Databases

Björn Þór Jónsson

April 20, 2022

Instructions

You have 4 hours to answer 6 problems described in the following. The exam consists of 10 numbered pages. Unless instructed otherwise your answers must be provided in the Canvas quiz *Final Exam April 2022*. Note that since the Canvas quiz has one qestion for each of questions 1–3, but sub-questions for questions 4–6, there are in total 13 questions in the quiz.

Database Description for Questions 1–3

In this exam you will work with a fictional (and poorly designed!) database of sales of produce (vegatables and fruits) to restaurants. To start working with the database, run the commands in `gag-april-2022.sql` found in the Canvas quiz using the PostgreSQL DBMS on your laptop. It is recommended to use `psql` for this purpose. The database has the following schema:

```
outlets (oid, zip)
humans (hid, hname, oid)
monitors (hid, mrating)

produce (pid, pname)
customers (cid, cname)

sells (sid, hid, cid, pid, sprice, swhen)
reviews (sid, hid, rgrade, rwhen)
```

Primary keys and foreign keys are defined and attributes are largely self-explanatory. You may study the DDL commands to understand the details of the tables (the CREATE TABLE statements are at the top of the script), consider the ER-diagram in Figure 1, or inspect the tables using SQL queries. Following are some additional notes that are important for your queries:

- The tables `sells` and `reviews` are created using option 2 for aggregation from the lectures.
- Two date values can be compared with the usal comparison operators ($=, <, >, \ldots$).
- If a sale is made by a human working in an outlet, we say that the sale is made from that outlet.
- If a sale is made to a customer, we also say that the customer has made a purchase.
- The data is randomly generated with some intentional data errors.

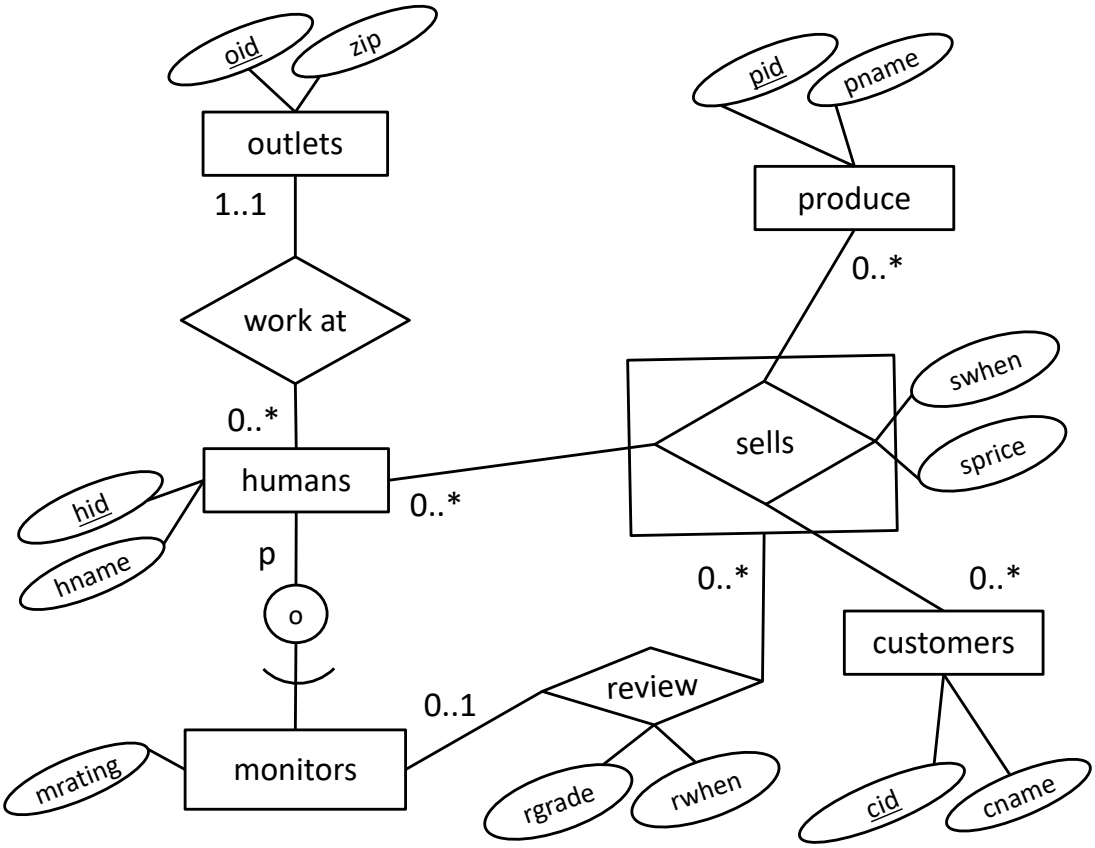

Figure 1: ER Diagram for the produce sales database.

