LAST NAME: ______

FIRST NAME: Solution

DATABASE SYSTEMS
CSCI 331, course # 66047
CSCI 711, course # 66048
Test # 1
March 30, 2016
instructor: Bojana Obrenić

<u>NOTE:</u> It is the policy of the Computer Science Department to issue a failing grade in the course to any student who either gives or receives help on any test.

Your ability and readiness to follow the test protocol described below is a component of the technical proficiency evaluated by this test. If you violate the test protocol you will thereby indicate that you are not qualified to pass the test.

this is a **closed-book** test, to which it is **forbidden** to bring anything that functions as: paper, calculator, hand-held organizer, computer, telephone, camera, voice or video transmitter, recorder or player, or any device other than pencils (pens), erasers and clocks;

answers should be written only in the space marked "**Answer:** " that follows the statement of the problem (unless stated otherwise);

scratch should never be written in the answer space, but may be written in the enclosed scratch pad, the content of which *will not be graded;*

any problem to which you give **two or more (different) answers** receives the **grade of zero** automatically;

student name has to be written **clearly** on **each page** of the problem set and on the first page of the **scratch pad** the during the **first five minutes of the test**—there is a penalty of **at least 1 point** for each missing name;

when requested, **hand in** the problem set together with the scratch pad;

once you leave the classroom, you cannot come back to the test;

your **handwriting** must be legible, so as to leave no ambiguity whatsoever as to what exactly you have written.

You may work on as many (or as few) problems as you wish.
time: 75 minutes.

full credit: 100 points.

Good luck.

problem:	01	02	03	04	05	06	total:	[%]
grade:								

Problem 1 [**30 points**] Write the set of relational schemas for the database of **GloFaF**, specified by the narrative given on page 10. Implement all applicable constraints. Your design should have an adequate level of normalization. For each relational schema, specify the primary key (if any), and the foreign keys (if any).

LAST NAME:

FIRST NAME: Solution

Answer:

1) farm(name, location, address, phoneN)
2) employee(gloper, name, position, phoneN, farmId, salary, startDate)
3) animal(anid, speciesId, dateBorn)
4) residence(anid, farmid, lastAdmitted)
5) departure(anid, dateDeparted)
6) species(name, reference)
7) flyingSpecies(name, distance, height)
8) strongSpecies(name, force)
9) magicSpecies(name, adviser)

foreign keys:

field (set)	in	references
farmId	employee	farm
speciesId	animal	species
anid	residence	animal
farmId	residence	farm
anid	departure	animal
name	flyingSpecies	species
name	strongSpecies	species
name	magicSpecies	species
adviser	magicSpecies	employee

LAST NAME:

FIRST NAME: Solution

Problem 2 [**34 points**] Refer to page 10 for definition of a relational schema R, a dependency set F, and a decomposition of R into R_1 and R_2.

(a) Find *all* candidate keys for R, and prove that each one indeed is a candidate key and that there are no others.

Answer: KP is the only key. It is minimal as K, P do not occur on RHS in F. It is superkey, as $KP^+ = R$ which is proved by application of 10, 3, 7, 11, 13, 1, 5, 9, 4, 8, 6 in that order.

(b) Find a candidate key for R_1, and prove that it is a candidate key.

Answer: BDHN is the only key. R_1 sees 1, 5, 7, 2, 4. BDHN is minimal as it does not appear on the RHS in 1, 5, 7, 2, 4. BDHN is superkey, since $BDHN^+ = R_1$, by application of 1, 7, 5, 4, 2.

(c) Find a candidate key for R_2, and prove that it is a candidate key.

Answer: AJKP is the only key. R_2 sees 3, 11, 13, 10, 12. AJKP is minimal as it is not found on RHS. $AJKP^+ = R_2$, by application of 10, 3, 11, 13.

(d) Is it possible for two tuples of R to agree on the (composite) value under attributes PC but to disagree on the (composite) value under attribute M? Prove your answer.

