School of Computer Science and Software Engineering

1st SEMESTER EXAMINATIONS 2010

CITS2232 DATABASES

SURNAME: ___________________________ STUDENT NO: ___________
GIVEN NAMES: ______________________ SIGNATURE: ___________

This paper contains: 9 pages (including the title page)
Time allowed: 2 hours 10 minutes

Section A: 1 question 30 marks
Section B: 2 questions 30 marks
Section C: 4 questions 40 marks
TOTAL MARKS: 100 marks

You should attempt ALL questions in this paper.
The questions should be answered in the supplied answer booklet.

PLEASE NOTE

Examination candidates may only bring authorised materials into the examination room. If a supervisor finds, during the examination, that you have unauthorised material, in whatever form, in the vicinity of your desk or on your person, whether in the examination room or the toilets or en route to/from the toilets, the matter will be reported to the head of school and disciplinary action will normally be taken against you. This action may result in your being deprived of any credit for this examination or even, in some cases, for the whole unit. This will apply regardless of whether the material has been used at the time it is found.

Therefore, any candidate who has brought any unauthorised material whatsoever into the examination room should declare it to the supervisor immediately. Candidates who are uncertain whether any material is authorised should ask the supervisor for clarification.
Instructions

This paper contains three sections totalling 100 marks.
Section A is worth 30 marks.
Section B is worth 30 marks.
Section C is worth 40 marks.

Candidates should attempt ALL questions in this paper.

*Please fill in the question numbers in the spaces on the front of the answer booklet.*
Section A

1. Consider a database with three tables declared as follows:

```
CREATE TABLE part (
    pid: integer,
    description: varchar(128),
    PRIMARY KEY (pid)
) Engine = InnoDB;

CREATE TABLE warehouse (
    wid: integer,
    address: varchar(128),
    PRIMARY KEY (wid)
) Engine = InnoDB;

CREATE TABLE stock (
    pid: INTEGER,
    wid: INTEGER,
    quantity: INTEGER,
    unitcost: FLOAT
    FOREIGN KEY (pid) REFERENCES part (pid),
    FOREIGN KEY (wid) REFERENCES warehouse (wid)
) Engine = InnoDB;
```

These tables contain information about various machinery parts stored in different warehouse locations.

The `stock` table indicates the number of items currently held at a particular warehouse, and the cost of each item if it were to be delivered from that warehouse. For example, suppose that the table contained the following rows:

<table>
<thead>
<tr>
<th>pid</th>
<th>wid</th>
<th>quantity</th>
<th>unitcost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
<td>23.50</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>15</td>
<td>22.95</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>22.00</td>
</tr>
</tbody>
</table>

This would indicate “Warehouse 1 has 10 units in stock of Part 1 and each one costs $23.50, while Warehouse 2 has 15 units in stock at a cost of $22.95 each, etc.”
Suppose that you frequently need to answer questions of the form

“What is the cheapest total cost of an order of 23 units of Part 1?”

where the number of units and the part id change from query to query. In this case the cheapest way of supplying this order would be to purchase all 7 items in stock at Warehouse 3, all 15 from Warehouse 2 and just 1 item from Warehouse 1 at a total cost of

\[ 7 \times 22.00 + 15 \times 22.95 + 1 \times 23.50 = 521.75 \]

(a) Write a Java method

```java
public double totalCost(int pid, int numItems)
```

that computes the cheapest total cost of supplying the specified number of units of part with id pid. You may assume that the method has access to a variable conn that refers to an already-created `java.sql.Connection`.

Your Java methods may NOT call the stored procedures of the following question.

(b) Write a MySQL stored procedure `totalCost` that takes two IN parameters `pid` and `numItems` and has a single OUT parameter. This procedure should calculate the cheapest total cost of fulfilling this order and assign this value to the OUT parameter.
Section B

2. Queries in SQL

Consider the following schema which represents information in a customer order database.

CUSTOMERS(cnum:INT, cname:STRING, address:STRING) – Customer information
ORDERS(onum:INT, date:DATE, cnum:INT) – Customer orders
ITEMS(inum:INT, iname:STRING, unitprice:DECIMAL[6,2]) – Items available for order
ORDERED(onum:INT, inum:INT, quantity:INT) – Quantity of an item in an order

All identically named attributes are shared keys across schemas.

(a) Write a MySQL query to display the order number, date, customer number, the item name and quantity of each item in order number 1001.

(b) Write a MySQL query to display each item name ordered in order number 1357, its unit price and the total cost of each item in the order.

