Final Exam  
CS 122A  
Winter 2013  
Max. Points: 100

(Please read the instructions carefully)

Instructions:
- The total time for the exam is 120 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.
- The last page of this exam is blank; you can use it as scratch paper.

---

**NAME:**  M. KEREKC  
**STUDENT ID:**

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>POINTS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100</strong></td>
<td>\textbf{100}</td>
</tr>
</tbody>
</table>
Background Information

During this exam you will be helping a new eBay wannabe, scrappy little Newport Bay online auction startup myBay.com, to get its data management story all figured out. Their most important tables and columns are listed below. You should take a few minutes to soak in this data now, as then you will not have to waste time later on reading and interpreting new database descriptions. The short list of tables below will be repeated again where needed; you can refer back to this page later if you forget what the columns mean or what their data types are.

Tables:  
- **User** (uid, email, name, age, joindate)  
- **Item** (itemid, name, descrip, condition, minprice, sellerid, enddate, soldprice)  
- **Bid** (itemid, bidnum, bidderid, bidprice)

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Description</th>
<th>SQL Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>uid</td>
<td>User id (primary key)</td>
<td>INTEGER</td>
</tr>
<tr>
<td>User</td>
<td>email</td>
<td>User’s e-mail address (not nullable)</td>
<td>VARCHAR(50)</td>
</tr>
<tr>
<td>User</td>
<td>name</td>
<td>User’s name</td>
<td>VARCHAR(50)</td>
</tr>
<tr>
<td>User</td>
<td>age</td>
<td>User’s age in years</td>
<td>INTEGER</td>
</tr>
<tr>
<td>User</td>
<td>joindate</td>
<td>User’s date of joining myBay.com</td>
<td>DATE</td>
</tr>
<tr>
<td>Item</td>
<td>itemid</td>
<td>Item id (primary key)</td>
<td>VARCHAR(15)</td>
</tr>
<tr>
<td>Item</td>
<td>name</td>
<td>Item’s name (not nullable)</td>
<td>VARCHAR(100)</td>
</tr>
<tr>
<td>Item</td>
<td>descrip</td>
<td>Item’s description</td>
<td>VARCHAR(500)</td>
</tr>
<tr>
<td>Item</td>
<td>condition</td>
<td>Item’s stated condition</td>
<td>VARCHAR(25)</td>
</tr>
<tr>
<td>Item</td>
<td>minprice</td>
<td>Item’s minimum acceptable bid price</td>
<td>DECIMAL(8,2)</td>
</tr>
<tr>
<td>Item</td>
<td>sellerid</td>
<td>Item’s seller (refers to User.uid)</td>
<td>INTEGER</td>
</tr>
<tr>
<td>Item</td>
<td>enddate</td>
<td>Item’s closing date for the auction</td>
<td>DATE</td>
</tr>
<tr>
<td>Item</td>
<td>soldprice</td>
<td>Item’s final selling price (null if unsold)</td>
<td>DECIMAL(8,2)</td>
</tr>
<tr>
<td>Bid</td>
<td>itemid</td>
<td>Bid’s auction item (refers to Item.itemid)</td>
<td>VARCHAR(15)</td>
</tr>
<tr>
<td>Bid</td>
<td>bidnum</td>
<td>Bid sequence number (w.r.t. the auction item)</td>
<td>INTEGER</td>
</tr>
<tr>
<td>Bid</td>
<td>bidderid</td>
<td>Bid’s bidding user (refers to User.uid)</td>
<td>INTEGER</td>
</tr>
<tr>
<td>Bid</td>
<td>bidprice</td>
<td>Bid’s offered price for the item</td>
<td>DECIMAL(8,2)</td>
</tr>
</tbody>
</table>

Question 1: SQL DDL (15 points)

(a) (5 pts) Using SQL DDL, write a proper SQL CREATE TABLE statement for the **Item** table that reflects everything shown/described above (including all of the “key concepts” in the description).

```sql
CREATE TABLE Item  
  (  
    itemid VARCHAR(15) NOT NULL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    descrip VARCHAR(500),  
    condition VARCHAR(25),  
    minprice DECIMAL(8,2),  
    sellerid INTEGER REFERENCES User(uid) ON DELETE CASCADE,  
    enddate DATE,  
    soldprice DECIMAL(8,2)  
  )
```

SCORE: 5
Question 1: SQL DDL (continued)

Tables:  
User (uid, email, name, age, joindate)  
Item (itemid, name, descrip, condition, minprice, sellerid, enddate, soldprice)  
Bid (itemid, bidnum, bidderid, bidprice)

(b) (10 pts) The myBay.com application developers are aware of a number of constraints that need to be enforced to ensure data validity and integrity, but they aren’t exactly five-star SQL wizards. For each of the following English descriptions of a constraint, indicate the kind of SQL constraint or SQL feature that can be used to enforce it. If the indicated constraint can’t be handled in SQL, your answer can simply be: “This one needs to be enforced in application code.” If there are multiple ways to enforce one of the constraints, choose the one that seems like the best (i.e., most natural/direct) fit. Consider the following universe of options: primary key, foreign key, not null, unique, check, trigger, grant/revoke.

