

Name: \_\_\_\_\_

This test is **open book, open notes**, but there can be no sharing of any material, and no communication except with me. You can use the Internet, but only as a library. If you are not sure if something is allowed, check with me.

Many questions in this quiz use the database schema and sample instance depicted below. The `AuctionItems` table records information about items in a silent auction (as used by *eBay* and similar services). For each item, the table records a unique identifier `id`, a brief description `bdesc`, a longer description `ldesc`, the starting price for bidding `sprice`, the shipping and handling charge `shcost`, and the date and time at which the auction ends `etime`. The `Bids` table holds information about bids placed on the items in the auction. For each bid, the table records a unique bid confirmation number `conf`, the name of the bidder `bidder`, the identifier of the item on which the bid is placed `item`, the amount of the bid `amount`, and the date and time the bid was placed `btime`. The type of each attribute appears directly below its name. *Primary key attributes* are underlined.

When asked for queries, you must provide answers that work for all possible database instances, not just the example instance depicted below. For brevity, relational algebra expressions abbreviate the schemas of the `AuctionItems` and `Bids` tables as  $A(I, B, L, P, S, T)$  and  $B(C, B, I, A, T)$ , respectively.

**The following tables are repeated on the last page of the quiz.** You may detach that page and use it for reference. There is no need to reattach it

AuctionItems

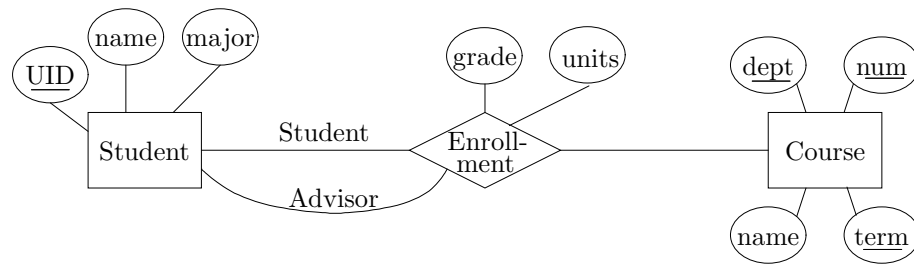
<u>id</u>	bdesc	ldesc	sprice	shcost	etime
char(10)	varchar(100)	text	decimal(6,2)	decimal(6,2)	timestamp
101	Amazing Aardvark	A pet...	101.99	51.00	2005-03-10 11:00:00
115	Befuddling Bees	Ten for...	10.99	11.00	2005-03-11 17:00:00

Bids

<u>conf</u>	bidder	item	amount	btime
char(12)	varchar(100)	char(10)	decimal(6,2)	timestamp
101101	alice01	101	105.00	2005-03-10 01:00:00
101221	bob02	101	107.00	2005-03-10 01:05:00
101335	alice01	101	125.00	2005-03-10 01:10:00

- (1 pt.) Write your name in the space provided above.

2. (5 pts.) Provide an ODL model that is as faithful as possible to the following E-R model.



3. (5 pts.) Write a SQL trigger that disallows bids with a timestamp that is later than the end time of the corresponding item's auction. That is, all transactions that attempt to insert a tuple into the Bids table with `btime` greater than the corresponding item's `etime` in `AuctionItems` should be rolled back.

4. (4 pts.) Assuming that the tables `AuctionItems` and `Bids` already exist in the database (without any constraints), write SQL statements to declare the primary key constraints for both tables. (The tables are not to be dropped.)
5. (5 pts.) Given the state of the database immediately after execution of the SQL statements in your answer to Question 4, write a SQL statement to express the constraint that every value appearing in the `item` column of `Bids` must also occur in the `id` column of `AuctionItems`.
6. (5 pts.) Given the state of the database immediately after execution of the SQL statements in your answer to Question 5, write a SQL statement to express the constraint that values appearing in the `sprice` and `shcost` columns must be nonnegative.

7. (5 pts.) Given the state of the database immediately after execution of the SQL statements in your answer to Question 6, write a SQL statement to express the constraint that every bid for an item must list an amount that is at least as large as that item's start price.
8. (5 pts.) We say a bid is *active* if there is no bid for the same item that has a greater amount. Define a bidder's *amount outstanding* to be the sum of the amounts in that bidder's active bids. Given the state of the database immediately after execution of the SQL statements in your answer to Question 7, write a SQL statement to express the constraint that the amount outstanding cannot exceed 1000.

9. (5 pts.) Write a SQL statement that creates a view `BidsIn` that consists of the second, third, and fourth columns of the `Bids` table.

10. (5 pts.) For the view `BidsIn` of Question 9, create a trigger that results in the following behavior: When a transaction attempts to insert a tuple  $(b, i, a)$  into `BidsIn`, a tuple  $(c, b, i, a, t)$  is inserted into `Bids` instead, where  $t$  is the time of day at the time of insertion and  $c$  is the string obtained by concatenating  $b$  and  $t$  (with  $t$  expressed as a string denoting the number of seconds since 1970-01-01 00:00:00).

You may assume that the SQL function `current_timestamp` returns the current date and time as a timestamp and that the function `extract(epoch, T)` returns an integer that is the timestamp `T` expressed as the number of seconds since 1970-01-01 00:00:00.

11. (5 pts.) Write one or more SQL triggers that prohibit the insertion into **Bids** of tuples that satisfy any of the following conditions:
- (a) There is a tuple in **Bids** with the same item as the new tuple and with an amount that is greater than the new tuple's amount minus one.
  - (b) There is a tuple in **Bids** with the same item and bidder as the new tuple, and this tuple has the highest amount of all tuples *for that item*.
- (Intuitively, the above conditions ensure that a new bid for an item is at least one dollar greater than the current highest bid and that bidders cannot outbid themselves.)

## Scratch page

Material here will not be graded. You may detach and discard this page.

AuctionItems

<u>id</u>	bdesc	ldesc	sprice	shcost	etime
char(10)	varchar(100)	text	decimal(6,2)	decimal(6,2)	timestamp
101	Amazing Aardvark	A pet...	101.99	51.00	2005-03-10 11:00:00
115	Befuddling Bees	Ten for...	10.99	11.00	2005-03-11 17:00:00

Bids

<u>conf</u>	bidder	item	amount	btime
char(12)	varchar(100)	char(10)	decimal(6,2)	timestamp
101101	alice01	101	105.00	2005-03-10 01:00:00
101221	bob02	101	107.00	2005-03-10 01:05:00
101335	alice01	101	125.00	2005-03-10 01:10:00