COMP 231 - Fall 2006
MIDTERM EXAM

Name: SOLUTION	Student ID:
ITSC Account:	Program:

All questions should be answered within the space provided after each problem.

PROBLEM 1 (15%) - ER Diagrams
Suppose an art gallery has a database system containing the following five relational tables:

PAINTING (PID, TITLE, *AID*, CATEGORY, PRICE)

ARTIST (AID, NAME, CITY)

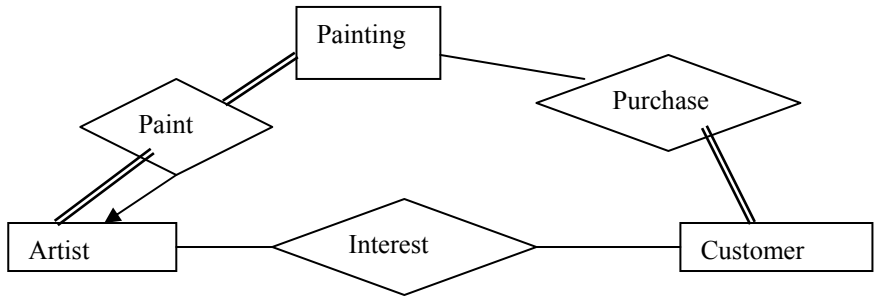
CUSTOMER (CID, NAME, CREDIT)

PURCHASE (*CID*, *PID*, YEAR)

INTEREST (*CID*, *AID*)

ASSUMPTIONS:
Keys are underlined and foreign keys are in *italics*. Each artist has painted at least one painting. Each painting is painted by exactly one artist. Each customer has purchased at least one painting and each painting may be purchased by zero or more customers. A customer can be interested in the work of any number of artists and an artist may draw interest from any number of customers.

Draw an E-R diagram that results in these five tables.



State your assumptions (if any):

Attributes are omitted in the diagram.

PROBLEM 2 (48%) - SQL, Algebra, Calculus

Using the tables of Problem 1, write the following queries in SQL, relational algebra, and domain relational calculus.

Query 1: Find the distinct titles of all paintings painted by artists from the city of Monterey.

SQL (3%):

```
SELECT DISTINCT P.TITLE
FROM PAINTING P, ARTIST A
WHERE A.CITY = 'Monterey' AND A.AID = P.AID
```

algebra (3%):

π_{TITILE} ($\sigma_{CITY = 'Monterey'}$ (PAINTING JOIN ARTIST))

calculus (3%):

{<T> | ∃ I,A, C, P (<I,T,A,C,P>∈PAINTING AND ∃ N, CI (<A, N, CI>∈ARTIST AND CI='Monterey'))}

Query 2: Find the unique IDs of all customers who bought paintings from both "PORTRAIT" and "STILL LIFE" categories.

SQL (4%):

```
SELECT DISTINCT PU1.CID
FROM PURCHASE PU1, PAINTING PA1
WHERE PU1.PID = PA1.PID
    AND PA1.CATEGORY = 'PORTRAIT'
    AND PU1.CID IN
            (SELECT PU2.CID
            FROM PURCHASE PU2, PAINTING PA2
            WHERE PU2.PID = PA2.PID AND PA2.CATEGORY = 'STILL LIFE')
```

algebra (4%):

(π_{CID} ($\sigma_{CATEGORY= 'PORTRAIT'}$ PAINTING JOIN PURCHASE))
∩
(π_{CID} ($\sigma_{CATEGORY ='STILL LIFE'}$ PAINTING JOIN PURCHASE))

calculus (4%):

{<CID> | ∃PI1,Y1 <PI1,CID,Y1>∈PURCHASE AND ∃ PI2, Y2 <PI2,CID,Y2>∈PURCHASE AND ∃T1, A1, CA1, PR1 (<PI1,T1,A1,CA1,PR1>∈PAINTING AND CA1='PORTRAIT') AND ∃T2,A2,CA2,PR2 (<PI2,T2,A2,CA2,PR2>∈PAINTING AND CA2='STILL LIFE')}

Query 3: Find the unique IDs of the customers with the highest credit.

