Database Systems, CSCI 4380-01
Homework # 1 Answers
Due Thursday February 7, 2013 at 2 pm

Homework Statement. This homework aim to teach you how to construct complex queries using relational algebra. Please do the parts in sequence, as they build on each other. Suppose you are given the following database that tracks the activities of the Mars Curiosity Rover :

```
RoverLocations( locationId, name, arrivalDT, south, east, departureDT )
Distances( locationid1, locationid2, distance)
TestTypes ( testTypeID, name, instrumentName, description )
Tests ( testDT, testTypeID, parameters )
Chemistry ( chemId, name, description )
TestResults ( testDT, testTypeID, chemId, amount, unit )
```

Note: All datetime fields are indicated as `DT` and combine both the date and the time, e.g. a valid value would look like `datetime '01-31-2013 08:00:00'`. You can assume that you can check if a date time `X` comes after another date time `Y` by checking whether `X > Y`.

The relation `RoverLocations` tracks the location of the rover (in terms of south and east degrees). Each time the rover stops, its location has a new identifier. When the rover departs from a location, the departure date and time is filled in. The attribute `name` is the name given to a specific location if such a name exists. For the rover's current location, you can assume the departureDT has a special value `Null` which is compatible with the datetime data type.

The relation `Distances` tracks the actual distance between two locations. Technically we can use this to compute speed for rover, but we will not do this at this time. Distance information is stored only for a couple of locations that the rover visited in sequence, meaning that the rover visited `locationid2` right after `locationid1`.

The rover carries multiple instruments that can run many different types of tests. Each test type is given an identifier and the detailed information for these types are stored in the `TestTypes` relation.

The relation `Tests` records the many different types of tests that the rover runs. Each test is done when the rover is stopped at a specific location. The rover cannot run tests when moving. The tests may have many parameters. We assume they are stored as a text field for now.

Generally, the tests search for different chemical species (for example specific carbon isotopes, or elements, minerals, etc.) Each species is given a different identifier and stored in the relation `Chemistry`.

Finally, after each test, if certain chemical species are identified, we this store in relation `TestResults`. The test results include which species are identified in a specific test and the amount measured together with the unit.

You can see a visual explanation of the data model in the last page.

Write the following queries using relational algebra:

Part 1. The following queries only need a single SELECT (σ), followed by a PROJECT (π):

(a) Find and return the dateTime of all tests that found at least 0.5 parts per million (i.e. unit of `ppm`) of the chemistry with id 100.

Answer.

$$Answer = \pi_{testDT}(\sigma_{chemid=100 \text{ and } unit=ppm} TestResults)$$

(b) Find the location id, south and east coordinates of all locations that the rover visited before `datetime '11-01-2012 08:00:00'`.

Answer.

$$Answer = \pi_{locationID,south,east}(\sigma_{arrivalDT<datetime'11-01-201208:00:00'} RoverLocations)$$

(c) Find all test type identifiers conducted on the instrument `RAD` (short for Radiation Assessment Detector).

Answer.

$$Answer = \pi_{testTypeID}(\sigma_{instrumentName=RAD} TestTypes)$$

Part 2. The following queries combine SELECT (σ), SET operations ($\cap, \cup, \setminus$), PROJECTION (π) and RENAMING (ρ) as necessary:

(a) Find all test identifiers of test that are conducted on the instrument `DAN` (short for Dynamic Albedo of Neutrons) after `datetime '1-01-2013 08:00:00'` and have found the chemistry with identifier `123`.

Answer.

$$\begin{aligned} T1 &= \pi_{testTypeID}(\sigma_{chemID=123 \text{ and } testDB>datetime'1-01-201308:00:00'} TestResults) \\ T2 &= \pi_{testTypeID}(\sigma_{instrumentName=DAN} Tests) \\ Answer &= T1 \cap T2 \end{aligned}$$

(b) Find all chemistry that the have not been found in any tests for more than 0.2 parts per million (e.g. unit `ppm`). (Note: we are disregarding tests using different units and making no conversions.)

Answer.

$$\begin{aligned} T1 &= \pi_{chemId}(\sigma_{amount>0.2 \text{ and } unit=ppm} TestsResults) \\ T2 &= \pi_{chemId} Chemistry \\ Answer &= T2 - T1 \end{aligned}$$

(c) Find all tests that have either found one of the chemical species with identifier 101, 102 or 155 with at least 0.2 parts per million (e.g. unit `ppm`), or found both chemical species 104 and 105 both with at least 0.4 parts per million (e.g. unit `ppm`).

Answer.

$$\begin{aligned}
T1a &= \sigma_{(chemId=101 \text{ or } chemId=102 \text{ or } chemId=155) \text{ and } amount=0.2 \text{ and } unit=ppm} TestResults \\
T1b &= \pi_{testDT,testTypeID}(T1a) \\
T2 &= \pi_{testDT,testTypeID}(\sigma_{chemId=104 \text{ and } amount=0.4 \text{ and } unit=ppm} TestResults) \\
T3 &= \pi_{testDT,testTypeID}(\sigma_{chemId=105 \text{ and } amount=0.4 \text{ and } unit=ppm} TestResults) \\
T4 &= T2 \cap T3 \\
Answer &= T1 \cup T4
\end{aligned}$$

Part 3. The following queries combine SELECT (σ) statements with JOIN ($\bowtie$) only, followed by a PROJECT (π):

(a) Find the name of all chemical species that are found by a test of the instrument `RAD`.

Answer.

$$Answer = \pi_{name}(Chemistry \bowtie (TestResults \bowtie (\pi_{testTypeId}(\sigma_{instrumentName=RAD} TestTypes))))$$

Note: I am projecting out attributes in `TestTypes` to remove `TestsTypes.name` attribute. Otherwise, the resulting relation will have two attributes called `name` and the last projection would be ambiguous and incorrect.

(b) Find the identifier of all tests conducted at the location with coordinates `south=4.5`, and `east=137.4`.

Answer.

$$\begin{aligned}
T1 &= \sigma_{south=4.5 \text{ and } east=137.4} RoverLocations \\
T2 &= TestResults \bowtie_{testDT>=arrivalDT \text{ and } (testDT<=departureDT \text{ or } departureDT=Null)} T1 \\
Answer &= \pi_{testDT,testTypeID} T2
\end{aligned}$$

Part 4. Freeform, you decide which combination is needed.

(a) **Bonus/Optional.** Find two locations `L1,L2` that the rover visited in sequence and that in both locations, it has found the chemical species with name `methane`. Return the id of the locations.

Answer.

$$\begin{aligned}
T1 &= \pi_{chemId}(\sigma_{name=methane} Chemistry) \\
T2 &= \pi_{testDT}(TestResults \bowtie T1) \\
T3a &= T2 \bowtie_{testDT>=arrivalDT \text{ and } (testDT<departureDT or departureDT=Null)} RoverLocations \\
T3b &= \pi_{testDT,locationId}(T3a) \\
T4 &= \pi_{locationId1,locationId2}(T3b \bowtie_{locationID=locationID1} Distances) \\
Answer &= \pi_{locationId1,locationId2}(T4 \bowtie_{locationID2=locationID} T3)
\end{aligned}$$

(b) Find the name of all the test types that have not been performed before date time `datetime '1-1-2013 00:00:00'`.

Answer.

$$
\begin{aligned}
T1 &= \pi_{testTypeID}(\sigma_{testDT < datetime'1-1-2013 00:00:00'} TestResults) \\
T2 &= (\pi_{testTypeID} TestTypes) - T1 \\
Answer &= \pi_{name}(T2 \bowtie TestTypes)
\end{aligned}
$$