(i) (2 pts) The (only) valid item conditions are “poor”, “fair”, “good”, “very good”, and “excellent”.

CHECK

(ii) (2 pts) Bid prices for an item must be monotonically increasing; that is, each bid price for an item must be larger than the last bid price for that same item.

TRIGGER

(iii) (2 pts) Each bid for an auction item must be associated with a valid user.

FOREIGN KEY

(iv) (2 pts) Two different users may not share the same e-mail address.

UNIQUE

(v) (2 pts) A bid may not be placed on an item once it has been sold or after the auction’s end date.

TRIGGER

Question 2: SQL Reloaded (35 points)

Tables:  
User (uid, email, name, age, joindate)
Item (itemid, name, descrip, condition, minprice, sellerid, enddate, soldprice)
Bid (itemid, bidnum, bidderid, bidprice)

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

(a) (5 pts) Some myBay.com sellers are trying to make their auction items look “hot” by bidding on them themselves. Print an alphabetized list of the names, user ids, and e-mail addresses of such sellers so that the company can contact them and ask them to stop.

```sql
SELECT DISTINCT U.name, U.uid, U.email
FROM User U, Item I, Bid B
WHERE U.uid = I.sellerid AND I.itemid = B.itemid
AND U.uid = B.bidderid
ORDER BY U.name
```

(b) (7 pts) The team of business analysts at myBay.com wants to analyze the bidding behavior of users who are highly active. Create a view containing information about all users who have placed bids on 25 or more distinct items. For each such user, the view should list their user id, name, joindate, number of distinct items bid on, total number of bids, minimum bid price, and maximum bid price.

```sql
CREATE VIEW Active (uid, name, joindate, items, bids, minbid, maxbid) AS
SELECT U.uid, U.name, U.joindate, COUNT(DISTINCT I.itemid),
COUNT(DISTINCT B.bidprice), MIN(B.bidprice), MAX(B.bidprice)
FROM User U, Bid B
WHERE U.uid = B.bidderid
GROUP BY U.uid, U.name, U.joindate
HAVING COUNT(DISTINCT I.itemid) >= 25
```

(c) (7 pts) myBay.com’s wealthy CEO is Gil Bates, and Gil is one of the company’s users himself. He wants to please the company’s users and gain early national attention by having outstanding successful sale metrics. Write an update statement for the Item table that Gil can use to personally buy all of the unsold items whose auctions ended yesterday (a.k.a. CURRENT_DATE - 1). He wishes to buy each such item at 5% over its minimum acceptable bid price.

```sql
UPDATE Item
SET soldprice = 1.05 * minprice
WHERE I.soldprice IS NULL
AND I.enddate = (CURRENT_DATE - 1)
```

**SCORE: 19**
Question 2: SQL Reloaded (continued)

Tables:  

- User (uid, email, name, age, joindate)  
- Item (itemid, name, descrip, condition, minprice, sellerid, enddate, soldprice)  
- Bid (itemid, bidnum, bidderid, bidprice)

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

(d) (7 pts) The business analysis group is now turning its attention to analyzing the nature of auction activity for very high end items. For those items whose minimum bid price is listed as $1 million or more, list their item id, minimum acceptable bid price, number of bids, and bid ratio (highest/lowest) based on the highest and lowest bids that were made for the item. For completeness, be sure not to exclude items from the analysis that are unsold and/or that have not been bid upon.

```
SELECT I.itemid, I.minprice, COUNT(B.bidnum), MAX(B.bidprice)/MIN(B.bidprice)
FROM Item I LEFT OUTER JOIN Bid B
ON (I.itemid = B.itemid)
WHERE I.minprice >= 1000000.00
GROUP BY I.itemid, I.minprice
```

(e) (9 pts) Gil is at it again, and this time he wants to make sure that myBay.com gains a reputation for dealing largely in quality goods from quality sellers. Write the SQL delete statement(s) that Gil can use to remove those users for whom all of the items listed for sale have been in "poor" condition. To get full credit, explain (briefly!) why your answer suffices to ensure that those users and their junky items will both be gone.

```
DELETE FROM User U
WHERE NOT EXISTS
    (SELECT * FROM Item I
     WHERE I.condition < 'poor' AND I.sellerid = U.uid)
AND EXISTS
    (SELECT * FROM Item I
     WHERE I.sellerid = U.uid)
```

This will suffice because if deletes sellers of only poor condition items—and that deletes through use of a foreign key constraint on Item.sellerid with ON DELETE CASCADE, will also cascade delete of his/her items as well.

