Practice Final Exam Review

Question 1

. Using the relational algebra operators (set semantics) and relations $R(A, B), S(A, B)$ $R \bowtie S = (R \cup S) \cap (S \cup R)$.

False

In materialized views, it is always possible to synchronize updates between the base table and the view.

False

In a 3NF decomposition, we are guaranteed dependency preservation and lossless joins.

True

Conflict serializable schedules are recoverable and avoid cascading aborts.

False

Question 1

In the wound-wait policy, higher priority transactions always receive the requested lock(s) immediately.

True

Question 2

Consider the two relational instances, *Emps* and *WorksIn*. Compute the result of the following query on these instances.

Emps:

Emp	Sal	Mgr
Xin	100	Pat
Jen	200	Jen
Pat	100	Meg
Meg	150	Meg
Joe	110	Ami
Ami	200	Bob

WorksIn:

Dept	Emp
Toy	Xin
Car	Bob
Car	Meg
Art	Pat
Art	Meg

```
SELECT *
FROM Emps,
    (select Emps.emp, max(sal) as sal
    from Emps, WorksIn
    where Emps.emp = WorksIn.emp
    group by Emps.emp) as foo
WHERE Emps.mgr = foo.emp
```

Solution:

```
emp | sal | mgr | emp | sal
-----+-----+-----+-----+-----
 Xin | 100 | Pat | Pat | 100
 Pat | 100 | Meg | Meg | 150
 Meg | 150 | Meg | Meg | 150
(3 rows)
```

Question 3

(a) Write an SQL query that returns the names of all managers, along with the highest number of employees they have managed in any single company (and the name of that company). For example, if manager Fred has managed two employees at IBM and three at Microsoft, then the query should return "(Fred, 3, Microsoft)". If there are ties (for example, Fred managed three people at IBM and three people at Microsoft), then a manager may be returned more than once (for this example, two tuples "(Fred, 3 Microsoft)" and "(Fred, 3, IBM)" would be returned).

– eid: unique employee id
– name: employee's name
– city: employee's city of residence
Employee(eid,name,city)

– cid: unique company id
– name: company name
– city: city where company is located
Company(cid,name,city)

– eid: foreign key of Employee
– cid: foreign key of Company
– an employee can work for a company at
– salary: employee's salary for the said con
WorksFor(eid,cid,salary)

– eid: foreign key of Employee
– cid: foreign key of Company
– mid: employee id of eid's manager (foreign key of Employee)
Manages(eid,cid,mid)

Solution:

```
SELECT E.name, count(M.eid), C.name
   FROM Manages M, Employee E, Company C
   WHERE M.mid = E.eid and M.cid = C.cid
   GROUP BY M.mid, M.cid, E.name, C.name
   HAVING count(M.eid) >= ALL (SELECT count(M2.eid)
                               FROM Manages M2
                               WHERE M2.mid = M.mid
                               GROUP BY M2.mid, M2.cid) ;
```

Question 4

Consider the following schema (keys are underlined):

Product(<u>pid</u>, name, type, mfgr, price), Buys(<u>cid</u>, <u>pid</u>), Customer(<u>cid</u>, cname, age, gender)

a) Write the following query in relational algebra: *Find the names of all customers who have purchased a product manufactured by 'Apple'.*

Solution:

- $\Pi_{cname}(Customer \bowtie (Buys \bowtie (\Pi_{pid}(\sigma_{mfgr=`Apple'}Product))))$

Question 4(b)

Consider the following schema (keys are underlined):

Product(<u>pid</u>, name, type, mfgr, price), Buys(<u>cid</u>, <u>pid</u>), Customer(<u>cid</u>, cname, age, gender)

b) Write the following query in SQL: *Find the names of all customers who have not purchased the most expensive product.*

-
  ```
  (SELECT cname
  FROM Customer)

  EXCEPT

  (SELECT cname
  FROM Product, Buys, Customer
  WHERE Product.pid = Buys.pid AND Customer.cid = Buys.cid AND
  price >= ALL (SELECT price
               FROM Product)
  ```

Question 5

Consider the following schema for tracking customers' underlined.

- Books(ISBN, title, author, year, length)
  ISBN is a string that is used internationally for ide
- Authors(AID, name)
- Customers(CID, name)
- Ratings(ISBN, CID, rating)
  Rating is an integer.

The following foreign key (FK) constraints hold:

- Books(author) is FK to Authors(AID)
- Ratings(ISBN) is FK to Books(ISBN)
- Ratings(CID) is FK to Customers(CID)

Using the relational algebra operators $\Pi, \sigma, \bowtie, \times, \cap, \cup, -, \rho, :=$, write the following queries, assuming set semantics.

(a) Find the CID of every customer who has rated every book by author "Tolkien".

Solution:

$Shouldhave := (\Pi_{CID} Customers) \times (\Pi_{ISBN} \sigma_{author=\text{"Tolkien"}} Books)$

$Dohave := \Pi_{CID,ISBN} Ratings$

$Missedsome := Shouldhave - Dohave$

$Answer := \Pi_{CID} Customer - \Pi_{CID} Missedsome$

Question 5(b)

- Books(<u>ISBN</u>, title, author, year, length)
  ISBN is a string that is used internationally for identifying books.
- Authors(<u>AID</u>, name)
- Customers(<u>CID</u>, name)
- Ratings(<u>ISBN, CID</u>, rating)
  Rating is an integer.

(b) Find the name of every author who has published exactly two books before the year 2000, and none since 2000 (*i.e.*, none in the year 2000 or later).

