Student number:_____ **Name**:_____

UNIVERSITY OF VICTORIA
Faculty of Engineering
Department of Computer Science

CSC 370 (Database Systems)
Instructor: Daniel M. German

Midterm
Oct 15, 2012

Duration: 75 minutes

This is a closed-book exam.

This examination paper consists of **6** pages and **2** sections. Please bring any discrepancy to the attention of an invigilator. The number in parenthesis at the start of each question is the number of points the question is worth.

Answer all questions.

Please write your answers clearly.

For instructor's use:

	Score
1 (44)	
2 (4)	
Total (48)	

For this exam, consider the following schema and instances of the relations. Feel free to remove this page from the exam.

Our database is very simple. It is composed of three relations: *Parts*, *Suppliers* and *Catalog*. The *Catalog* table contains the parts that are being offered by a given supplier at a given price (a part is missing a price if this field is NULL). Every *pid* in *Catalog* exists in *Parts*, and every *sid* in *Catalog* exists in *Suppliers*.

```
Parts(pid: integer, pname character(40), color character(20));
```

Primary key: *pid*.

pid	*pname*	*color*
6	Anti-Gravity Turbine Generator	Cyan
7	Anti-Gravity Turbine Generator	Magenta
8	Fire Hydrant Cap	Red
9	7 Segment Display	Green
10	SQL queries	Green

```
Suppliers(sid: character(10), sname: character(40), address: char(50));
```

Primary key: *sid*.

sid	*sname*	*address*
amazon	Amazon Canada	1 Grub St., Potemkin Village, IL 61801
walmart	Walmart Inc	4 My Way, Bermuda Shorts, OR 90305
rim	Research in Motion	99999 Short Pier, Terra Del Fuego, TX 41299
google	Google Inc.	2 Groom Lake, Rachel, NV 51902

```
Catalog(sid: character(10), pid: integer, price: real);
```

Primary key: *(sid,pid)*.

sid	*pid*	*cost*
amazon	8	11.7
walmart	8	7.95
rim	8	12.5
rim	9	1
amazon	10	10.5
amazon	9	

1. Writing queries in Relational Algebra and SQL

Give both relational algebra and SQL queries to answer the following questions. **Your relational algebra should match your SQL queries.**

1.1) [4] For every supplier, lists its *sname* and the *pid* of each of the parts they offer. Result should contain two attributes.

$$\Pi_{sname,\ pid}(C \bowtie S)$$

```
SELECT sname, pid FROM
  Catalog NATURAL JOIN Suppliers;
```

1.2) [4] List the *pname* of parts that are being offered at $10 or more. Result should contain only one attribute.

$$\Pi_{pname}\sigma_{price>=10}(C \bowtie P)$$

```
SELECT pname FROM
   Catalog NATURAL JOIN Parts
   WHERE PRICE >= 10;
```

1.3) [4] For every *pid* in relation *Parts*, list the number of suppliers that offer it, and the minimal price at which it is offered. Result should contain three attributes.

$$\Pi_{pid,count,m}\gamma_{pid,count(sid)\rightarrow count,min(price)\rightarrow m}(P \bowtie_L C)$$

```
SELECT pid, count(sid) as count, min(price) as m FROM
   Parts NATURAL LEFT JOIN Catalog
   GROUP BY pid;
```

1.4) [4] How many parts in table *Parts* are not being offered by any supplier? Result should contain only one attribute.

$$\gamma_{count(pid)}(\Pi_{pid}P - \Pi_{pid}C)$$

```
SELECT count(pid) FROM
  (SELECT pid FROM Parts EXCEPT SELECT pid FROM Catalog);
```

1.5) [4] List the *pid* and *sid* of parts that offered by such supplier and are missing a *price*. Result should contain two attributes.

$$\Pi_{pid,sid}\sigma_{price\ is\ NULL}C$$

```
SELECT pid, sid FROM
   Catalog
   WHERE price is NULL;
```

1.6) [4] For every supplier, list its *sid* and the average *price* of the parts they offer. Result should contain two attributes.

$$\Pi_{sid,avg}\gamma_{sid,avg(price)}C$$

```
SELECT sid, avg(price) AS avg FROM
   Catalog
   GROUP BY sid
```

1.7) [4] List the *pid* and the *pname* of parts that are offered by exactly 3 suppliers. Result should contain two attributes.

$$\Pi_{pid,pname,count}(\sigma_{count=3}\gamma_{pid,count(sid)\rightarrow count}C) \bowtie P$$

```
SELECT pid, pname FROM
   (SELECT pid, count(sid) AS count FROM
    Catalog
    GROUP BY pid
    HAVING count = 3)
   NATURAL JOIN Parts
```

1.8) [4] List the *pid* of the parts that are being offered by both suppliers: `Amazon` and `Walmart` (these are their *sid*). Result should contain one attribute.

$$(\Pi_{pid}\sigma_{sid='Amazon'}C) \bigcap (\Pi_{pid}\sigma_{sid='Walmart'}C)$$

```
SELECT pid FROM Catalog WHERE sid = 'Amazon'
INTERSECT
SELECT pid FROM Catalog WHERE sid = 'Walmart'
```

1.9) [4] Compute the difference between the average price of parts with *pid* `12` and `32`. In other words, compute (the average price of partid `12`) minus (the average price of pid `32`). The result should contain one tuple with one attribute.

$$\Pi_{x-y}[(\gamma_{avg(price)\rightarrow x}\sigma_{pid=12}C) \times (\gamma_{avg(price)\rightarrow y}\sigma_{pid=32}C)]$$

```
SELECT x - y FROM
   (SELECT avg(price) AS x FROM
      Catalog
      WHERE pid = 12
      GROUP by pid) as A,
   (SELECT avg(price) AS y FROM
      Catalog
      WHERE pid = 32
      GROUP by pid) as B
```

1.10) [4] For every *pid* in the relation *Catalog*, list the *sname* of the supplier who offers it a the lowest *price*, and such *price*. Result should contain three attributes.

This requires a bit of explanation. First get the minimum price of each part. Call it M. Then use this to find the tuples in C that have this price (you can do this with a join or an IN, then join this to Suppliers to find their sname.

$$M = \gamma_{pid,min(price)\rightarrow price}C$$

$$\Pi_{pid,sname,price}(M \bowtie C \bowtie S)$$

```
SELECT pid, sname, price FROM
   (SELECT pid, min(price) as price) FROM CATALOG) as M
   NATURAL JOIN Parts NATURAL JOIN Catalog
```

1.11) [4] List the *pid* of parts that are being offered by at least two suppliers at exactly the same price. Your result should contain two columns: the *pid* of the two parts, and their *price*.

$$\Pi_{pid,price}\sigma_{c>=2}\gamma_{pid,price,count(sid)\rightarrow c}C$$

```
SELECT pid, price FROM
Catalog
GROUP BY pid, price
HAVING count(sid) >= 2
```

2. Relational Model

2.1) [4] Given the relation $R(A, B, C, D)$ and the set of functional dependencies $A \rightarrow BC$, $BC \rightarrow A$, and $B \rightarrow D$. Find all the candidate keys of this relation. Show all your work.

For this you have to compute the closure of each combination of attributes ABCD, ABC, ABD, ... A, B, C, D (15 in total). The candidate keys are only A and BC.

End of examination
Total pages: 6
Total marks: 48