Endterm Exam
CS 122A
Spring 2014

Max. Points: 100

(Please read the instructions carefully)

Instructions:
- The total time for the exam is 80 minutes; be sure to budget your time accordingly.
- The exam is closed book and closed notes but "open cheat sheet".
- Read each question first, in its entirety, and then carefully answer each part of the question.
- If you don't understand something, ask the instructor for clarification.
- If you still find ambiguities in a question, note the interpretation you are taking.

I THINK WE SHOULD BUILD AN SQL DATABASE.
UH-OH
DOES HE UNDERSTAND WHAT HE SAID OR IS IT SOMETHING HE SAW IN A TRADE MAGAZINE AD?
WHAT COLOR DO YOU WANT THAT DATABASE?
I THINK MAUVE HAS THE MOST RAM.
E-mail: SCOTTADAMS@AOL.COM
© 1995 United Feature Syndicate, Inc.(NYC)
S. Adams

NAME: **STUDENT ID:**

QUESTION	***TOPIC***	***POINTS***	***SCORE***
1	**SQL Constraints**	**10**	
2	**SQL Queries & Updates**	**30**	
3	**Physical DB Design**	**20**	
4	**B+ Trees**	**20**	
5	**Relational DB Design Theory**	**20**	
TOTAL	**All**	**100**	

Background Information

During this exam, as you are anxious to go live with your online car exchange idea and take your startup to an early IPO, you will be working more on your car exchange application. Some of the most important tables and columns from your very latest relational schema design are listed below. (Recall that in the project's E-R model, we said that every user is a Borrower and that some will also be Lenders.)

Tables: *Borrower* (*bid*, name, email, age, bcount, brating)
Lender (*lid*, lcount, lrating)
CarCatalog (*ccid*, category, make, model, year, color)
Car (*cid*, car_cat_id, color, owner_id)

```
create table Borrower(
  bid char(20) primary key,
  name varchar(100),
  email varchar(100),
  age integer,
  bcount integer,
  brating integer default 1);
```

```
create table Lender(
  lid char(20) primary key,
  lcount integer,
  lrating integer default 1,
  foreign key (lid) references Borrower(bid));
```

```
create table CarCatalog(
  ccid integer primary key,
  category varchar(20),
  make varchar(20),
  model varchar(30),
  color varchar(20),
  year integer);
```

```
create table Car(
  cid char(20) primary key,
  car_cat_id integer,
  color varchar(20),
  owner_id  char(20),
  foreign key (car_cat_id references CarCatalog(ccid),
  foreign key (owner_id) references Lender(lid));
```

Question 1: SQL Constraints (10 points)

(b) (10 pts) For each of the following business rules, odd as some of them might seem, circle the type of SQL feature that can be used to enforce it. If the given rule can't be handled in SQL, your answer should be "Application", i.e., this rule needs to be enforced in application code. If there are several ways to enforce a given rule, choose the one that seems like the best (i.e., most natural) fit.

(i) (2 pts) E-mail address is a candidate key for Borrower.

Primary key   Foreign key   Not null   Unique   Check   Trigger   Application

(ii) (2 pts) Borrowers must be at least 16 years old.

Primary key   Foreign key   Not null   Unique   Check   Trigger   Application

(iii) (2 pts) Borrower and lender ratings must be in the range from 0 to 5, inclusive.

Primary key   Foreign key   Not null   Unique   Check   Trigger   Application

(iv) (2 pts) A user whose borrower rating drops to a value below 1 should actually be deleted as a user.

Primary key   Foreign key   Not null   Unique   Check   Trigger   Application

(v) (2 pts) The current color of a car must be provided when registering it (in case it has been repainted).

Primary key   Foreign key   Not null   Unique   Check   Trigger   Application

SCORE: _________

Question 2: SQL Queries & Updates (30 points)

Tables: *Borrower* (*bid*, name, email, age, bcount, brating)
Lender (*lid*, lcount, lrating)
CarCatalog (*ccid*, category, make, model, year, color)
Car (*cid*, car_cat_id, color, owner_id)

Express each of the following queries (or other database actions) using the SQL language.

(a) (10 pts) Print a list of users who are lenders, ordered by name, at least one of whose cars' colors are still the same as that car's original color (as it is listed in the car catalog). The printed list should include the lender's user id, name, email, age, and their borrower and lender counts and ratings.