No, since $PC \to M$ holds, as $PC^+ = ABCDEGLMP$ and $M \subseteq PC^+$. Calculate PC^+ by application of 10, 3, 1, 5, 4, 2 in that order.

(e) Does the decomposition of R into R_1 and R_2 have a lossless join? Prove your answer.

Answer: No

$R_1 \cap R_2 = ABDN$

$ABDN^+ = ABCDEGLMN$ by aplication of 1, 7, 4, 12, 2

$R_1 \not\subseteq (ABDN)^+$

$R_2 \not\subseteq (ABDN)^+$

(f) Does the decomposition of R into R_1 and R_2 preserve dependencies? Prove your answer.

Answer: No. 6, 8, 9 lost.

Note: This problem continues on the following page.

Problem 2, cont'd:

LAST NAME:

FIRST NAME: Solution

(g) State the highest normal form—if any—attained by the universal schema R and prove your answer.

Answer:

None.
D is not in 3NF say because in (1), D is not superkey (as KP is the only key) and E is not prime.

(h) State the highest normal form—if any—attained by the component R_1 and prove your answer.

Answer:

None. 3NF violated by (7): BD is not superkey as BDHN is the only key, and C is not prime

(i) State the highest normal form—if any—attained by the component R_2 and prove your answer.

Answer:

None, 3NF violated by (10). P is not superkey as AJKP is the only key, and G is not prime.

(j) Is the universal schema R adequately normalized? Prove your answer.

Answer:

No, as preceding.

(k) Is the decomposition of R into R_1 and R_2 satisfactory? Prove your answer.

Answer:

No, lossy join (part e), dependency loss (part f) and not normalized (parts h, i.)

(l) If your answer to part (k) is "NO", construct a good decomposition of R. If R cannot have a better decomposition, state it and explain your answer.

Answer:

DE
GB
BCA
BDC
KLGM
GKQ
GKN
GD
LM
AL
HEJ
AKH
PG
ANG
KP

(m) If you provided a decomposition in your answer to part (l), prove that it is good.

Answer:

obtained by algorithm which guarantees 3NF, lossless join, and dependency preserved

Problem 3 [**22 points**] Refer to page 10 and page 11 for definition of the **Database of Pure Wood.**

LAST NAME:

FIRST NAME: Solution

Assume that table `instance` is stored as a file of fixed-length records, such that the field `labelId` occupies 8 bytes and field `productId` occupies 8 bytes. This file has a primary index P and no other indices. The block size is equal to 8 kilobytes, and the block access time is approximately (1/2048) seconds. Sixteen gigabytes of core memory are available.

Assume that the file `instance`, when at its maximum intended size, accommodates about 8,000 billion product instances, and assume that the file is at this maximum intended size wherever this assumption may apply.

Answer the following questions, explain your answers briefly, and show the calculation that you have performed to obtain the answers. Express your results in reasonable units of physical quantities.

(a) Calculate the maximum possible number of bytes occupied by the file `instance`.

Answer:

$L = T\ell = 2^{43+4} = \boxed{2^{47}}$

$T \approx 2^3 \cdot 2^{10} \cdot 2^{30} = 2^{43}$

$\ell = 8 + 8 = 2^4$

(b) Calculate the maximum possible number of blocks the file `instance` may have.

Answer:

$B = \dfrac{L}{b} = 2^{47-13} = \boxed{2^{34}}$

$b = 8 \cdot 2^{10} = 2^{13}$

(c) State the name of the field on which the index P is built.

Answer:

LabelID

(d) Calculate the length of records of the file P.

Answer:

two fields:

1) labelId: 8 bytes

2) block pointer: at least 34 bits, round up to 8 bytes;

$\ell_P = 8 + 8 = \boxed{2^4}$

(e) Estimate the maximum possible number of records the file P may have.

Answer:

$T_P = B = \boxed{2^{34}}$

(f) Estimate the maximum possible number of blocks the file P may have.