Solution:

$Oldbooks := \sigma_{year<2000} Books$

$Twoplus := \Pi_{author} \sigma_{B1.ISBN \neq B2.ISBN \wedge B1.author=B2.author} (\rho_{B1} Oldbooks \times \rho_{B2} Oldbooks)$

$Threeplus :=$

$$\Pi_{author} \sigma_{B1.ISBN \neq B2.ISBN \wedge B2.ISBN \neq B3.ISBN \wedge B1.ISBN \neq B3.ISBN \wedge B1.author=B2.author \wedge B2.author=B3.author} (\rho_{B1} Oldbooks \times \rho_{B2} Oldbooks \times \rho_{B3} Oldbooks)$$

$Exactlytwo := Twoplus - Threeplus$

$Answer := Exactlytwo - \Pi_{author}(\sigma_{year \geq 2000} Books)$

Question 6

R:

A	B
1	2
3	4
1	3

S:

B	C
1	3
2	4

T:

B	C
2	5
2	3

V:

E	F
1	4
2	5

(a) What is the result of the following query (assuming set semantics)? Provide the resulting schema and result tuples.

$$\mathbf{V} \times (\pi_{\mathsf{B}}(\mathbf{R} \bowtie \mathbf{S}) - \pi_{\mathsf{B}}(\mathbf{T}))$$

Solution:

(empty table)

B	E	F

Q6 (b)

R:

A	B
1	2
3	4
1	3

S:

B	C
1	3
2	4

T:

B	C
2	5
2	3

V:

E	F
1	4
2	5

(b) What is the result of the following query (assuming bag semantics)? Provide the resulting schema and tuples.

$$\pi_{\mathrm{A}}(\mathrm{R}) - \pi_{\mathrm{A}}(\rho_{\mathrm{A}\leftarrow\mathrm{B}}((\mathrm{S})))$$

Solution:

A
1
3

Question 7

Question 7. Illustrate with an example how views can be used for security. Your example should include a table, a view defined over the table, and a brief description of how the view hides some information from an unauthorized user.

Q8

Consider relation R(A,B,C,D) and functional dependencies

$$\{ABC \rightarrow D,\ CD \rightarrow A,\ CA \rightarrow B,\ AD \rightarrow C,\ CD \rightarrow B\}.$$

a) List all the keys of R

AD, AC, CD

(b) Given a minimal cover of $\{AC \rightarrow D, AC \rightarrow B, CD \rightarrow A, AD \rightarrow C\}$, compute a 3NF decomposition of R.

3NF: R1(ACD), R2(ACB)

c) Is your decomposition dependency preserving?

Yes

Q8 continued

(d) Consider relation R(A,B,C,D) and functional dependencies $\{D \rightarrow B,\ C \rightarrow A,\ A \rightarrow B\}$.

Is the decomposition of R into R1(A,B), R2(A,C) and R3(A,D) a lossless join decomposition? If not, give an example instance of R where the join of R1, R2, and R3 is not R.

No, it is not the case that $A \rightarrow C$ nor $A \rightarrow D$

(e) Is the decomposition of R into R1(A,B), R2(A,C) and R3(A,D) dependency preserving?

No, $D \rightarrow B$ is lost.

Q9

Question 9. Consider the relation R with schema R(A,B,C,D), and functional dependencies $\{AB \to C, C \to D, D \to A\}$.

(a) Is the relation in BCNF? Is it in 3NF? Explain why or why not.

(b) Is the decomposition of R into AB, BC and CD lossless? Why or why not?

(c) Is the decomposition of R into AB, BC and CD dependency preserving? Why or why not?

a) The keys are AB, BC, BD (by augmentation of FDs). Hence, the latter two dependencies violate BCNF but not 3NF (since all attributes are prime). R is in 3NF not BCNF.

b) No, since B is not a key for AB nor BC.

c) No, AB→C is lost.

Q10

Prove the following inference rule for functional dependencies using only Armstrong's axioms:
If $P \rightarrow QR$ and $R \rightarrow S$, then $P \rightarrow QS$.

Show the steps of your proof, and indicate which of Armstrong's axioms is applied in each step.

Derived in class.

Q11

- For each schedule indicate which properties hold.

a) $\boxed{R_1(A), W_2(A)}, \boxed{R_1(B)}, Commit_1, W_3(B), R_3(B), W_3(A), Commit_3, R_2(C), Commit_2.$

Conflict serializable	Yes
ACA	Yes
Recoverable	Yes
2PL	No
Strict 2PL	No

Q11

- For each schedule indicate which properties hold.

b) $R_1(A), \boxed{W_2(B)}, \boxed{R_1(B)}, \boxed{Commit_1, Commit_2}$

Conflict serializable	Yes
ACA	No
Recoverable	No
2PL	Yes
Strict 2PL	No

c) $R_1(A), W_2(B), R_1(B), Commit_2, Commit_1$

Same as above except this schedule is recoverable.

Q12 (see practice final for full details)

- Give the sequence of lock requests using Wound-Wait policy.

(a) $R_1(X), W_2(Y), W_2(X), W_3(Y), W_1(Y)$

  - T1 acquires shared lock on X; T2 acquires an exclusive lock on Y.
  - T2 requests an exclusive lock on X. Since T2 is lower priority, it will wait.
  - T3 requests an exclusive lock on Y; it also waits.
  - T1 now requests an exclusive lock on Y; since it has the higher priority than T2, it will abort T2.

Q12

- Given the following actions, under strict 2PL with deadlock detection, is there a deadlock? If so, show the WFG.

$$R_1(X), W_2(Y), W_2(X), W_3(Y), W_1(Y), Commit_1, Commit_2, Commit_3$$

- Yes, deadlock exists.