(b) (8 pts) Print a list of information about users who are under 18 years of age with e-mail accounts in the hotmail.com domain. The information printed should be each user's user id, name, email, age, borrow count, borrower rating, lend count, and lender rating. As some of the users may be borrowers but not lenders, the last two fields of the result may be null for some users, but *all* qualified users should indeed be represented in the results.

SCORE: _________

Question 2: SQL Queries & Updates (continued)

Tables: *Borrower* (*bid*, name, email, age, bcount, brating)
Lender (*lid*, lcount, lrating)
CarCatalog (*ccid*, category, make, model, year, color)
Car (*cid*, car_cat_id, color, owner_id)

Express each of the following queries (or other database actions) using the SQL language.

(c) (8 pts) Happy New Year! Your users are a year older and your lender rating system is changing! Write a SQL update statement (or a set of UPDATE statements) to add one to every user's age and to – if the user is a lender – double their lender rating (as that scale is expanding by 2x).

(d) (4 pts) Suppose that your UPDATE statement in (c) runs as a transaction T1. Suppose that there is another UPDATE statement, let's call it transaction T2, that adds 1 to some particular lender's lend count. Consider what can happen if T1 and T2 are run concurrently. (Fortunately, you have chosen a DBMS with good transaction support.) How many possible *serial* schedules are there? Show them briefly.

SCORE: _________

Question 3: Physical Database Design (20 points)

Tables: *Borrower* (*bid*, name, email, age, bcount, brating)
Lender (*lid*, lcount, lrating)
CarCatalog (*ccid*, category, make, model, year, color)
Car (*cid*, car_cat_id, color, owner_id)

You're getting ready to take your site live soon, so you need to design a physical storage and indexing strategy that will serve the following expected mixed query workload well. As you identify indexes, be sure to consider primary keys – i.e., you are to specify the information about those indexes as well as any secondary indexes that you identify as being beneficial. Assume that you are using a DBMS that supports both hashed and B+ tree indexes of both the clustered and unclustered varieties.

1. SELECT * FROM Borrower B WHERE B.email = ?
2. SELECT * FROM Lender L, Borrower B WHERE L.lid = B.bid AND B.age < ? AND B.age > ?
3. SELECT * FROM Car C, CarCatalog CC WHERE C.car_cat_id = CC.ccid AND C.year = 2000

(a) (6 pts) List the indexes that you recommend creating on the *Borrower* table; for each index that you recommend, specify the column(s) it should index on and whether or not it is clustered and/or unique.

Indexed Columns	Hashed or B+ Tree?	Clustered? (Y/N)	Unique? (Y/N)

(b) (4 pts) List the indexes that you recommend creating on the *Lender* table; for each index that you recommend, specify the column(s) it should index on and whether or not it is clustered and/or unique.

Indexed Columns	Hashed or B+ Tree?	Clustered? (Y/N)	Unique? (Y/N)

(c) (6 pts) List the indexes that you recommend creating on the *CarCatalog* table; for each index that you recommend, specify the column(s) it should index on and whether or not it is clustered and/or unique.

Indexed Columns	Hashed or B+ Tree?	Clustered? (Y/N)	Unique? (Y/N)

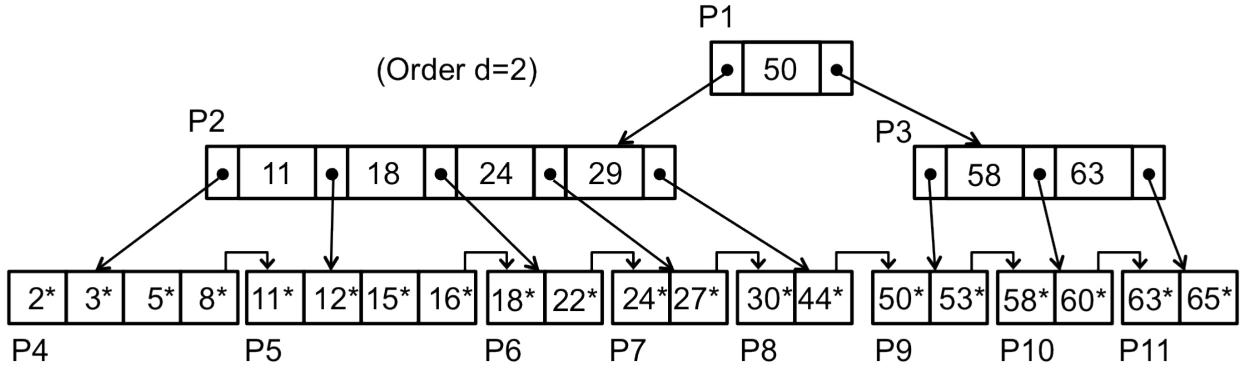
(d) (4 pts) List the indexes that you recommend creating on the *Car* table; for each index that you recommend, specify the column(s) it should index on and whether or not it is clustered and/or unique.

