**Introductory Databases**

**Exercise 2: A Database Application for the Birmingham Bespoke Cracker company**

| Deadline | You must have submitted your code to Canvas for this exercise by 30\textsuperscript{th} November. After this date a penalty of 5 marks per day (or part of a day) will apply - except in cases where we have explicit permission to grant an extension from the welfare team. Students with extensions granted should also notify the course lecturer by email. Vivas will take place in the ground floor lab. In the last week of term (w/c 4\textsuperscript{th} December). Vivas will be scheduled by a timetable, which will be published on the module web page. The viva must demonstrate your code working on the lab machines and using the school DB server. |
| Marking scheme | This exercise is worth 10\% of your total mark for Introductory Databases. Your mark will be determined by the amount of the exercise you have completed and the quality of your solution. |
| Marking format | Marks will be awarded by viva and may be altered by supplementary tests, including plagiarism detection, which we perform on your electronic submission. You should also fill out a viva form before your viva so as not to delay the demonstration. The code that you demonstrate in your viva must be the code that you submitted electronically. |
| Printable sheets | Viva form is available here [PDF] |

**Introduction**

For this exercise you will set up and manipulate a simple data base to record and manipulate information for the Birmingham Bespoke Cracker company. The data base is a simplified system which does not capture all of the intricacies that would be necessary in a real system.

The exercise has three parts:

1. Completing the design of the database
2. Setting up and populating the data base
3. Creating an interface to the DB using Java's JDBC package. You will use this interface to create a database management program with a number of different functions.

**Part 1: Data Definition (20\%)**

The BBC manufactures bespoke Christmas crackers. Each cracker has an individual selling price. A cracker is made up of a (bad) joke, a hat and a gift and has an individual name and selling price. The quantity of each cracker sold also needs to be
recorded. Each joke has the text of the joke recorded and a royalty payment to be paid to the jokes author each time the joke is used. Every gift has a brief description and a cost price. Every hat has a brief description and a cost price.

An outline of the database design is:

- Cracker(cid, name, jid, gid, hid, saleprice, quantity)
- Joke(jid, joke, royalty)
- Gift(gid, description, price)
- Hat(hid, description, price)

The aim is to set up this database with some test data and to generate some example reports.

**Part 1.1 (20%)**
Write down the full definition of the database include domains for the attributes and any constraints that you identify. Do this as a set of SQL DDL statements.

**Part 2: Creating and populating the Data Base (30%)**
This part involves writing a Java program to set up and populate the database. You should do this using a separate java program which will set up a clean database for use during development and testing.

**Part 2.1 (10%)**
For the data base in part 1 you should set up the data base tables. Make sure that you include all the appropriate constraints. You’ll have to show the code that does this in the viva.

**Part 2.2 (20%)**
Populate the data base with a set of test data that is, at least, adequate to test and demonstrate the functionality in part 3.

As a guideline, you should have:
1. At least 1000 Crackers
2. At least 100 Jokes
3. At least 100 Hats
4. At least 100 Gifts
5. Sufficient realistic data to demonstrate the later part of the exercise. For instance, you should have at least 10 sensible entries for each category.

Note: You can do this by:
1. Explicitly creating a small amount of this data
2. Using loops to create the rest as synthetic data eg. Generating Jokes with the text “Joke <n>” and a randomly generated royalty fee.

*Show your data set to the demonstrator in the viva and be prepared to justify its adequacy.*

**Part 3: An interface using JDBC (50%)**
The BBC wants to test the database by generating example reports and being able to insert new data and check that the data base integrity is preserved.

Note:
1. This exercise is not about the UI. You can produce a sophisticated interface but it is also perfectly okay to have a very simple text based interface with no error checking (infact, see point 2 below)
2. You need to be able to show that your underlying database and java code will detect and correctly handle errors (e.g. selecting a jid that is not in the database). So, bear this in mind when you design the UI.

Part 3.1 (20%)
Produce a report for a cracker. Select the cracker by inputting the cracker’s id. Your report should list:
• The cracker id and name
• The description of the gift
• The Joke
• The description of the hat
• The unit saleprice
• The unit cost price
• The quantity sold
• The net profit

Part 3.2 (20%)
Produce a report for a joke. You should read the Joke’s id and use that to select the data. You should show:
• The Joke’s id and the joke
• The unit royalty payment
• The number of times the joke was used
• The total royalty payment due

Part 3.3 (10%)
Insert a new cracker into the data base.
You should read and insert:
• The cid,
• The cracker name,
• The jid,
• The gid,
• The hid,
• The saleprice
• And set quantity to zero
It’s important that you can demonstrate that if any constraints SHOULD be violated then these are detected BY THE DATABASE and handled by your Java code – so don’t restrict the input to legal values!

Marking scheme
The mark scheme is documented on the viva form. You should download, print out and fill in the form prior to asking a demonstrator to give you your viva.