Name	CWID

Midterm Exam

October 24th, 2016
1:50-3:05

CS425 - Database Organization
Results

Please leave this empty!

1.1		1.2		1.3		1.4		Sum	

Instructions

- Try to answer all the questions using what you have learned in class. Keep hard questions until the end.
- **When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!**
- The exam is closed book and closed notes!
- **For relational algebra questions assume set semantics!**

Consider the following database schema and example instance for a car database:

person

name	age	state
Peter	32	IL
Alice	57	CA
Bob	17	NY

car

license	color	productionYear	model
A54-FGY	green	2011	Porsche 510
F55-JRK	red	2010	Impala
A33-IKF	red	1980	Golf

owns

owner	car	ownsSince
Peter	A54-FGY	2015
Peter	F55-JRK	2010
Alice	A33-IKF	1990

model

mId	brand	weight
Porsche 510	Porsche	1300
Porsche 310	Porsche	1400
Impala	GM	3000
Golf	Volkswagen	2500

Hints:

- Attributes with black background form the primary key of a relation (.e.g, `name` for relation `person`)
- The attribute `car` of relation `owns` is a foreign key to `license` of relation `car`.
- The attribute `owner` of relation `owns` is a foreign key to `name` of relation `person`.
- The attribute `model` of relation `car` is a foreign key to `mId` of relation `model`.
- All foreign keys have been created with the **CASCADE** option.

Part 1.1 Relational Algebra (Total: 28 Points)

Question 1.1.1 (8 Points)

Write a **relational algebra** expression that returns the *model* and *productionYear* of all **red** *cars*.

Solution

$$\pi_{mode,productionYear}(\sigma_{color=red}(\text{ car }))$$

Question 1.1.2 (10 Points)

Write a **relational algebra** expression that returns the *average weight* of *car models* per *brand*.

Solution

$${}_{avg(weight)}\mathcal{G}_{brand}(\text{ model }))$$

Question 1.1.3 (10 Points)

Write a **relational algebra** expression that returns the *name* of persons that own *green cars* produced by *Volkswagen* (the *brand* of the car).

Solution

$$\pi_{name}(\sigma_{color=green}(\text{ car }) \bowtie_{license=car} \text{ owns } \bowtie_{owner=name} \text{ person } \bowtie_{model=mId} \sigma_{brand=Volkswagen}(model))$$

Part 1.2 SQL - DDL (Total: 16 Points)

Question 1.2.1 (16 Points)

Write an **SQL DDL statement** that creates a new relation `driversLicense` that records information about drivers licenses of persons. This relation should have attributes `person`, `licenceNr`, `state`, `issueDate`, and `status`. A drivers license is uniquely identified by the combination of its `licenseNr` and `state`. Attribute `person` is a foreign key to relation `person`. Note that the `licenseNr` is an alphanumeric value that is precisely 11 characters long. Attribute `status` is a single character attribute that can take either of the following two values: `A` (active) or `S` (suspended). Attribute *state* is a 2 character state code (e.g., `IL` for *Illinois*).

Solution

```
CREATE TABLE driversLicense (
    person VARCHAR(30),
    licenseNr CHAR(11),
    state CHAR(2),
    issueDate DATE,
    status CHAR(1),
    PRIMARY KEY (licenseNr, state),
    FOREIGN KEY (person) REFERENCES person,
    CHECK (status = 'A' OR status = 'S')
);
```

Part 1.3 SQL - Queries (Total: 36 Points)

Question 1.3.1 (10 Points)

Write an **SQL query** that returns the license plate number (attribute `license`) and color of cars owned by persons that are less than 18 years old.

Solution

```
SELECT license, color
FROM car c, owns o, person p
WHERE c.license = o.car AND o.owner = p.name AND age < 18;
```

alternatively explicit joins of course.

Question 1.3.2 (11 Points)

Write an **SQL query** that returns the *number of cars* owned per `state` (the person's state).

Solution

```
SELECT count(*) AS numCars, state
FROM person p, owns o
WHERE p.name = o.owner
GROUP BY state;
```

Question 1.3.3 (15 Points)

Write an **SQL query** that returns **states without** any Porsche cars. That is, a state should be returned if no person in that state owns a car with *brand* Porsche.

Solution

```
WITH
    porscheState AS (SELECT DISTINCT state
                     FROM person p, car c, owns o, model m
                     WHERE p.name = o.owner
                           AND o.car = c.license
                           AND c.model = m.mId
                           AND m.brand = 'Porsche')

SELECT DISTINCT state
FROM person p
WHERE state NOT IN (SELECT * FROM porscheState);
```

alternatives include aggregation (count is 0) and set difference

Part 1.4 SQL - Updates (Total: 20 Points)

Question 1.4.1 (8 Points)

Write an **SQL statement** that inserts a new car model into the database with mId Hummer, brand US, and weight 10,000.

Solution

```
INSERT INTO model
VALUES ('Hummer', 'US', 10000);
```

Question 1.4.2 (12 Points)

Write an **SQL statement** that updates the brand of all car models to GM if their current brand is Volkswagen and their weight is less than 2000.

Solution

```
UPDATE model SET brand = 'GM' WHERE brand = 'Volkswagen' AND weight < 2000;
```