INFS 614, Dr. Jessica Lin

Homework #4

Problem 1. Consider the following relation schemas for an Art Gallery database. Primary keys are underlined.

Relation Schemas	Meaning and additional information
Artists (<u>artist_name</u>, age, birthplace, style)	Artists have a name, an age, a birthplace, and the style of art
Customers (<u>cust_name</u>, address, amount)	Each customer has a name, address, and the total amount of dollars spent in the gallery
Artwork_Paints (<u>title</u>, artist_name, type, price, year)	Each piece of artwork has a title, the artist that painted it, type (e.g. painting, photograph, etc), price, and the year it was painted
Like_Artist (<u>cust_name, artist_name</u>)	Each customer has his/her favorite artist(s)
Art_Group (<u>group_name</u>)	Names of different groups of artwork (e.g. 19th century art, portrait, etc)
Like_Group (<u>cust_name, group_name</u>)	Each customer has his/her favorite group(s) of artwork
Classify (<u>title, group_name</u>)	Classify each piece of artwork into different groups.

Implement the following queries in SQL.

1. Find the number of artists whose birthplaces are known (i.e. birthplace is not 'NULL')
2. For each art group, list the number of art pieces classified into that group.
3. For each art group, list the total price for all the arts belonging in that group.
4. For each art group, list the number of distinct artists that contribute to that group after 1950 (i.e. consider only art pieces created after 1950).

REM #1

SELECT COUNT(*) AS number_of_artists FROM artists WHERE birthplace IS NOT NULL;

```
REM #2

SELECT C.group_name, count(c.title) FROM classify C

                GROUP BY c.group_name;

REM #3

SELECT C.group_name, SUM(ap.price)

FROM classify C, artwork_paints AP WHERE c.title = ap.title

                GROUP BY c.group_name;

REM #4

SELECT C.group_name, COUNT(AP.artist_name)

FROM classify C, artwork_paints AP

WHERE c.title = ap.title AND AP.year > 1950

                GROUP BY c.group_name;
```

Problem 3. Exercise 19.2.

1. Keys for R:
We compute the attribute closure for each attribute and combinations of attributes. If X+ = R+ then X is a key. We also look at the FDs to determine which attributes should or shouldn't be in the key.
A+ = AB
B+ = B
BC+ = BCE
BCD+ = BCDEA
ACD+ = ACBED
CED+ = CEDAB
Keys are: BCD, ACD, CED

2. Is R in 3NF?
Yes. Because all B, E , A are parts of keys.

3. Is R in BCNF?
No. Because neither A, BC, nor ED are keys for R.

Problem 4. Given a relation R with four attributes ABCD and a set of FDs {A → B, BC → D, A → C}, compute F+ by

(a) first computing all attribute closures. A+ is completed for you.

	Attribute closure
A+	ABCD
B+	B
C+	C
D+	D
AB+	ABCD
AC+	ABCD
AD+	ABCD
BC+	BCD
BD+	BD
CD+	CD
ABC+	ABCD
ABD+	ABCD
ACD+	ABCD
BCD+	BCD
ABCD+	ABCD

(b) adding a √ to the table entry if (the column) is in (the row)+. Consult slide #30 (Compute F+) from Lecture 9. The first row is completed for you.

	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
A	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
B		√													
C			√												
D				√											
AB	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
AC	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
AD	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
BC		√	√	√				√	√	√				√	
BD		√		√					√						
CD			√	√						√					
ABC	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
ABD	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
ACD	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
BCD		√	√	√				√	√	√				√	
ABCD	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

(c) What are the candidate key(s) for R?

A

Problem 5. Exercise 19.8 from book.

1) (a) i. {ABC}, FD's are AB->C, AC->B, BC->A

Minimal Cover = AB->C, AC->B, BC->A

ii. BCNF

iii. No need for decomposition since R is in BCNF

(b) i. {ABCD}, FD's are AB->C, AC->B, B->D, BC->A

Minimal Cover = AB->C, AC->B, B->D, BC->A

ii. 1NF

iii. ABC, BD is the BCNF decomposition.

(c) i. {ABCEG}, FD's are AB->C, AC-B, BC-A, E-G

Minimal Cover = AB->C, AC->B, BC->A, E->G

ii. 1NF

iii. ABE, ABC, EG is the BCNF decomposition.

(d) i. {DCEGH}, FD's are E->G

Minimal Cover = E->G

ii. 1NF

iii. DCEH, EG is the BCNF decomposition.

(e) i. {ACEH}, NO FD's exist.

Minimal Cover is also empty (**NOTE**: The correct answer is AC->E, but since this one is a bit difficult, it's OK if you didn't find it).

ii. BCNF (**NOTE**: if you find the minimal cover, then the correct answer for this would be 1NF)

iii. This is already in BCNF (**NOTE**: if you find the minimal cover, then the correct answer for this would be ACE and ACH}.

2) (a) This is not lossless-join and dependencies such as AB→C (and others in the ABC "trio") are not preserved.

(b) This is lossless decomposition. Because if we take the join of R1= ABC, and R2= ACDE: this is lossless because R1 ∩ R2 = AC, and AC→ B, AC→R1 (ABC). Then if we take ACDE with ADG, the intersection is AD, and AD→G, so AD → R2 (ADG). Therefore this is lossless decomposition. However, the dependencies {B→D, E→G} are not preserved.