

Databases

TDA357/DIT621

Exercise 1

SQL

1 History

(Adapted from a previous exam question)

A database containing historical periods and historical events has the following schema:

Periods(pname, started, ended)

Events(ename, year)

Periods have a starting and ending year (both are inclusive) e.g. (“World war II”, 1939, 1945) could be an entry. Events have a year when they occurred e.g. (“SQL became an ISO standard”, 1987) could be an entry.

- a) Define SQL tables for the relational schema above. Make sure to add reasonable constraints to your data.
- b) Write a query that finds the names of all events that occurred during any of the same historical periods as “The Great Collapsing Hrungr Disaster” (a fictional event that you may assume is in the Events table). Make sure each such event occurs only once in the result.

To clarify: “The Great Collapsing Hrungr Disaster” happened in some year, and that year is during some number of historical periods. Your job is to find all events that occurred during all those periods.

Hint: First write a query for finding all periods the event is in, and then use it to find all events in those periods.

- c) Write a query that finds the name of the most eventful historical period(s). In other words, the period with the greatest number of events in it. May be more than one period only if there are several periods with the same number of events.

2 Online sales platform

(Adapted from a previous exam question.)

Consider this schema for an online sales platform:

Items (itemname, price)
 Categories (catname)
 Categorized(item, category)
 category → Categories.catname
 item → Items.itemname
 Discounts (category, pricefactor)
 category → Categories.catname

Items are things that can be sold. Each item has a base price. Some items belong to a category, and some categories have an active discount. The attribute pricefactor specifies a discount as a factor, e.g. a pricefactor of 0.75 means all items in the specified category are discounted by 25%.

Write an SQL query for solving each of these tasks:

- a) Define SQL tables for the relational schema above. Make sure to add reasonable constraints to your data.
- b) Write a query that finds the name and actual price of each item, factoring in discounts where applicable.
- c) Write a query that finds the largest difference in base price (ignoring discounts) between any two products in the same category. The result should be a single number. Items that have no category are irrelevant to this query.
- d) Write a query that finds the average price of all products that do not have a category. The result should be a single number.

3 Veterinary clinic admin system

(Previous exam question.)

A veterinary clinic has the following information about their clients, the pets of the clients and the veterinarians that work there. The clinic only takes care of dogs, cats or rabbits.

Clients (name, email, phonenr)
 Pets (chipnr, name, owner, type, born)
 owner → Client.name
 type ∈ {dog, cat, rabbit}
 Vets (id, name, phonenr, specialisation)
 specialisation ∈ {dog, cat, rabbit}

where chipnr and id are numbers that identify the pets and the employees respectively, and born is the year on which the pet was born. Assume name is enough to identify a client.

- a) Define SQL tables for the relational schema above.

Make sure to add reasonable constraints to your data, for example no pet should have been born in the future! :-)

- b) Define a table to keep track of the bookings. The table needs to contain information about the veterinarian that is booked, which animal the consultation is about, which day and time the consultation will take place, and the length of the consultation which can be 30 or 60 minutes long, with 30 minutes as default. Consultations can only be booked at 9, 10, 11, 13, 14 or 15 hours.

Observe that it might be possible that the same pet has 2 appointments the same day (on different hours) but of course, neither pets nor veterinarians can have two different appointments at the same day and time.

- c) Write an SQL query that gives the total amount of money that needs to be billed to each costumer for all the bookings taking place this month. Every 30 min of consultation costs 500 kr. Each customer that needs to be billed should only appear once and those with no bill this month should not be part of the result.

Note: In order to obtain the current year one can use the following PostgreSQL expression: `date_part('year', CURRENT_DATE)`. Similar with month.

4 Tenant association

(Adapted from questions in a previous exam.)

The company *StairwayToHeaven* has been requested to help with the booking system of the tenant's association *ParadiseOnEarth*. For this purpose they have created a database with the following relational schema. Notice that the schema is not complete since for example it is missing, among other things, some information on keys!

Tenants (personal_nr, name, tel)

Building (tenant, apt_nr)

tenant → Tenants.personal_nr

BookingPrices (facility, price)

Bookings (facility, day, time, apt_nr)

facility → BookingPrices.facility

The needed information about the tenants is kept in Tenants.

The building has 100 apartments with numbers 1 to 100. Who rents which apartment is stored in Building. Observe that more than one person can formally be registered as tentants of a particular apartment; also a person can be renting/be the tenant of more than one apartment.

BookingPrices keeps the information of how much it currently costs to book any of the different facilities. The available facilities are 2 different washing rooms (with numbers 1 and 2) that are booked separately, a room for parties and a sauna.

Bookings states which apartment number has booked a particular facility on a particular day and time. The party room can only be booked for the whole day and then the time is set to 8. The other facilities can be boked for 3 hours at times 8, 11, 14, 17 or 20. No apartment can book the same facility more than once per day but can book different facilities (including both washing rooms) on the same day.

- a) Define SQL tables for the relational schema above. Make sure to define all reasonable constraints, including primary and foreign keys, and unique constraints (if any).

- b) When a person misbehaves in one of the facilities (for example, makes too much noise when having a party or does not clean after using the sauna), the person is added to a list of *banned* people, which means that none of the apartments the person is tenant of can place a booking for that particular facility during the following 90 days after the ban has been issued, even if the actual day the person would like to use the facility is far away in the future (beyond 90 days in time).

Create an SQL table that keeps the information of who has been banned for booking which facility. Even past bans are to be kept so that the steering group of the association can see how many bans were issued for a particular tenant.

Make sure to add all reasonable constraints, including primary and foreign keys, and unique constraints (if any).

- c) Create an SQL query that lists the top 10 bookings this year. To identify the top bookings you need to count how many times each apartment has booked each of the facilities and then select the top 10.

Suppose for example that apartment 1 has booked the sauna 5 times and the washing room 1 3 times, and that apartment 2 has booked the washing room 1 4 times. Then the very top booking is that of the sauna by apartment 1 (5 times), followed by the booking of washing room 1 by apartment 2 (4 times), followed by the booking of washing room 1 by apartment 1 (3 times).

The output should contain the apartment number, the name of all the tenants in the apartment, which facility the top booking is about and the total number of times the facility was booked by that apartment.

- d) Write an SQL view that lists the name of all the tenants with their apartment numbers who have a particular booking today (`CURRENT_DATE`). The output should also contain which facility has been booked and at what time. Order the output first by the facility and then by the (ascending) time.
- e) Bills for an apartment (for example the rent or the cost of all the monthly bookings) are equally distributed among all the tenants of that particular apartment. So, if apartment number 1 has 2 tenants then each of them pays half of the bills associated to the apartment.

Write an SQL query that computes the percentage that needs to be charged to each of the tenants of each apartment. In the example of apartment number 1 above, the computed percentage should be 50. Order the output in a descendent order by the percentage.

- f) You should now help *StairwayToHeaven* to keep track of the monthly fee to be charged to each tenant for the use (booking) of the facilities this month. If a person is a tenant in more than one apartment, then the person will only get one fee which is the sum of all the fees he/she has to pay in each of the apartment he/she is a tenant of.

The output should contain the personal number of the tenant and the amount to pay.

Note: The expression `date_part('month', CURRENT_DATE)` gives the number of the current month.