

2 Databases (djg11)

Tattersalls sells horses, one at a time, by auction. A fresh rDBMS table is created every auction day, with a new record entered each time the hammer falls. The fields are: Date, Time, LotNumber, SellerID, SellerName, HorseName, HorseBreed, HammerPrice, BuyerName, BuyerID.

- (a) Draw a suitable E/R diagram for this data. Note that sellers and buyers are drawn from the same community of people and both pay a fixed-rate commission to the auction house based on the HammerPrice. Use underlining to denote primary keys and state any needs for synthetic keys or assumptions on recycling of values, such as LotNumber or HorseName. [4 marks]
- (b) An expert says the resulting table is ‘excessively denormalised’. What aspects might they be referring to and why might denormalised data be helpful in this situation? Give a normalised schema which uses several rDBMS tables, but that sticks to their preferred data handling approach of always creating at least one fresh table per auction day. [4 marks]
- (c) A report is needed with columns HorseBreed and TotalSales that totals the value of sales on a given day for each breed. Give suitable SQL that uses your normalised schema. [3 marks]
- (d) A second report of TotalSales per breed is needed, formatted as an rDBMS table that has one row for each auction date and one column for each breed of horse. For either of the above schemas, give two reasons why this might be hard to code using simple SQL (*i.e.* SQL that maps directly to the Relational Algebra) and explain what changes or ‘fudges’ are needed to achieve it within that subset. [4 marks]
- (e) The rDBMS being used supports many advanced features, such as ‘pivot’ and meta-programming operations that are, allegedly, elegant and easy-to-use. These could overcome the ‘fudges’ needed for part (d). How might one of these work? Would the most elegant solution be to use a better schema to start with? [5 marks]