RISK ANALYSIS MANGEMENT

Hyundeok Park | October 22, 2018

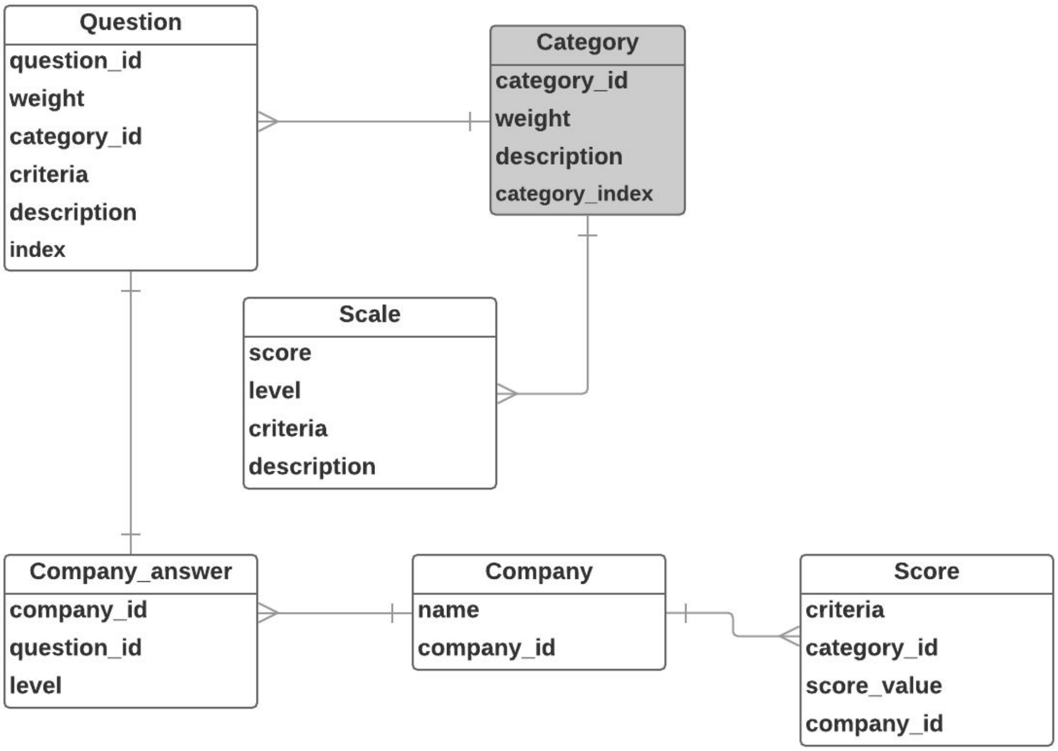

Table: Question

question_id	primary key	
index	int	#the order of a question e.g 1 ~ 75
description	varchar	
weight	float	#weight of a question in its own category
~~user~~	~~#dummy for now~~	
default		
category_id	foreign key	
criteria	char	#we are putting security and business trust in the same table(security/business)

Table: Category

category_id	primary key
category_index	int
weight	float
description	varchar

E.g

1) Mission and Security Requirements, Roles, Responsibilities and Policies [System Design]

Weight 0.0556

An 'overall' category?

A category has many questions

Table: Scale

description	varchar	
score	float	#user defined value e.g 0.0 ~ 0.95 in scale format definition
level	int	# 0 means insufficient, 6 means Not Applicable
criteria	char	#(S for security/B for business)

The purpose of this table is to distinguish between two answer scales. Both Business and Security have an answer scale of 0-6. However, on Business questions, "5" means "Very High Level of Trust". On Security questions, "5" is labelled "Corporate Optimization".

For a question + its answers to appear, join on the "criteria" column.

This is a reference table, not an object table.

Table: Company_answer

company_id	foreign key	#the company that's being assessed
question_id	foreign key	
level	int	#as in Scale table

Company 1	Question 1	5
Company 1	Question 2	4
Company 1		
Company 1		

Is there a way to avoid duplicating the "company 1" entry over and over? Or is it okay to duplicate it?

Let's say it's okay to duplicate it for now.

"How to store score?" Could be a question for the TA.

Table: Company

#companies that answer the above questions, to be improved

name	varchar
company_id	primary key

# not needed if Score table is included

security_score	float
trust_score	float
financial_score	float#(maybe not necessary)

# choose one to implement

Table: Category score #only categories

company_id
category_id
score

Is it better to have this separate kind of table? Or should we calculate scores on-the-fly and have that code in our views/controllers?

Alternative to the “Table: Category score” idea:

Table: Score

Criteria: Business or Security (because business, category will have different categories)

Category_id: “A”, “B”, “C”, “D”….or “Overall”, for the overall score (e.g. overall business, or overall security)
But, if category_id is a foreign key, then overall must also be a foreign key. So overall must then be a distinct category in the category table?

Score_value: The actual score value (float, valued from 0 to 1)

Company_id: Maybe?

The idea: to get a company’s business scores, join on company_id and filter by criteria “B”. This gives the join (the resulting view) the score for each category of questioning. The overall score is calculated from the individual category score (as a weighted average), and is also available to the join.