in aircraft\ \exists\, f \in flights\ (t[aircraft\_id] = a[aircraft\_id]$
$\wedge\ f[from] = \text{"Vancouver"} \wedge f[to] = \text{"Tokyo"}) \wedge a[range] > f[distance]\}$

iii) $\{<a> \mid \exists\, r, f, t, d\ (<a, r> \in aircraft \wedge <f, t, d> \in flights \wedge f = \text{"Vancouver"}$
$\wedge\ t = \text{"Tokyo"} \wedge r > d)\}$

d) (6 marks) Identify the flight numbers that can be piloted by every pilot whose salary is more than $100,000.

Note: the key is to realize that only those flights with distances less than a given aircraft's range and the given aircraft are certified with pilots earning more than $100,000 can meet the requirements of the query.

i) $\Pi_{flight\_no}(\sigma_{range\ >\ distance\ \wedge\ salary\ >\ 100{,}000}(aircraft \bowtie flights \bowtie certified \bowtie employee))$

ii) $\{t \mid \exists\, e \in employee\ \exists\, f \in flights\ \exists\, c \in certified\ \exists\, a \in aircraft$
$(t[flight\_no] = f[flight\_no] \wedge a[range] > f[distance]$
$\wedge\ e[salary] > 100{,}000 \wedge a[aircraft\_id] = c[aircraft\_id]$
$\wedge\ e[emp\_id] = c[emp\_id])\}$

iii) $\{<f> \mid \exists\, e, a, r, d, s\ (<e, a> \in certified \wedge <a, r> \in aircraft \wedge <f, d> \in flights$
$\wedge\ <e, s> \in employee \wedge s > 100{,}000 \wedge r > d)\}$

e) (6 marks) Find the names of pilots who can operate planes with a range greater than 3,000 miles but are not certified on any aircraft manufactured by "Boeing".

i) $\Pi_{name}(employee \bowtie (\Pi_{emp\_id}(\sigma_{range > 3000}(aircraft \bowtie certified))$

$- \Pi_{emp\_id}(\sigma_{manufacturer = \text{"Boeing"}}(aircraft \bowtie certified))))$

ii) $\{t \mid \exists\, e \in employee\ \exists\, c \in certified\ (\exists\, a \in aircraft$

$(t[name] = e[name] \wedge a[aircraft\_id] = c[aircraft\_id]$

$\wedge\ e[emp\_id] = c[emp\_id] \wedge a[range] > 3000))$

$\wedge\ \neg(\exists\, c2 \in certified\ (\exists\, a2 \in aircraft\ (a2[manufacturer] = \text{"Boeing"}$

$\wedge\ a2[aircraft\_id] = c2[aircraft\_id] \wedge e[emp\_id] = c2[emp\_id]))))\}$

iii) $\{<n> \mid \exists\, e, a, r, a2, m\ (<e, a> \in certified \wedge <a, r> \in aircraft$

$\wedge <e, n> \in employee \wedge r > 3000 \wedge \neg(<e, a2> \in certified$

$\wedge <a2, m> \in aircraft \wedge m = \text{"Boeing"}))\}$

f) (6 marks) Find the employee id's of the employees who make the highest salary.

Note: first find all the employees who do not have the highest salary and subtract these from the original list of all employees. The remaining employees will have the highest salary.

i) $\Pi_{emp\_id}(employee) - (\Pi_{e2.emp\_id}(employee \bowtie_{employee.salary > e2.salary} \rho_{e2}(employee)))$

ii) $\{t \mid \exists\, e1 \in employee\ (t[emp\_id] = e2[emp\_id]$

$\wedge\ \neg(\exists\, e2 \in employee\ (e2[salary] > e1[salary]))\}$

iii) $\{<e1> \mid \exists\, e2, s1, s2\ (<e1, s1> \in employees$

$\wedge\ \neg(<e2, s2> \in employees \wedge s2 > s1)\}$

5. (4 marks) What is an unsafe query? Give an example and explain why it is important to disallow such queries.

An unsafe query is a query in relational calculus which returns an infinite number of results. It is important to disallow such queries because a database needs to return results for a query after a finite amount of time. An example of an unsafe query is:

$$\{ e \mid \neg (e \in employees)\}$$

This query returns all objects which are not employees. The results of this query is infinite and thus the query is unsafe. (2 marks for definition/explanation, 2 marks for a reasonable example).