Example Paper

THE UNIVERSITY OF SYDNEY

SEAT NUMBER: ....

STUDENT ID: ....

COMP9120
Relational Database Management Systems

Practice Questions
Semester 1, 2018

This examination paper consists of 10 pages

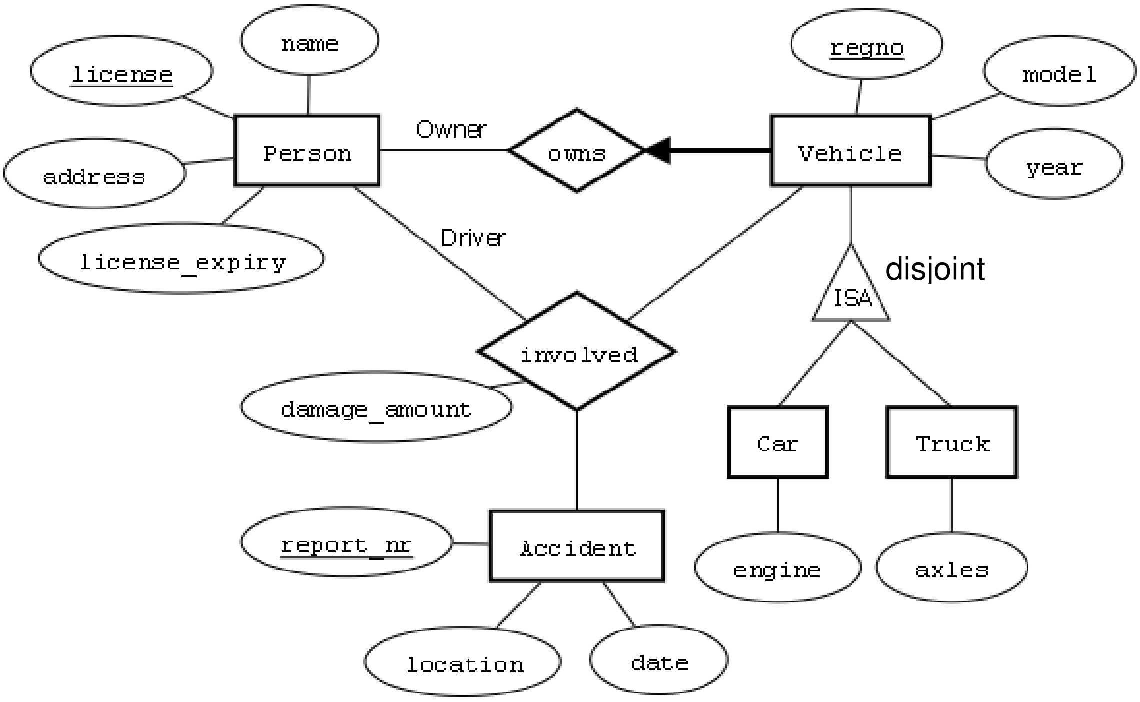
INSTRUCTIONS TO CANDIDATES

1. In a real exam these questions would take 96 minutes, plus 10 minutes reading time. You may find it helpful to time yourself to see how much you can do in this period.
2. You should try to do any necessary calculations in your head or with a simple calculator.
3. The paper comprises 3 questions each with multiple parts. ANSWER ALL THREE QUESTIONS.
4. The mark available for each question is indicated beside the question heading. Each question consists of several parts, and the associated points are also indicated.
5. Answer all questions in the spaces provided on this question paper.
6. If you need more space to write, use additional paper and attach to the exam paper.
7. Take care to write legibly. Write your final answers in ink, not pencil.

Question	Mark	Out of
Question 1		/30
Question 2		/26
Question 3		/24
		/80

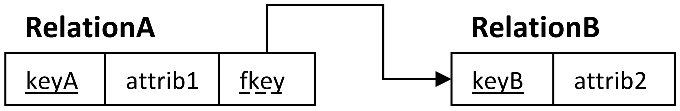
Question 1: ER mapping and Data Integrity [30 points]

Consider the following ER diagram:



It describes a database that stores information about traffic accidents, recording which vehicles were involved and who was driving them. Use this diagram in the following parts of this question. Space has been left to give you room to write your answers across several pages, so take care to make sure you attempt all parts.

In questions asking you to translate an ER diagram into relations using a graphical notation, you must use the notation below:



Make sure your writing is clearly legible, taking special care when indicating that a column should be both a primary key and a foreign key.

a) Translate the ER diagram given above into relations using a graphical notation. Clearly indicate all primary and foreign keys.

[10 points]

b) Give the SQL CREATE TABLE statements needed to create a database for your schema. Include reasonable data types, and other required domain constraints (eg: NOT NULL), primary key and referential integrity (foreign key) constraints.

[12 points]

c) Write the appropriate alter table SQL code to add a table check constraint that evaluates whether a truck has 2 or more axles.

[3 points]

```
ALTER TABLE Truck  ADD CHECK(axles >= 2);
```

d) Write an assertion that checks if all driver licenses are valid until at least January 1, 2015.

[3 points]

```
CREATE ASSERTION check_license CHECK(license_expiry > DATE '2015-01-01');
```

```
CREATE ASSERTION check_license
CHECK(
    NOT EXISTS(
         SELECT 1
         FROM Person
         license_expity < '2015-01-01'
    )
);
```

e) Briefly describe one possible advantage and one possible disadvantage of SQL assertions relative to table check constraints.

