

Name

CWID

Midterm Exam

October 28st, 2014

10:00-11:00

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 crime report database:

incident

crime	victim	delinquent	location	year
H-1	123	444	60615	2014
T-1	333	444	60615	2013
L-1	222	555	15544	2013

delinquent

dSSN	dName	hasCriminalRecord
444	Gertrud	yes
555	Heinz	no

victim

vSSN	vName	age	livesAtZip
123	Peter	53	60615
222	Alice	20	60616
333	Bob	65	60615

location

zip	city
60653	Chicago
60616	Chicago
60602	Chicago
15544	Buffalo

crime

code	description	category	prisonMonths	fine
H-1	homicide	Violent Crimes	600	NULL
H-9	multiple homicide	Violent Crimes	900	NULL
T-1	pickpocket	Theft	12	1000
T-3	theft with weapon	Theft	36	5000
T-9	shoplifting	Theft	3	300
L-1	money laundering	White-collar crimes	NULL	10000

Hints:

- Attributes with black background form the primary key of a relation (.e.g, *vSSN* for relation *victim*)
- The attribute *crime* of relation *incident* is a foreign key to *code* of relation *crime*.
- The attribute *victim* of relation *incident* is a foreign key to *vSSN* of relation *victim*.
- The attribute *delinquent* of relation *incident* is a foreign key to *dSSN* of relation *delinquent*.
- The attribute *location* of relation *incident* is a foreign key to *zip* of relation *location*.
- 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 name of all victims living at zip code 60615.

Solution

$$\pi_{vName}(\sigma_{livesAtZip=60615}(\text{victim}))$$

Question 1.1.2 (10 Points)

Write a **relational algebra** expression that returns the victim name, delinquent name, and crime code for all incidents in 2014.

Solution

$$\pi_{vName,dName,crime}(\text{incident} \bowtie_{victim=vSSN} \text{victim} \bowtie_{delinquent=dSSN} \text{delinquent})$$

Question 1.1.3 (10 Points)

Write a **relational algebra** expression that returns the number of incidents per zip code (attribute *location*).

Solution

$$\text{count}(*)\mathcal{G}_{\text{location}}(\text{incident})$$

Part 1.2 SQL - DDL (Total: 16 Points)

Question 1.2.1 (16 Points)

Write an **SQL statement** that creates a new relation *sentence* that records prison sentences and/or fines for delinquents. This relation should have attributes *delinquent*, *crime*, a *date*, the *monthInPrison*, and the *fine*. Note that not all sentences include a fine and not all sentences include months in prison. However, each sentence has either a fine or months in prison. The delinquent, date, and crime uniquely identify a sentence. Both months in prison and fines are positive integer numbers (tip: take into account that some sentences do not have a fine or months in prison). Attribute *delinquent* should be a foreign key to relation *delinquent*.

Solution

```
CREATE TABLE sentence (  
    delinquent VARCHAR(9),  
    crime VARCHAR(3),  
    d DATE,  
    monthInPrison NUMERIC(8),  
    fine NUMERIC(12),  
    PRIMARY KEY (delinquent, crime, d),  
    FOREIGN KEY (delinquent) REFERENCES delinquent,  
    CHECK (monthInPrison IS NOT NULL OR fine IS NOT NULL),  
    CHECK ((monthInPrison IS NULL) OR (monthInPrison > 0)),  
    CHECK ((fine IS NULL) OR (fine > 0))  
);
```

Part 1.3 SQL - Queries (Total: 36 Points)

Question 1.3.1 (8 Points)

Write an **SQL query** that returns the code and description of all types of violent crimes (attribute *category*) that result in more than 36 months of prison.

Solution

```
SELECT code, description
FROM crime
WHERE category = 'Violent_Crimes'
      AND prisonMonths > 36;
```

Question 1.3.2 (13 Points)

Write an **SQL query** that returns the names of all homicide victims (crime code H-1 or H-9) in Chicago in 2012.

Solution

```
SELECT vName
FROM incident i,
      victim v,
      location l
WHERE i.victim = v.vSSN
      AND i.location = l.zip
      AND year = 2012
      AND city = 'Chicago'
      AND (crime = 'H-1' OR crime = 'H-9');
```

alternatively explicit joins of course.

Question 1.3.3 (15 Points)

Write an **SQL query** that returns the city with the most thefts (crimes of type "Theft").

Solution

```
WITH
  cityT AS (SELECT city, COUNT(*) AS numT
            FROM location l, crime c, incident i
            WHERE i.crime = c.code
                  AND i.location = l.zip
                  AND i.category = 'Theft'
            GROUP BY city)

SELECT city
FROM cityT t
WHERE t.numT = (SELECT max(numT) FROM cityT);
```

Part 1.4 SQL - Updates (Total: 20 Points)

Question 1.4.1 (8 Points)

Write an **SQL statement** that inserts a new victim into the database with SSN 111, name Fritz, age 55, living at zip code 65555.

Solution

```
INSERT INTO victim
VALUES (111, 'Fritz', 55, 65555);
```

Question 1.4.2 (12 Points)

Write an **SQL statement** that sets the fine for all crimes to 500 if the crime currently does not have a fine or the current fine is below 500.

Solution

```
UPDATE schedule SET fine = CASE WHEN fine > 500 THEN fine ELSE 500 END;
```

However, many students probably will use a where clause condition:

```
UPDATE schedule SET find = 500
WHERE find IS NULL OR fine < 500;
```