Answer:

$B_P = \dfrac{T_P \ell_P}{b} = 2^{34+4-13} = \boxed{2^{25}}$

NOTE: This problem continues on the following page.

Problem 3 cont'd:

LAST NAME:

FIRST NAME: Solution

(g) Estimate the maximum possible number of pointers per node of the B-tree that implements the index P.

Answer:

$$\Delta_{max} = \frac{b}{\ell_p} = 2^{13-4} = \boxed{2^9}$$

(h) Estimate the minimum possible number of pointers per node of the B-tree that implements the index P.

Answer:

$$\Delta_{min} = \frac{\Delta_{max}}{2} = \boxed{2^8}$$

(i) Calculate the maximum possible depth of the B-tree that implements the index P.

Answer:

$$h_{max} = \lceil \log_{\Delta_{min}} B_p \rceil = \left\lceil \frac{\lg B_p}{\lg \Delta_{min}} \right\rceil = \left\lceil \frac{25}{8} \right\rceil = \boxed{4}$$

(j) Describe an adequate procedure (applicable under the assumptions given) to search the file `instance` according to the field `labelId`, estimate the worst-case running time of the procedure that you propose, and explain your analysis.

Answer:

traverse the B-tree index P, and perform one more data-file access. time:

$$(h_{max}+1)t_1 = (4+1)\cdot 2^{-11} < \boxed{2.5\,ms}$$

(k) Describe an adequate procedure (applicable under the assumptions given) to search the file `instance` according to the field `productId`, estimate the worst-case running time of the procedure that you propose, and explain your analysis.

Answer:

sequential scan.

time: $Bt_1 = 2^{34-11} = \boxed{2^{23}}$

seconds

LAST NAME:

FIRST NAME: Solution

Problem 4 [**18 points**] Refer to page 10 and page 11 for definition of the **Database of Pure Wood.** Write an expression in the **relational algebra** for the following queries. Be careful to use renaming where necessary. Your answer may employ only the following operators: $\sigma, \pi, \times, \cup, \setminus$. In particular, do not employ $\bowtie$. Your code ought to be not only correct but also readable.

(a) Find names and addresses of those suppliers that supply at least one product whose purpose is PASS EXAM.

Answer:

$$\pi_{name_1, address_1} \sigma_F (supplier_1 \times product_2 \times serve_3)$$

$$F = [name_1 = supName_2 \wedge code_2 = productCode_3 \wedge purposeTitle_3 = \text{'PASS EXAM'}]$$

(b) Find names and addresses of those discount stores where some items are located whose purpose is PREVENT SLEEPING IN CLASS.

Answer:

$$\pi_{name_1, address_1} \sigma_F (store_1 \times discountStore_2 \times location_3 \times instance_4 \times serves_5)$$

$$F = [name_1 = dName_2 \wedge dName_2 = storeId_3 \wedge labelId_3 = labelId_4 \wedge productId_4 = productCode_5 \wedge purposeTitle_5 = \text{'PREVENT SLEEP'}]$$

LAST NAME:

FIRST NAME: Solution

Problem 5 [**18 points**] Refer to page 10 and page 11 for definition of the **Database of Pure Wood.** Write an expression in the **relational algebra** for the following queries. Be careful to use renaming where necessary. Your answer may employ only the following operators: $\sigma, \pi, \times, \cup, \setminus$. In particular, do not employ $\bowtie$. Your code ought to be not only correct but also readable.

(a) Find product code, price, and supplier of those products that have no purpose on file.

Answer:

losers: $\Psi = \pi_{productCode}(serve)$

winners: $\rho = (\pi_{code}(product)) \setminus \Psi$

result:

$\Omega = \pi_{code_1, price_1, supName_1} \sigma_F (product_1 \times \rho_2)$

$F \equiv [code_1 = code_2]$

(b) Find name(s) and address(es) of the discount store(s) with the largest parking lot.

