Fall 2007 – Introduction to Databases
Wednesday, October 31, 2007
Midterm Test

This is a closed book test. You have 90 minutes to complete your answers, worth a total of 250 points, and 25% of your final mark. Good luck!

Last Name: _____

First Name: _____

Last Name: _____

First Name: _____

Student Number: _____

Question 1. _____/50

Question 2. _____/30

Question 3. _____/50

Question 4. _____/50

Question 5. _____/70

TOTAL _____/250

(Blank Page)

Question 1 [Computing queries – 50 points]. Consider the following relations

R(A,B,C) = {<1,2,3>,<1,2,0>,<1,3,1>,<6,2,3>,<1,4,2>,<3,1,4>}
S(B,C,D) = {<2,3,7>,<1,4,5>,<1,2,3>,<2,3,4>,<3,1,4>}

(1.a) [10 points] Compute the result of the following relational algebra expression:

$\pi_{A, D}(R \bowtie S)$

R ⋈ S = A,B,C,D
1,2,3,7
1,2,3,4
1,3,1,4
6,2,3,7
6,2,3,4
3,1,4,5

$\pi_{A, D}(R \bowtie S)$ = A,D
1,7
1,4
1,4
6,7
6,4
3,5

A,D (collapsed)
1,7
1,4
6,7
6,4
3,5

(1.b)[20 points] Compute the result of the following SQL query:

```
SELECT R.B, AVG(S.B)
FROM R, S
WHERE R.A = S.C AND S.D < 7
GROUP BY R.B
```

```
SELECT *
FROM R, S
WHERE R.A = S.C AND S.D <7;
```

```
A | B | C | B | C | D
---+---+---+---+---+---
 1 | 2 | 0 | 3 | 1 | 4
 1 | 3 | 1 | 3 | 1 | 4
 1 | 4 | 2 | 3 | 1 | 4
 1 | 2 | 3 | 3 | 1 | 4
 3 | 1 | 4 | 2 | 3 | 4
(5 ROWS)
```

```
SELECT R.B, AVG(S.B)
FROM R, S
WHERE R.A = S.C AND S.D < 7
GROUP BY R.B
```

```
B |        AVG
---+--------------------
 1 | 2.0000000000000000
 2 | 3.0000000000000000
 3 | 3.0000000000000000
 4 | 3.0000000000000000
(4 ROWS)
```

(1.c) [20 points] Compute the result of the following SQL query:

```
SELECT DISTINCT S.B, MIN(S.C)
FROM S
GROUP BY S.B
HAVING COUNT(DISTINCT S.D) > 1;
```

```
B | MIN
---+-----
 1 |  2
 2 |  3
(2 ROWS)
```

Question 2 [SQL DDL – 30 points]. Consider the following clothing inventory schema, to be used in the next three questions. The schema is about clothing items, with an emphasis on dresses:

- `ClothingItem(mfg,modelNbr,type)` – `mfg` is the manufacturer of the item; `type` takes only these values: `"dress", "shirt", "belt"`; `modelNbr` is the integer identifier which uniquely identifies each clothing item.
- `Dresses(modelNbr,size,colour,qty,price)` – The `modelNbr`, `size` and `quantity` are all numbers which you may assume to be integers; the `colour` is expressed in up to 20 characters; and the price is a dollar and cents value (represented as a number up to 9999.99).
- `Dresses(modelNbr)` $\subseteq$ `ClothingItem(modelNbr)`.

Give the DDL schema that captures these relations, their keys and referential constraints, as well as the additional constraints:

- A dress cannot be removed from `ClothingItem` if it exists in `Dresses`.
- If a model number is changed in `ClothingItem` it is also changed in `Dresses`.

```
CREATE TABLE clothingItem(
  mfg VARCHAR(20) NOT NULL,
  modelNbr NUMERIC PRIMARY KEY,
  type VARCHAR(20) NOT NULL,
  CHECK (((type = 'shirt') OR (type = 'dress')) OR (type = 'belt'))
  )

CREATE TABLE dresses(
 modelNbr NUMERIC,
 size NUMERIC,
 colour VARCHAR(10) ,
 qty NUMERIC,
 price NUMERIC(6,2),
 PRIMARY KEY (modelNbr,size,colour),
 FOREIGN KEY (modelNbr) REFERENCES clothingItem(modelNbr)
       ON DELETE RESTRICT ON UPDATE CASCADE) (or ON DELETE NO ACTION)
```

(Blank Page)

