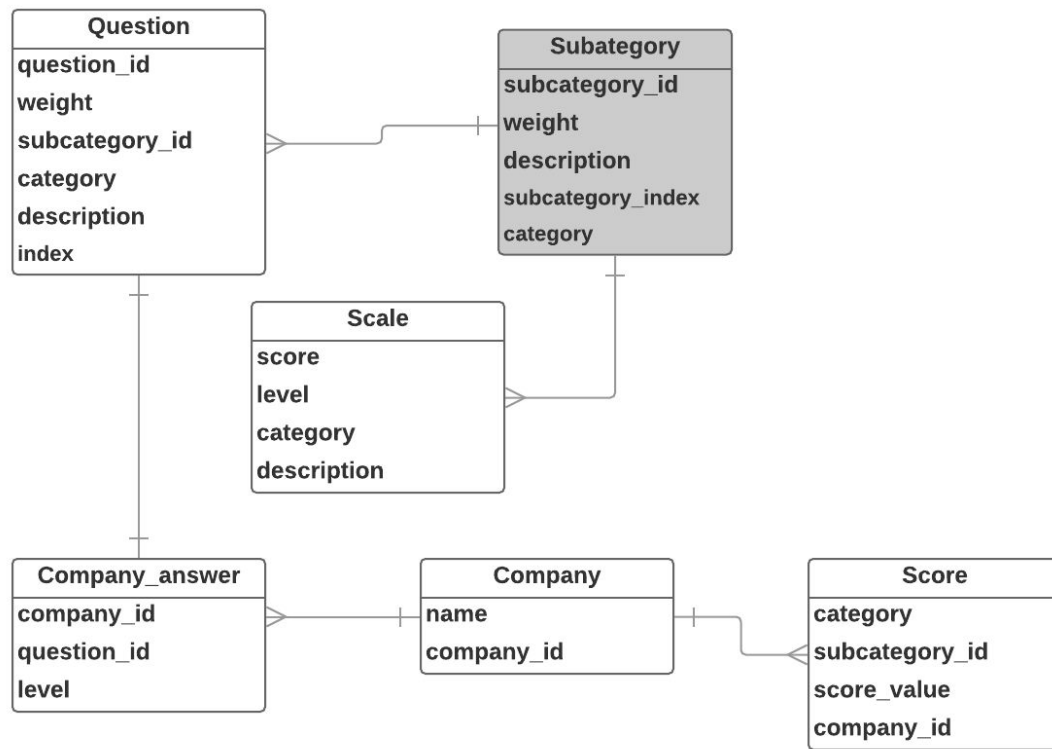


# RISK ANALYSIS MANGEMENT

Chengyi Min | November 4, 2018



## Table: Question

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

## Table: Subcategory

<b>subcategory_id</b>	primary key
<b>subcategory_index</b>	varchar #because the index can be a serial of letters
<b>weight</b>	float
<b>description</b>	varchar

**?category** char **#we need category to locate the subcategory**  
**of a question**

E.g

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

Weight 0.0556

An 'overall' subcategory?

A subcategory has many questions

## Table: Scale

<b>description</b>	varchar	
<b>score</b>	float	#user defined value e.g 0.0 ~ 0.95 in scale format
definition		
<b>level</b>	int	# 0 means insufficient, 6 means Not Applicable
<b>subcategory</b>	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 "category" column.

This is a reference table, not an object table.

## Table: Company\_answer

<b>company_id</b>	foreign key	#the company that's being assessed
<b>question_id</b>	foreign key	
<b>level</b>	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

<b>name</b>	varchar
<b>company_id</b>	primary key

*# not needed if Score table is included*

<b>security_score</b>	float
<b>trust_score</b>	float
<b>financial_score</b>	float#(maybe not necessary)

# choose one to implement

## **Table: subcategory score**

**#only subcategories**

company_id
subcategory_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: subcategory score" idea:

## **Table: Score**

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

subcategory\_id: "A", "B", "C", "D"....or "Overall", for the overall score (e.g. overall business, or overall security)

*But, if subcategory\_id is a foreign key, then overall must also be a foreign key. So overall must then be a distinct subcategory in the subcategory 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 category "B". This gives the join (the resulting view) the score for each subcategory of questioning. The overall score is calculated from the individual subcategory score (as a weighted average), and is also available to the join.