(c) Write a MySQL query to display the names and addresses of customers who have not placed orders.

(d) Write a MySQL query to list the order number, item number and quantity for all orders for which the quantity of the item ordered is greater than the average quantity ordered of that item.
3. **ER Diagramming**

(a) The **Student** entity contains `student_id` (the key), `name`, `birthdate` and `major`. A **Course** is represented by a `course_id` (the key) and `title` and `semester`. All students must **enrol** for the courses they are undertaking, and the `semester` of their enrolment must be recorded.

Draw an ER diagram (using the convention employed in lectures) that captures the above relationship. Your diagram must incorporate all the necessary attributes and constraints.

(2)

Translate your ER diagram into a relational schema(s) (using the descriptive syntax employed in Question 2 above) that best implements the above relationship. Clearly indicate any foreign keys.

(2)

(b) The **Employee** entity contains `employee_ID` (the key), `name` and `birthdate`. Each employee (the **reviewee**) must undertake a **Review**. They are reviewed by **one** fellow employee (the **reviewer**).

Draw an ER diagram that captures the above relationship. Your diagram must incorporate all the necessary attributes and constraints.

(2)

Translate your ER diagram into a relational schema(s) that best implements the above relationship. Clearly indicate any foreign keys.

(3)

(c) **Items** in a hardware are identified by the `item_num` (the key), `name` and the `unitcost`. Each item can **contain** or be made up of other items. We need to keep track of how many items are needed to make the other item.

Draw an ER diagram that captures the above relationship. Your diagram must incorporate all the necessary attributes and constraints.

(2)

Translate your ER diagram into a relational schema(s) that best implements the above relationship. Clearly indicate any foreign keys.

(4)
Section C

4. Data Mining

Consider the following data for market-basket analysis. The data refers to transactions involving purchasable items $A – E$. Each transaction is denoted $T_i$.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$T_2$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$T_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$T_4$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$T_5$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(a) Calculate all the frequent itemsets with minimum support 70%.

(4)

(b) What is the confidence in the association rule {$D \Rightarrow A$}?

(2)

(c) What level of confidence would you have with the association rule that having purchased items $C$ and $D$, item $A$ would also be purchased ({$CD \Rightarrow A$})? What is the confidence in having purchased item $A$, items $C$ and $D$ would also be purchased ({$A \Rightarrow CD$})?

(4)
5. **Normalization**

Consider the relation \( R = (ABCDEF) \) with the set of functional dependencies \( A \rightarrow BCE, ABD \rightarrow CF, B \rightarrow DE, AC \rightarrow DE \).

(a) Derive the candidate key(s) of the relation \( R \). \hspace{1cm} (3)

(b) Does the decomposition of \( R \) into \( (ABDE) \) and \( (CEF) \) result in a lossless-join? \hspace{1cm} (2)

(c) Use the synthesis approach to give a dependency-preserving decomposition of \( R \) into 3NF. Prove it is a lossless-join decomposition. \hspace{1cm} (5)

---

6. **Transactions**

(a) Define the properties of *Atomicity* and *Isolation* in the context of a database transaction. What capability must a DBMS possess to support *Atomicity* and *Isolation*? \hspace{1cm} (4)

(b) What is *interleaving* and why is it supported by a DBMS? \hspace{1cm} (2)

(c) List three anomalies that can arise from interleaving. How does a DBMS ensure these anomalies cannot occur? \hspace{1cm} (4)
7. **Relational Algebra**

Consider the following schema

\begin{align*}
\text{SUPPLIERS} & (\text{sid:INT}, \text{sname:STRING}, \text{address:STRING}) \\
\text{PARTS} & (\text{pid:INT}, \text{pname:STRING}, \text{colour:STRING}) \\
\text{CATALOGUE} & (\text{sid:INT}, \text{pid:INT}, \text{price:DECIMAL[6,2]})
\end{align*}

Write the following queries in relational algebra:

(a) Find the *snames* of suppliers who supply every part.

(b) Find pairs of *sids* such that the supplier with the first *sid* charges more for some part that the supplier with the second *sid*.

(c) Describe in an English sentence the meaning of the relation resulting the following algebraic expressions.

\[
\rho(s, \pi_{\text{sid}} \text{ Suppliers}) \\
\rho(c, \pi_{\text{sid}} \text{ Catalogue}) \\
\quad s - (s - c)
\]

(d) Which relational algebraic operator is equivalent to \( s - (s - c) \)?