Assignment 2 Notes

Wednesday, 28 November 2018 09:02

Tricky thing is to get anything to work!
He's tried to make it quite simple

Working from home:
Get VPN working
Make sure you have drivers installed
fully qualify the database name to "mod-intro-databases.cs.bham.ac.uk"

Viva
Mon-Fri slots for viva
Print viva sheet out and fill it in before viva
If you're slot doesn't work for you - find someone to swap with

Marks
Part doesn't work - no marks
Part works sort of - some of the marks
Part works well - all/most of the marks
Wonderful/clever stuff = bonus marks
Horrible/ugly code = knock marks off

Not asking for a fancy GUI:
1. Exercise is about using the DB not making a gui
2. Wants you to demonstrate that the database has been set up correctly so that the DB is catching any errors in the data that is being inputted
    - e.g entering a new party and you choose a menu that doesn't exist - if your GUI doesn't let you input invalid data then you can't demonstrate that your DB can handle the invalid data

Database design
- Not complete - will probably need a bunch of other things, but he's tried to keep it as simple as possible
- Can make this more fit for purpose by adding extra tables (really don't have to though)
- Set up 4 tables: party, venue, menu, entertainment
- Menu cost is unit price so per person
- Entertainment cost is not per person

Part 1.1

- Definition of the tables - domain, constaints
- Pid - some sort of integer
- Price - he thinks it should be an integer
- Constraints - range constraints e.g price should be positive, don't want negative number of guests, dates in the future, primary keys (pid, vid, mid, eid), foreign keys
- Money - do not represent money as floating point number! Absolute requirement that you use an integer representation otherwise gradually get rounding errors occurring
- Can make the primary keys auto-increment

Part 2.1

- Write code that will create a database
- It makes it easier to have code that will create a clean test database i.e a Java program that sets up a clean database that doesn't have any rubbish hanging around from your previous runs
- Start by dropping tables that already exist - you will get errors if those tables don't exist -> it doesn't matter, just catch the exceptions and print out that they don't exist
- Create tables completely afresh
- Everyone has their own database on the school's server - you cannot create a new database, but what you can do is create tables in that database

Part 2.2

- Create some realistic data
- Don't need to write 1000!!
- Create a for loop that:
    - randomly chooses a menu, venue, entertainment from a small selection
    - randomly generates a name and a cost
    - sticks it in the DB
- Make the parties last - do the menus, venues, entertainments first

Part 3.1

- Generate a report for the party
- Write a query that fetches the data, print it out appropriately and calculate the costs etc.
- Should be easy

Part 3.2

- Isomorphic to above
- Write a query that fetches the menu id, description and cost per item
    - how many times was it used? i.e how many parties was it used for and the total number of guests across all those parties

Part 3.3

- Important to be allowed to input invalid menu/venue
- DB should have been set up correctly so an exception is thrown, catch that exception and do something sensible with it

Extra Notes

- The reports should be sensible but don't waste time making it look beautiful! Format a little but just so it's readable
- Don't need to comment all the code
- Will deduct marks for horrible Java and SQL