Math 283

Homework #6

Below is a map of the database that this homework will involve.

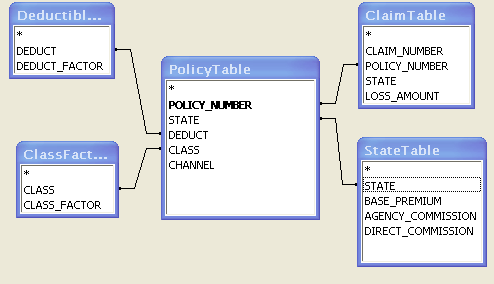
PolicyTable: A list of all the policies in the company

DeductibleFactors: A list of the possible deductibles, and the associated rating factor

ClassFactors: A list of the possible classes (age/sex combinations), and the factor

StateTable: Contains the base premium amount, as well as the commission for agency sales and direct sales.

ClaimTable: List of claims and amounts of claims. Only has records for policies that actually have a claims.



Problem 1:

1. (2 pts) Create a Query in access that returns a list of all the policy numbers in the states of IL, IN, OH, and MI. Save this query with name “Problem 1a”. In the criteria, use in (“IL”, “IN”, “OH”, “MI”)
2. (2 pts) Create a list of policies that returns a list of all the fields in the policy table that are for Males (first position of class code = “M”) and have a deductible of >=250. To get the first position of the class code, you can use the MID function in access, which has the same syntax as in Excel. Save this query with name “Problem 1b”.

Problem 2 (4 pts)

Create a query that calculates the rate for each policy. The rate is calculated by the following formula:

Rate = State Base Rate \* Deductible Factor \* Class Factor \* (1 + Commission).

Get the base rate from the state table.

Get the appropriate deductible factor and class factor from the respective table.

You should join these tables with the policy table. Since each possible value of state, class code, and deductible code are in the tables, you can use inner joins.

The commission depends on whether the policy was sold through an agent or through a direct channel. You can tell this on the policy record by using the CHANNEL column. Use an IIF statement to select the appropriate commission for a given policy.

(e.g. IIf([PolicyTable]![CHANNEL]="AGENT",[StateTable]![AGENCY\_COMMISSION],[StateTable]![DIRECT\_COMMISSION]))

You should output the following fields:

POLICY\_NUMBER

RATE

Sort by policy number in ascending order. Save query with name “Problem 2”.

Problem 3 (2 pts)

Create a list of all the policies, their rate, and the loss amount (if any).

You can do this by using your query from Problem 2 as an input to another query. Join “Problem 2” with the ClaimTable. Keep in mind that not all records in the Problem 2 output will have an associated claim. So you will need to do an outer join, retaining all the records in the Problem 2 output.

Save your query as ‘Problem 3’.