



Computational Thinking

Solutions to Exercise 7 (Data and Storage)

1 Journal Article Database

- a) Both $\{ID, TR-ID\}$ and $\{ID, title, TR-ID\}$ are superkeys, because they uniquely identify any row within the table. As there are two rows with the same ID and two rows with the same TR-ID, it is not sufficient to use only one of these two columns to identify a row. Hence, $\{ID, TR-ID\}$ is a candidate key. Thus, $\{ID, title, TR-ID\}$ is not a candidate key, because the `title` column can be omitted.

- b) Query 3. results in:

```
ERROR: function sum(text) does not exist
```

PostgreSQL does not accept strings as input to the SUM function. STRING_AGG would work to concatenate strings.

Query 6. results in:

```
ERROR: aggregate functions are not allowed in WHERE
```

SQL's WHERE clause does not work with aggregate functions like SUM, AVG, MAX, COUNT and so on. Instead, the HAVING keyword was introduced to SQL in order to quantitatively compare aggregated values. A correct query would look like this:

```
SELECT year, COUNT(*) FROM Articles GROUP BY year HAVING COUNT(*) > 10;
```

- c) 3 rows

2 Database Design

The following Entity Relationship Diagram describes the animals database. Owners and animals are in a 1-to- n relation. Each owner may own multiple animals, but every animal can have exactly one registered owner in the database. Animals and animal types are in a n -to-1 relation. Any animal cannot be both a cat and a dog, but the animal type table may very well contain multiple cats or dogs. Animals and breeds are in a n -to- n relation. Any animal can be a mixed breed and there may be multiple animals of the same breed in the database. Animals and allergies are in a n -to- n relation. Any animal may have multiple allergies and any allergy may afflict more than one animal in the database. For every animal allergy, we reserve a level field that denotes how strongly allergic the animal is to the allergy in question. Notice that we underline primary key attributes and we use an italic font to label unique attributes. The `id` field is chosen as the primary key for the animal table because even though the `chip_id` is a unique value, we would like to allow for the possibility that the unique `chip_id` is changed, for example, when the animal chip breaks or if it's updated due to a change in chip standard.