Answer:

losers: $\Psi = \pi_{dName_1} \sigma_F (discountStore_1 \times discountStore_2)$

$F \equiv [lotSize_1 < lotSize_2]$

winners: $\rho = (\pi_{dName}(discountStore)) \setminus \Psi$

result:

$\Omega = \pi_{name_1, address_1} \sigma_G (store_1 \times \rho_2)$

$G \equiv [name_1 = dName_2]$

LAST NAME: ______________________

FIRST NAME: Solution

Problem 6 [**18 points**] Refer to page 10 and page 11 for definition of the **Database of Pure Wood.** Write an expression in the **relational algebra** for the following queries. Be careful to use renaming where necessary. Your answer may employ only the following operators: $\sigma, \pi, \times, \cup, \setminus$. In particular, do not employ $\bowtie$.

Your code ought to be not only correct but also readable.

(a) Find names and addresses of those stores where some items are located whose purpose is EATING, but no items are located in those stores whose purpose is DRINKING.

Answer:

candidates:

$$\rho = \pi_{name_1, address_1} \sigma_F(store_1 \times location_2 \times instance_3 \times serve_4)$$

$$F \equiv [name_1 = storeId_2 \wedge labelId_2 = labelId_3 \wedge productId_3 = productCode_4 \wedge purposeTitle_4 = \text{'EATING'}]$$

losers:

$$\psi = \pi_{name_1, address_1} \sigma_G(store_1 \times location_2 \times instance_3 \times serve_4)$$

$$G \equiv [name_1 = storeId_2 \wedge labelId_2 = labelId_3 \wedge productId_3 = productCode_4 \wedge purposeTitle_4 = \text{'DRINK'}]$$

result: $\Omega = \rho \setminus \psi$

(b) Find name, address and salary of those managers who do not manage a discount store but manage a store where at least one item is located that costs more than $1,000.

Answer:

candidates:

$$\rho = \pi_{managerId_1} \sigma_F(manage_1 \times location_2 \times instance_3 \times product_4)$$

$$F \equiv [storeName_1 = storeId_2 \wedge labelId_2 = labelId_3 \wedge productId_3 = code_4 \wedge price_4 > 1000]$$

losers:

$$\psi = \pi_{managerId_1} \sigma_G(manage_1 \times discountStore_2)$$

$$G \equiv [storeName_1 = dName_2]$$

winners: $\alpha = \rho \setminus \psi$

result: $\Omega = \pi_{name_1, address_1, salary_1} \sigma_H(employee_1 \times \alpha_2)$

$$H \equiv [id_1 = managerId_2]$$

LAST NAME: ____________________

FIRST NAME: ____________________

Problem 1: The international company GloFaF (abbreviation of Global Farming Franchise) owns and operates many farms worldwide, nurturing many fine animals in most favourable conditions. GloFaF has a database.

The first concern of GloFaF data administrators is to keep track of the Company's own farms. Farms are identified by name, and further described by geographic region (location), regular mailing address and a preferred contact phone number. Every farm has many employees, which are identified by employee global personal number (GloPeN.) GloFaF records name, position title, and office phone number of each employee, along with the farm at which they are employed. Furthermore, the salary and employment start date are recorded for each employee.

Every animal that has ever been in care of GloFaF has its own individual identifier, called AnId (which stands for universal Animal Identifier.) In addition to AnId, GloFaF records the species to which the animal belongs and its date of birth. For each farm where an individual animal has resided, GloFaF records the date of the (most recent) admission of the animal to that farm. For those animals that are not hosted by GloFaF any more, the date of departure is recorded.

The species of GloFaF animals are named according to their regular scientific names. GloFaF does not store extensive descriptions of animal species, but instead each species is assigned a carefully reviewed reference bibliography entry, which in fact is a pointer to a single scientific publication (in an external library) that guarantees a good introduction into scientific facts about the particular species.