Instructions for SQL Queries in Question 1

Queries must return correct results for any database instance. They should avoid system-specific features, including the LIMIT keyword. Queries should not return anything except the answer; a query that returns more information will not receive full points, even if the answer is part of the returned result. A sequence of several queries that answer the question will not receive full points, but subqueries and views can be used. Queries should be as simple as possible; queries that are unnecessarily complex may not get full marks, despite returning the correct answer. If you are unable to complete the query you can still submit your attempt, along with a brief description, and it may be given partial points.

1 SQL (40 points)

Answer each of the following questions using a single SQL query on the examination database. Submit one SQL file with all queries. The queries must be in the correct order and each query must be clearly labelled using a comment. If you use views, each view must be defined with the (first) query that uses the view. Queries must adhere to the detailed guidelines given on Page 3.

(a) There are 234 monitors who have received an mrating. How many monitors, who have received an mrating, work in an outlet with zip = 200?

(b) There is a total of 231 outlets in the database. How many of those outlets have no human working there?

(c) According to this database, how many of the sales have been reviewed before they were made?

(d) The highest price of any sale is 9920. Write a query to output the name(s) of all produce sold at that price.

Note: The output of this query may contain more than one row of data. Your query must work for such instances.

(e) 19 monitors have not received an mrating. How many humans have not received an mrating?

(f) The highest sale total of any human is 110373. What is the name of the human(s) that sold in total for this amount?

Note: The output of this query may contain more than one row of data. Your query must work for such instances.

(g) In all, 38 kinds of produce have been sold from **all** outlets with zip = 600 (meaning that each of those 38 kinds has been sold in all the outlets with zip = 600.) How many kinds of produce have been sold from **all** outlets with zip = 200?

Note: This is a division query; points will only be awarded if division is attempted.

(h) Write a query that outputs customer name, human name, sale price and sale date, for all sales from outlets with zip = 130, that are reviewed by the same human as the one that made the sale.

Note: The output of this query may contain more than one row of data. Each row must have four columns.

2 SQL programming (5 points)

Consider the SQL trigger code in Figure 2. The goal of the trigger is to prevent unacceptable reviews being entered into the `reviews` relation. To answer this question, you will need to study the database instance.

```
DROP TRIGGER IF EXISTS CheckReviews ON reviews;
DROP FUNCTION IF EXISTS CheckReviews();

CREATE FUNCTION CheckReviews() RETURNS TRIGGER
AS $$ BEGIN
  -- Check 1: Is the grade OK?
  IF ((NEW.rgrade < 0) OR (NEW.rgrade > 10)) THEN
    RAISE EXCEPTION 'Grade is out of bounds'
    USING ERRCODE = '45000';
  END IF;
  -- Check 2: Did the monitor review their own sale?
  IF EXISTS (
      SELECT *
      FROM sells S
      WHERE S.hid = NEW.hid AND S.sid = NEW.sid) THEN
    RAISE EXCEPTION 'Monitor cannot review own sales'
    USING ERRCODE = '45000';
  END IF;
  RETURN NEW;
END; $$ LANGUAGE plpgsql;

CREATE TRIGGER CheckReviews
AFTER INSERT ON reviews
FOR EACH ROW EXECUTE PROCEDURE CheckReviews();

INSERT INTO vetters(hid) VALUES (720);
INSERT INTO reviews(sid, hid, rgrade, rwhen) VALUES (1, 720, -2, CURRENT_DATE);
INSERT INTO reviews(sid, hid, rgrade, rwhen) VALUES (2, 720, 2, CURRENT_DATE);
```

Figure 2: Insertion trigger `CheckReviews` for the `reviews` relation.

Select the true statements, given the current database instance:

(a) The first check (on `rgrade`) could be implemented as a CHECK constraint.
(b) The trigger could also be implemented as a BEFORE trigger.
(c) The first INSERT statement for `reviews` will succeed.
(d) The second INSERT statement for `reviews` will succeed.

3 Relational Algebra and Calculus (5 points)

Write the following SQL queries in both relational algebra and tuple relational calculus.

(a)
```
SELECT O.oid
FROM outlets O
WHERE O.zip = 200;
```

(b)
```
SELECT hname
FROM humans H
    JOIN monitors M ON H.hid = M.hid
WHERE M.mrating > 8;
```