SQL (4%):

```
SELECT C1.CID
FROM CUSTOMER C1
WHERE C1.CREDIT =
    (SELECT MAX(C2. CREDIT) FROM CUSTOMER C2)
```

algebra (4%):

π_{CID} (CUSTOMER) – $\pi_{C1.CID}$ ($\sigma_{(C1.CREDIT < C2.\ CREDIT}$ (P_{C1} (CUSTOMER) X P_{C2} (CUSTOMER)))

calculus (4%):

{<CID> | ∃ N,C (<CID,N,C>∈CUSTOMER AND¬ ∃ CID1,N1,C1 (<CID1,N1,C1>∈CUSTOMER AND C1>C))}

Query 4: Find the distinct IDs of the customers that are interested in **all** artists from the city of Monterey.
(HINT: this is a division query)

SQL (5%):

```
SELECT CID
FROM INTEREST I1
WHERE NOT EXISTS
    ((SELECT AID
    FROM ARTIST A
    WHERE CITY = 'Monterey')
    EXCEPT
    (SELECT I2.AID
    FROM INTEREST I2
    WHERE I2.CID = I1.CID))
```

algebra (5%):

INTEREST / (π_{AID} ($\sigma_{CITY = 'MONTEREY}$ ARTIST))

calculus (5%):

{<CID> |$\forall$ A,N,C(<A,N,C>$\in$ARTIST AND C='Monterey' $\Rightarrow$ < CID, A> $\in$ INTEREST)}

PROBLEM 3 (12%) – SQL (group-by)
Using the tables of Problem 1, write the following queries in SQL.

Query 5: Show the artist name and the number of his/her paintings in the category of "PORTRAIT", for artists who have more than 10 paintings in the subject of "PORTRAIT".

SQL (6%):

```
SELECT ARTIST.NAME, COUNT(PID)
FROM ARTIST, PAINTING
WHERE ARTIST.AID = PAINTING.AID AND CATEGORY = "PORTRAIT"
GROUP BY ARTIST.AID, ARTIST.NAME
HAVING COUNT(PID)>10
```

Query 6: Show the id(s) of the customer(s) that made the largest amount of total spending in year 2005.

SQL (6%):

```
SELECT CID
FROM
      (SELECT CID, SUM(PRICE) AS TOTAL
       FROM  PURCHASE,PAINTING
       WHERE PURCHASE.PID=PAINTING.PID AND PURCHASE.YEAR = 2005
       GROUP  BY CID) AS TEMP
WHERE TOTAL = (SELECT MAX(TOTAL) FROM TEMP)
```

PROBLEM 4 (25%) - Relational Database Design
Consider a table **R** containing the following attributes:
(Painting_ID, Painting_Title, Category, Artist_Name, Artist_ID, Artist_City)

We are given the following constraints:
1] We cannot have two records with the same Painting_ID, but different Painting_Titles.
2] A painting of a given Painting_ID belongs to a single category.
3] We cannot have two records with the same Artist_ID but different Artist_Names.
4] We cannot have two records with the same Artist_ID but different Artist_Cities.

4.1 (6%) Write a minimal set of functional dependencies (canonical cover), capturing the above constraints.

Painting_ID → Painting_Title, Category
Artist_ID → Artist_Name, Artist_City

4.2 (3%) Write the candidate key(s) for table R:

(Painting_ID, Artist_ID)

4.3 (6%) Is R in 3NF? If yes explain briefly why. If not, provide a 3NF decomposition of R.

No. Both FDs violate 3NF. Decompose it into the following three relations:
R1 = (Painting_ID, Painting_Title, Category)
R2 = (Artist_ID, Artist_Name, Artist_City)
R3 = (Painting_ID, Artist_ID)

4.4 (10%) Is R in BCNF? If yes explain why. If not, provide a step-by-step BCNF decomposition and discuss if your decomposition is dependency preserving.

No. Both FDs violate BCNF. Decompose it as follows:

Step 1:
R1 = (Painting_ID, Painting_Title, Category)
R2 = (Painting_ID, Artist_Name, Artist_ID, Artist_City)

Step 2: Decompose R2 because it violates BCNF

R1 = (Painting_ID, Painting_Title, Category)
R3 = (Artist_ID, Artist_Name, Artist_City)
R4 = (Painting_ID, Artist_ID)

(R1, R3, R4) is the final BCNF decomposition schema. It is dependency-preserving.