Data Science Exam

pwn274 - exam# 56

June 2022

1 Introduction

1.1 List the names of dishes ordered in meals in June 2022 that took place in restaurants in Denmark with more than 10 tables.

```
rest := \pi{rid}(\sigma_{num\_tables>10 \wedge country = Denmark}(restaurants))

meal := \pi{mid}(\sigma_{month = June \wedge year = 2022}(meals)) \bowtie rest

all_dishes := meal \bowtie meal_dishes

\pi_{dishes.name}(all_dishes \bowtie dishes)
```

1.2 For each restaurant in Denmark, list the restaurant name and the total price of all dishes ordered in meals in 2022

```
Unsolved
```

1.3 List the months and years in which no dishes named 'Olives' were ordered in restaurants in the country 'Greece'.

Having made a toy data-set, it mostly required some logic and trial and error to come up with the following:

```
SELECT meals.month, meals.year FROM
meals
JOIN meal_dishes ON meals.mid = meal_dishes.mid
```

```
JOIN restaurants ON restaurants.rid = meals.rid
JOIN dishes ON dishes.did = meal_dishes.did
WHERE restaurants.country = 'Greece' AND dishes.name <> 'Olives'
```

1.4 List the pairs of dish names such that the corresponding dishes were ordered together in at least one meal

While most likely not the prettiest of solutions, the following query grants me the correct entries on my toy data set:

```
SELECT a.name, b.name
FROM dishes AS a
CROSS JOIN dishes AS b ON a.did <> b.did
JOIN meal_dishes AS mda ON mda.did = a.did
JOIN meal_dishes AS mdb ON mdb.did = b.did
WHERE mda.mid = mdb.mid AND a.did > b.did
```

2 2

Unsolved

3 3

3.1 What are all the keys?

Since there is no combination of attributes are super keys without `experiment_date` or `machine`. But it turns out that these two form a minimal super key.

```
experiment_log(scientist, lab, institution, country, experiment_date,
        machine, result_path)

scientist → lab, institution, country
lab, country → institution
institution → country
experiment_date, machine → result_path
experiment_date, machine → scientist
```

As can quite easily be surmised, only the following two attributes are required to uniquely identify every

```
experiment_date, machine
```

Thus, due to minimality, {experiment_date, machine} is a super key and is the only required key of the relation:

3.2 Why is relation experiment_log not in BCNF

The schema is not in BCNF as multiple attributes are functionally dependent on a non-superkey.

3.3 Is the relation experiment_log in 3NF? Why or why not?

Since right hand sides of the functional dependencies purely consist of prime attributes, the relation is 3NF.

3.4 Decompose experiment_log into a set of relations in BCNF

With the book as my guide, I start out by decomposing to make the first functional dependency BCNF-compliant:

```
scientists(scientist, lab, institution, country)
experiments(scientist, experiment_date, machine, result_path)
```

Now I see that the forth and fifth functional dependencies are BCNF in 'experiments', allowing me to kill two birds with one stone and decompose to the following:

```
labs(lab, country, scientist)
labs(lab, country, institution)
experiments(scientist, experiment_date, machine, result_path)
```

Now all the relations are in BCNF.

3.5 Is the decomposition you have derived dependency-preserving?

Yes, I would argue that it is. It it clear to see that both of the `{experiment_date, machine}`-dependencies are preserved. The `scientist`-dependency is also preserved in combination with `institution→country`. So all in all, yes, the decomposition was dependency-preserving.