Some species have properties that are immediately relevant for the safety and well-being of individual animals and entire farms, and such properties of these special species are recorded as follows. For the family of flying species (such as eagles, chickens, bats) a record is kept of the normal distance to which animals of these species desire to fly, and the flying height at which they feel comfortable. For the strong species (such as dinosaurs), a minimum safe resistance force of the restraining enclosure is stored. Finally, to each of the magic species (such as dragons), an individual employee is assigned to serve in the role of adviser (about magic habits of these animals.)

Problem 2: Consider the relational schema

$$R = ABCDEGHJKLMNPQ$$

with the set of functional dependencies F given as follows.

(1)	$D \to E$	(2)	$E \to M$
(3)	$G \to BD$	(4)	$A \to L$
(5)	$BC \to A$	(6)	$HE \to J$
(7)	$BD \to C$	(8)	$AK \to H$
(9)	$KQ \to M$	(10)	$P \to G$
(11)	$GK \to Q$	(12)	$AN \to G$
(13)	$GK \to N$		

Schema R is decomposed into R_1 and R_2 as follows:

$$R_1 = ABCDEHLMN$$

$$R_2 = ABDGJKNPQ$$

Problems 3–6: **Pure Wood** is a furniture store chain that offers finest furniture for home, office, kitchen, camp, and more. *Pure Wood* has a large database, which organizes data about the current inventory, organization, and people. (**NOTE:** This narrative continues on the following page.)

LAST NAME: ____________________

FIRST NAME: ____________________

Pure Wood, cont'd:
All employees have their IDs, names, and addresses on file, along with the date they were hired, and salary they are paid.

Pure Wood keeps a register of its suppliers, identified by name. Along with the name, records are kept of the mailing address, fax number, and ordinary contact voice phone number(s), if any.

Pure Wood identifies its shops (stores) by name, characterizes each shop by address, and assigns to each shop its manager, identified by employee ID. Some *Pure Wood* stores are operated as discount stores. One important merit of discount stores is that they always have a huge parking lot nearby—indeed, *Pure Wood* records the size of the available parking lot of each discount store.

Every item sold by *Pure Wood* bears a product code, which identifies a class of all items that are equivalent (as furniture.) In more words, two identical chairs that come off the same assembly line would have equal product codes and would be considered as *two distinct instances of the same product.* Indeed, the two chairs bear the same product identifier because they are indistinguishable and equivalent in every aspect. However, they are two distinct chairs and thus bear distinct instance identifiers. This instance identifier is written on a sticker which the item bears since the moment it is acquired by *Pure Wood* until it is delivered to the final buyer.

For every item instance, at any moment, *Pure Wood* knows the store where that item instance is located, and can determine its product code from its instance identifier.

Every product must have a type (such as chair, cushion, pillow, or bookcase), weight (such as 50 kilograms), height (such as 0.8 meters), length, width, and colour (such as white.) Also, products may be associated with purposes (such as studying, sleeping, meditating, and more.) Some products may be suitable for multiple purposes (and some need not have any purpose on file.) Whenever a product is found suitable for an individual purpose, a photograph is stored which illustrates this. All instances of the same product are sold at the same price in all stores of *Pure Wood*, and that price is on file, along with the supplier who has supplied the product.

The conceptual schema of **Pure Wood** Database consists of the following stored tables.

```
employee (id, name, address, dateHired, salary)
supplier (name, address, email, fax)
supplierPhone (supName, voice)
store (name, address)
discountStore (dName, lotSize)
manage (storeName, managerId)
product (code, price, type, weight, height, length, width, color, supName)
purpose (title)
serve (productCode, purposeTitle, photo)
instance (labelId, productId)
location (labelId, storeId)
```

field	in schema	is foreign key referencing
supName	supplierPhone	supplier
dName	discountStore	store
storeName	manage	store
managerId	manage	employee
supName	product	supplier
productCode	serve	product
purposeTitle	serve	purpose
productId	instance	product
labelId	location	instance
storeId	location	store