4 ER Diagrams and Normalization (30 points)

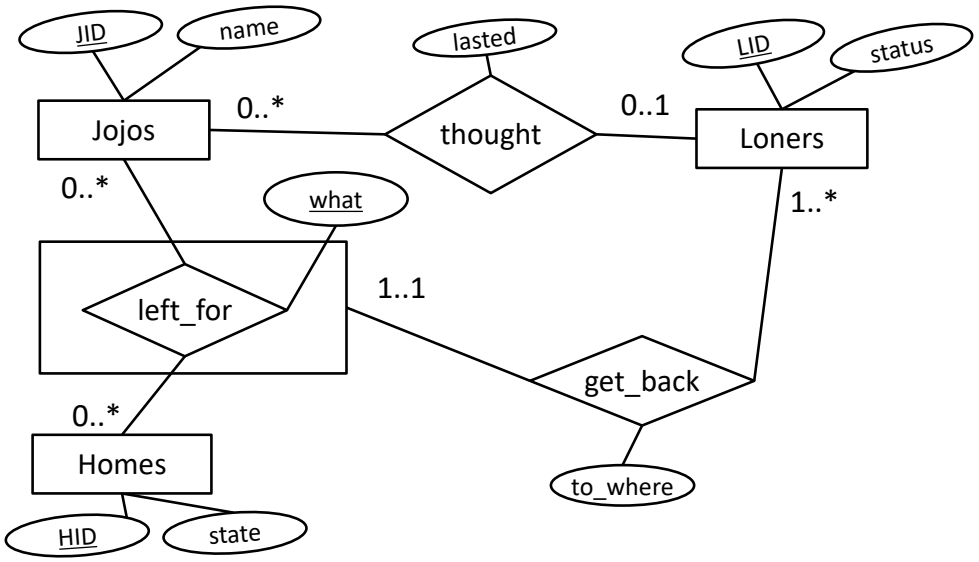

Figure 3: ER Diagram for a database of Jojos.

a) The ER diagram in Figure 3 shows a database for Jojos who thought they were Loners. (If you are wondering what a Jojo is, the ER diagram is very loosely based on lyrics of the Beatles' song Get Back. It is recommended to not worry about what a Jojo is, however, and focus instead on the relationships and their participation constraints.) Select the true statements. You should base your answers **only** on the ER diagram:

(a) All Jojos connect to at least one Loner through some relationships.

(b) All Loners connect to at least one Home through some relationships.

(c) All Homes connect to at least one Loner through some relationships.

(d) Some Loner may be connected to itself through some relationships.

(e) Some Home may be connected to itself through some relationships.

(f) All Jojos must be connected to themselves through some relationships.

b) Write SQL DDL commands to create a database based on the ER diagram in Figure 3. The DDL script *must run* in PostgreSQL as a whole. The relations must include all primary key and foreign key constraints. Constraints that cannot be enforced with standard primary key and foreign key constraints should be omitted. The type of key attributes should be INT, all other attributes may be of type VARCHAR.

c) Write an ER diagram for a database of thesis writing based on the following requirements. The diagram should clearly show the entities, attributes, relationships and participation constrains described below. Use the notation presented in the textbook and lectures. Attributes are only important if mentioned. If you need to make additional assumptions put them in the box below.

- Faculty members supervise students. Each student has a least 1 and at most 3 faculty members as supervisors.
- Each student writes one thesis. Students may collaborate on writing a thesis.
- Each thesis has exactly one topic.
- Faculty members can validate theses.
- Faculty members may also evaluate the supervision of students by other faculty members.

d) Consider a table $R(L, M, N, O, P)$ with the following dependencies:

$$\begin{aligned} MN &\rightarrow LOP \\ L &\rightarrow M \\ O &\rightarrow P \end{aligned}$$

Select the true statements:

(a) MN is the only (candidate) key of R.

(b) $NO \rightarrow N$ is a trivial functional dependency.

(c) Normalizing to 3NF or BCNF results in exactly two relations.

(d) The relation can be normalized to BCNF without losing dependencies.

e) Consider a table $R(L, M, N, O, P)$ with the following dependencies:

$$\begin{aligned} L &\rightarrow MNOP \\ M &\rightarrow L \\ N &\rightarrow O \\ O &\rightarrow P \end{aligned}$$

Normalize R to the highest possible normal form (3NF or BCNF), based on functional dependencies, while allowing all functional dependencies (excluding trivial, unavoidable, and redundant dependencies) to be checked within a single relation. For each resulting relation, write its columns and clearly indicate whether it is in BCNF.

5 Index Selection (10 points)

Consider the following large relation with information on canals:

Streets(id, lat, lon, length, <many long attributes>)

For each of the queries below, select the index that a good query optimiser is most likely to use for the Streets table to process the query. Assume that all indexes are *unclustered* B+-trees. Also, assume that (a) the relation has street data for a large country (meaning billions of streets); (b) the four given attributes are integer values, and none of these are nullable; and (c) the lat, lon and length attributes all range uniformly from 0 to 100000. For each case, select the best index for the query, or select "no index" if a full table scan would yield better performance than any of the possible indexes.

(a) Street(id)

(b) Street(lat)

(c) Street(lat, lon)

(d) Street(length, lat, lon)

(e) No index

Query 1

```
select count(*)
from Streets
where lat < 5000;
```

Query 2

```
select *
from Streets
where lat < 5000;
```

Query 3

```
select avg(length)
from Canals
where x = 20000 and y = 20000
```

6 Architecture, NoSQL and Big Data (10 points)

a) Select the correct statements below:

(a) A well-designed relational database can never have incorrect data.

(b) Social Value can be a significant reason for working with big data.

(c) SSDs are much faster than HDDs, especially for small random operations.

(d) Key-value stores a very good at joining large relations.

b) Discuss the primary disk access pattern of big data applications, and why traditional HDDs may be relatively useful for big data applications after all.