Indexed Columns	Hashed or B+ Tree?	Clustered? (Y/N)	Unique? (Y/N)

SCORE: _________

Question 4: B+ Trees (20 points)

Consider the following example of a B+ tree index of order d=2. Suppose that this is a dense secondary B+ tree index (on User.age) whose leaf entries represent key/record-id pairs.

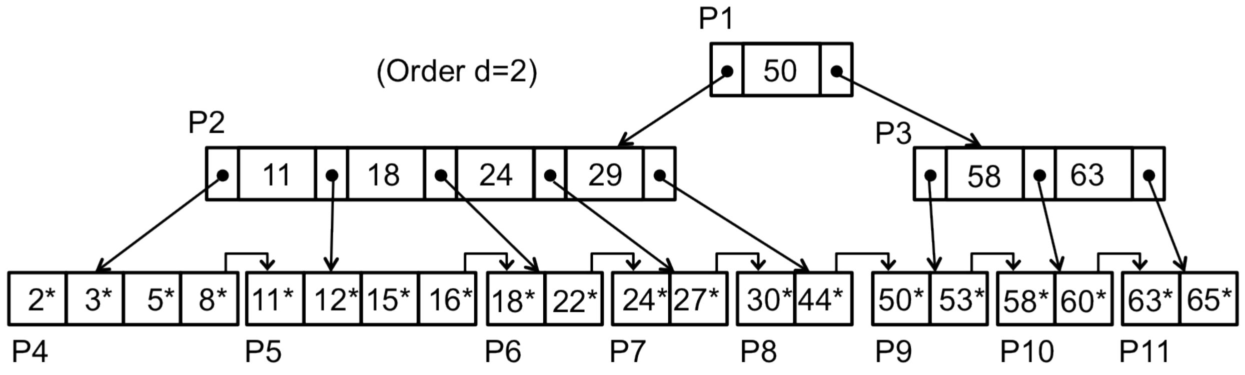


(a) (6 pts) List the page numbers of the pages that will be read when using this B+ tree index to perform the query SELECT * FROM User U WHERE U.age >= 25 and U.age < 55.

(b) (5 pts) Draw the B+ tree index that will result from inserting a new entry 13* with key value 13 and its associated record id. (Label the pages in your picture; reuse the same labels as above, inventing new labels only if they denote new pages.)

SCORE: _________

Question 4: B+ Trees (continued)



(c) (4 pts) Now suppose that the picture above represents an ISAM index instead of a B+ tree index. (Recall that ISAM indexes are essentially prehistoric B+ trees.) Draw the ISAM index that would have resulted from inserting the new entry 13* with key value 13 and its associated record id into this ***original ISAM*** index structure. (Label the pages in your picture; reuse the same labels as above, inventing new labels only if they denote new pages.)

(d) (5 pts) Now let us return to the modern world of B+ trees. Draw the B+ tree index that will result from deleting the entry labeled 58* (with key value 58 and its associated record id) from the ***original*** B+ tree index structure. (Label the pages in your picture; reuse the same labels as above, inventing new labels only if they denote new pages.)

Question 5: Logical Database Design (20 points)

Consider the relations and functional dependencies (FDs) specified in each sub-problem below. For each one, first identify the relation's candidate key(s) and current highest normal form and then decompose the relation (if needed) into 3NF or BCNF. Your final set of relations in each case should be a lossless join, dependency-preserving decomposition of the original relation with respect to the given FDs.

(a) (7 pts) S (p, q, r, s, t, u, v,w) with $t \rightarrow w$, $t \rightarrow u$, $p \rightarrow q$, $p \rightarrow r$, $(r,q) \rightarrow s$, $(r,q) \rightarrow t$, $p \rightarrow v$

Candidate key(s):

Current highest normal form:

Normalized design:

(b) (7 pts) T(a, b, c, d, e, f, g, h) with $a \rightarrow c$, $a \rightarrow f$, $a \rightarrow g$, $(a,b) \rightarrow h$, $b \rightarrow d$, $b \rightarrow e$, $b \rightarrow f$

Candidate key(s):

Current highest normal form:

Normalized design:

SCORE: _________