[2 points]

Adv: can check an crossed table assertion (does not like CHECK)
Disadv: It cost a lot since every time modify the table, we need to traverse the table.

"Advantage: Can enforce cross-table assertions, which CHECK constraints cannot."

disadvantage: Not adopted by major DBMS';

Question 2: Normal Forms [26 points]

Consider the following relation and its functional dependencies provided below, describing the visits to patients (identified with *PatID*) by doctors (with ID *ProvNo*) and the resulting diagnoses.

VisitID	VisitDate	PatID	Age	City	Zip	ProvNo	ProvSpeciality	Diagnosis
V10020	13/01/2004	P1	35	Denver	80217	D1	Internist	Ear Infection
V10020	13/01/2004	P1	35	Denver	80217	D2	Practitioner	Influenza
V93030	20/01/2004	P3	17	Ewood	80113	D2	Practitioner	Pregnancy
V82110	18/01/2004	P2	60	Boulder	85932	D3	Cardiologist	Murmur

PatID → Age, City, Zip

Zip → City

ProvNo → ProvSpeciality

VisitID → PatID, VisitDate, Age, City, Zip

VisitID, ProNo → Diagnosis

Each visit is for at most one patient,
it can take place on at most one day,
in at most one city,
in at most one zip code,
and for at most one age.

a) Give an interpretation in plain English for each of the functional dependencies.

[10 points]

b) Identify an insert anomaly, delete anomaly, and update anomaly.

[3 points]

INSERT: VisitID 10020
DELETE:
UPDATE:

Insert anomaly: We cannot insert a new ProvNo to describe a doctor’s speciality unless we also include a VisitID (i.e., we can’t record details about a doctor unless they’ve made a visit).

Delete anomaly: If we delete the only record of a visit made by a doctor, we lose the information about their speciality.

Update anomaly: If a patient’s address changes, we must update this address for every row in which that patient appears.

c) Identify all candidate keys.

[3 points]

d) Construct a set of BCNF relations decomposed from the original relation. Use the graphical notation described at the beginning of this paper to clearly indicate all primary and foreign keys.

[10 points]

Question 3: SQL and Relational Algebra [24 points]

Answer the given questions based on the following relational schema:

```
PowerClass(powerClass, min_consumption, max_consumption)
Device(maker, dev, price, powerClass)
Television(dev, type, screensize, hdtv)
Projector(dev, resolution, lumen, noise, interface)
DVDplayer(dev, discs, format, bits, sn_ratio)
```

a) Write an SQL query to find the count and the average screensize of televisions costing more than $2000.

[3 points]

```
SELECT COUNT(*), AVG(screensize)
FROM Device d
JOIN Television t ON d.dev = t.dev
WHERE price > 2000
```

b) Assume the following example database tuples for the above schema:

```
Device(’Alex’, ’T20FV300’, 300, 1)
Device(’Alex’, ’T34HS510’, 2000, 2)
Device(’Bundig’, ’B45’, 3899, 1)
Device(’ABC’, ’ABC tba’, 2500, 1)

Television(’T20FV300’, ’trinitron’, 20, FALSE)
Television(’T34HS510’, ’tube’, 34, TRUE)
Television(’B45’, ’Plasma’, 42, TRUE)
Television(’ABC tba’, ’Plasma’, null, null)
```

What will be the result of your SQL query from the previous subquestion (a) on this example database?

[3 points]

(2,42)
since AVG does not count NULL

c) Write an SQL query for “what are the cheapest (lowest price) DVDPlayers made by ’TEC’ and what do they cost?”

```
SELECT dv.dev, de.price
FROM DVDplayers dv
JOIN Device de ON dv.dev = de.dev
WHERE de.maker = 'TEC'
AND de.price <= ALL (
    SELECT de.price
    FROM DVDplayers dv
    JOIN Device de ON dv.dev = de.dev
    WHERE de.maker = 'TEC'
);
```

[6 points]

d) Consider the following SQL query:

```
SELECT p.dev, c.max_consumption
FROM Device p, Projector r, PowerClass c
WHERE p.maker = ’Alex’ AND
p.dev = r.dev AND
p.powerClass = c.powerClass
```

Write down an equivalent expression in relational algebra.

[4 points]

$$\pi_{dev,\ max\_consumption}\ (\sigma\ (D \bowtie P \bowtie C))$$

e) Write an SQL for the following: For each manufacturer (maker), find the price range (lowest and highest price) and the average price of televisions and projectors made by them. The result shall show the makers, the corresponding price ranges (i.e., minimum and maximum) and average of the prices. Results should only be shown for manufacturers where at least 3 devices are considered in calculating the results.

[8 points]

```
SELECT maker, MIN(price), MAX(price), AVG(price)
FROM Device
GROUP BY maker
HAVING COUNT(dev) >= 3;
```

```
SELECT maker, MIN(price), MAX(price), AVG(price)
FROM Device
NATURAL JOIN (
        SELECT dev
        FROM Television
UNION
        SELECT dev
        FROM Projector)
GROUP BY maker
HAVING COUNT(*) >= 3;
```