SCORE: 16
Question 3: Physical Database Design (15 points)

Tables:  
User (uid, email, name, age, joindate)  
Item (itemid, name, descrip, condition, minprice, sellerid, enddate, soldprice)  
Bid (itemid, bidnum, bidderid, bidprice)

Because of your performance on questions 1 and 2, you are now a finalist for a DBA job at myBay.com! If you get the job, significant stock options will be included in your offer. One of your interviewers has given you the following list of queries and wants you to design a physical storage and indexing strategy that will serve this mixed query workload well. NOTE: You must ensure that your proposed physical design will also efficiently support enforcement of the primary key constraint for each of the tables.

1. SELECT * FROM User U WHERE U.email = ?
2. SELECT * FROM User U WHERE U.joindate >= (CURRENT DATE - 7)
3. SELECT I.minprice FROM Item I WHERE I.name = ?
4. SELECT AVG(B.bidprice) FROM Item I, Bid B WHERE I.name = ? AND B.itemid = I.itemid

(a) (5 pts) List the indexes that you recommend creating on the User 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.

<table>
<thead>
<tr>
<th>Indexed Columns</th>
<th>Unique?</th>
<th>Clustered?</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>email</td>
<td>(Y)</td>
<td>✓</td>
</tr>
<tr>
<td>joindate</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

(b) (5 pts) List the indexes that you recommend creating on the Item 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.

<table>
<thead>
<tr>
<th>Indexed Columns</th>
<th>Unique?</th>
<th>Clustered?</th>
</tr>
</thead>
<tbody>
<tr>
<td>itemid</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>name</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>name, minprice</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

(c) (5 pts) List the indexes that you recommend creating on the Bid 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.

<table>
<thead>
<tr>
<th>Indexed Columns</th>
<th>Unique?</th>
<th>Clustered?</th>
</tr>
</thead>
<tbody>
<tr>
<td>itemid, bidnum</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

SCORE: 15
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 whose leaf entries represent key/record-id pairs. (The disk pages that comprise the tree are labeled for easy reference below.)

(a) (6 pts) List the page numbers of the pages that must be read to search for key values 18 and then 75. What’s similar about these two lists and what very important B+ tree property does that illustrate?

Pages read to search for key = 18: P1, P2, P6
Pages read to search for key = 75: P1, P3, P11
Similarity noted: BOTH HAVE SAME LENGTH (2)
B+ tree property: THEY ARE BALANCED FOR LOW SEARCH COST

(b) (6 pts) Draw the index that will result from inserting a new entry with key value 14. (Label the pages in your picture; reuse the same labels as above, inventing new labels only if they denote new pages.)
Question 4: B+ Trees (continued)

(c) (4 pts) How many disk reads and disk writes does the insert 14* operation in (b) require? Only count the first read and/or first write of a given page; e.g., if you read the root twice, just count that as one page read since the page will be in memory after the first read.

# reads = 3
list of pages read: P1, P2, P5

# writes = 5
list of pages written: P5, P12, P2, P3, P1

(d) (4 pts) Now consider the general case of a B+ tree of order d where d is much larger number (e.g., 100). Suppose such a tree is currently k levels deep (where d=2 and k=3 in the picture above). State the number of page reads and page writes needed for an insert operation in the common case for such a tree; express your answer in terms of d and/or k, and briefly (!) justify your answer.

# reads = k

# writes = 1

explanation: No split occurs, p; the common case of just 1 write

SCORE: 8
Question 5: Logical Database Design (15 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) (5 pts) I (iid, nm, dsc, cond, minp, slid, end, soldp)

   iid \rightarrow nm, nm \rightarrow iid, iid \rightarrow dsc, iid \rightarrow cond, cond \rightarrow minp, iid \rightarrow slid, iid \rightarrow end, iid \rightarrow soldp

Candidate key(s): iid, nm

Current highest normal form: 2NF

Normalized design:

\[
\begin{align*}
I_1 &= (iid, dsc, slid, end, soldp, cond) \\
I_2 &= (iid, nm) \\
I_3 &= (cond, minp)
\end{align*}
\]

(b) (5 pts) S (p, q, r, s, t, u)

p \rightarrow q, p \rightarrow r, p \rightarrow s

Candidate key(s): (p, t, u)

Current highest normal form: 1NF

Normalized design:

\[
\begin{align*}
S_1 &= (p, q, r, s) \\
S_2 &= (p, t, u)
\end{align*}
\]

(c) (5 pts) scrip (drg, pathname, pathname, dr)

Candidate key(s): (drg, pathname, pathname, dr)

Current highest normal form: BCNF

Normalized design: Already in BCNF

SCORE: 15