Question 3 [Relational Algebra – 50 points]. Write relational algebra expressions that compute the following queries:

```
ClothingItem(mfg,modelNbr,type)
Dresses(modelNbr,size,colour,qty,price)
```

(3.a) [10 points] "Find all the manufacturers who make red dresses in any large size greater than 16"

$\pi_{mfg}(\sigma_{colour = \text{'red'},\ size > 16}(\texttt{ClothingItem} \bowtie \texttt{Dresses}))$

(3.b) [20 points] "Find the manufacturers who make red dresses and do not make black dresses"

$\pi_{mfg}(\sigma_{colour = \text{'red'}}\ (\texttt{ClothingItem} \bowtie \texttt{Dresses}))$

$- \pi_{mfg}(\sigma_{colour = \text{'black}}\ (\texttt{ClothingItem} \bowtie \texttt{Dresses}))$

(3.c) [20 points] "Find all the highest priced dresses and return them along with their manufacturers" (Assume that every model is the same price across all colours and sizes.)

$\pi_{mfg,modelNbr}$ (ClothingItem ⋈

($\pi_{\underline{modelNbr,price}}$ Dresses

- $\pi_{\underline{modelNbr,price}}$ ($\sigma_{price1 > price}$ (($\rho_{modelNbr,price \to modelNbr1,price1}$ ($\pi_{\underline{modelNbr,price}}$ Dresses)) ⋈

($\pi_{\underline{modelNbr,price}}$ Dresses)))))

Question 4 [SQL – 50 points]. Write SQL expressions that compute the following queries:

```
ClothingItem(mfg,modelNbr,type)
Dresses(modelNbr,size,colour,qty,price)
```

(4.a) [10 points] "Find all dress models and their manufacturers for dresses with a non-zero quantity. Return each model no more than once."

```
SELECT DISTINCT c.mfg, d.modelnbr
  FROM clothingitem c, dresses d
  WHERE c.modelnbr = d.modelnbr and d.qty > 0
```

(4.b) [20 points] "Find the average price of a dress for each dress manufacturer, and return the results ordered alphabetically by manufacturer name"

```
SELECT c.mfg, avg(price)
  FROM clothingitem c, dresses d
  WHERE c.modelnbr = d.modelnbr
  GROUP BY c.mfg
  ORDER BY c.mfg asc
```

(4.c) [20 points] "Find the most expensive dresses that are size 12 or less which have 2 or more of each size in stock, and return each dress model along with their manufacturers exactly once "

```
SELECT DISTINCT modelnbr, price
  FROM clothingitem c, dresses d
  WHERE d.size <= 12
    AND c.modelnbr = d.modelnbr
    AND d.price =
      (SELECT max(price)
         FROM dresses)
    AND d.modelnbr NOT IN
      (SELECT modelnbr
         FROM dresses
        WHERE size <= 12
          AND qty < 2)
```

Question 5 [True/False questions – 70 points]. For each of the following statements, indicate whether they are true of false. A correct answer is worth 5 points, no answer is worth 0 points, wrong answer is worth -3 points.

(5.a) _ F _ A relation may have multiple keys, but only one superkey;

(5.b) _ F _ In SQL `DROP TABLE XYZ` will delete any table `XYZ` and all of its contents;

(5.c) _ F _ Some primary keys allow null values;

(5.d) _ T _ The value `'00000'` of `SQLSTATE` means the last command was executed successfully;

(5.e) _ F _ The constraint '`ON DELETE NO ACTION`' indicates that if the field referenced changes, don't do anything to it on the current table;

(5.f) _ T _ The Union operation (∪) cannot be performed between any two relations;

(5.g) _ F _ In relational algebra selection (**σ**) operates on the columns or attributes of a relation and projection (**π**) operates on the rows or tuples of a relation;

(5.h) _ T _ The result of a projection operation contains at most as many rows as the operand relation;

(5.i) _ F _ The cardinality of a natural join between two relations **A** and **B** with no common attributes between them, is equal to the cardinality of **A** plus the cardinality of **B**;

(5.j) _ F _ Query execution plans are only prepared by the DBMS for embedded SQL using a Statement-level interface and pre-compiler;

(5.k) _ F _ JDBC allows at most one database to be open at any one time;

(5.l) _ F _ Cursors that are not insensitive cannot detect any changes to the underlying database after the initial query is performed;

(5.m) _ T _ Cursors address the problem of impedence mismatch allowing rows to be processed one at a time;

(5.n) _ T _ JDBC uses a call-level interface to execute SQL from a JAVA program.