Practice Final Exam Review

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

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.

Conflict serializable schedules are recoverable and avoid cascading aborts.

False

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

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

Emp

Xin

Bob

Meg

 Pat Meg

ſ
2
ŗ
i
)
Ľ

Dept WorksIn:

Toy Car

Car

 Art

Art

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:

(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).

```
Solution:
- eid: unique employee id
- name: employee's name
- city: employee's city of residence
Employee(eid,name,city)
                                       SELECT E.name, count(M.eid), C.name
                                          FROM Manages M, Employee E, Company C
- cid: unique company id
                                           WHERE M.mid = E.eid and M.cid = C.cid
- name: company name
- city: city where company is located
                                           GROUP BY M.mid, M.cid, E.name, C.name
Company(\underline{cid}, name, \underline{city})
                                           HAVING count(M.eid) >= ALL (SELECT count(M2.eid)
                                                                                FROM Manages M2
- eid: foreign key of Employee
- cid: foreign key of Company
                                                                                WHERE M2.mid = M.mid
- an employee can work for a company at
                                                                                GROUP BY M2.mid, M2.cid);
- 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)
```

Consider the following schema (keys are underlined):

Product(pid, name, type, mfgr, price), Buys(cid, pid), Customer(cid, 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(pid, name, type, mfgr, price), Buys(cid, pid), Customer(cid, 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
```

Consider the following schema for tracking customers' underlined.

- Books(<u>ISBN</u>, title, author, year, length) ISBN is a string that is used internationally for ide
- Authors(AID, name)
- Customers(<u>CID</u>, 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:

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

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

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(AID, name)
- Customers(<u>CID</u>, name)
- Ratings(<u>ISBN</u>, <u>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:

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

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

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

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

R: A B

1 2
3 4
1 3

S: B C 1 3 2 4

 $T: \begin{array}{|c|c|c|}\hline B & C \\\hline 2 & 5 \\\hline 2 & 3 \\\hline \end{array}$

 $V: \begin{array}{|c|c|}\hline E & F \\\hline 1 & 4 \\\hline 2 & 5 \\\hline \end{array}$

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

$$V \times (\pi_B(R \bowtie S) - \pi_B(T))$$

Solution:

(empty table) B E F

Q6 (b)

$$T: \begin{array}{|c|c|c|}\hline B & C \\\hline 2 & 5 \\\hline 2 & 3 \\\hline \end{array}$$

$$V: \begin{array}{c|c} \mathbf{E} & \mathbf{F} \\ \hline 1 & 4 \\ \hline 2 & 5 \\ \hline \end{array}$$

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

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

Solution:

A

1

3

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.

 $\begin{array}{c} \text{Consider relation R(A,B,C,D) and functional dependencies} \\ \{ABC \rightarrow D,\ CD \rightarrow A,\ CA \rightarrow B,\ AD \rightarrow C,\ CD \rightarrow B\}. \end{array}$

a) List all the keys of R

AD, AC, CD

(b) Given a minimal cover of $\{AC \to D, AC \to B, CD \to A, AD \to C\}$, 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 $\{D \to B,\ C \to A,\ A \to B\}$.

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→B is lost.

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

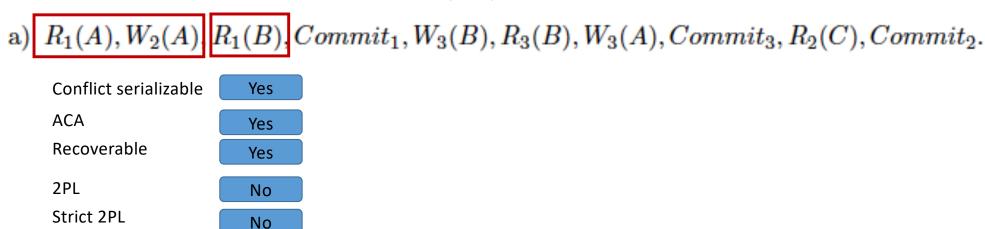
- (a) Is the relation in BCNF? Is it in 3NF? Explain why or why not.
- (b) Is the decomposition of R into AB, BC and CD lossless? Why or why not?
- (c) Is the decomposition of R into AB, BC and CD dependency preserving? Why or why not?
- a) The keys are AB, BC, BD (by augmentation of FDs). Hence, the 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 \rightarrow C is lost.

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

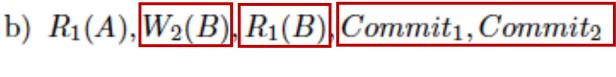
Show the steps of your proof, and indicate which of Armstrong's axioms is applied in each step.

Derived in class.

• For each schedule indicate which properties hold.



• For each schedule indicate which properties hold.





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.

• 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.