CS 348: Assignment 5

Question 1.

1. Use Armstrong's axioms to prove the soundness of the union and decomposition rules.

(*union rule*)

1.	$X \to Y$	(given)
2.	$X \to Z$	(given)
3.	$X \to XY$	(1 and augmentation with X)
4.	$XY \to YZ$	(2 and augmentation with Y)
5.	$X \to YZ$	(3, 4 and transitivity)

(*decomposition rule*)

1.	$X \to YZ$	(given)
4.	$YZ \to Y$	(reflexivity)
5.	$X \to Y$	(1, 2 and transitivity)

Question 2.
Prove that if $A \notin ComputeX^+(X, F)$ then $X \to A \notin F^+$ (you must prove this without using/referring to the *Correctness theorem* for $ComputeX^+(X, F)$ introduced in Module 9, slide 17).

Consider $R = (X_1, \ldots, X_k, B_1, \ldots, B_n, A, A_1, \ldots, A_m)$ and $X = \{X_1, \ldots, X_k\}$ and $ComputeX^+(X, F) = \{X_1, \ldots, X_k, B_1, \ldots, B_n\}$.

Let a_i, b_i, and c_i be pairwise distinct constants. Then a database instance (consisting of two tuples) $R = \{(c_1, \ldots, c_{k_n}, a_1, \ldots, a_m), (c_1, \ldots, c_{k_n}, b_1, \ldots, b_m)\}$ satisfies F and falsifies $X \to A$. Hence $X \to A \notin F^+$.

To show that this instance satisfies F: assume $Y \to Z \in F$ is violated by the instance above. Then $Y \subset \{X_1, \ldots, X_k, B_1, \ldots, B_n\}$ and $Z \cap \{A_1, \ldots, A_m\} \neq \emptyset$. This is contradiction with respect to the definition of $ComputeX^+$ (in particular the termination condition).

Question 3.

Translate each of the following SQL queries over the relational database schema for bibliography information given on Slide 10 of Module 2 to formulations in the relational algebra with multiset semantics defined in Module 10.

1. *All book titles.*

```
select distinct title
from publication, book
where publication.pubid = book.pubid
```

$$\mathsf{elim}\ \pi_{\#2}(\sigma_{\#1=\#3}(\texttt{PUBLICATION} \times \texttt{BOOK}))$$

---

2. *All publications with at least two authors.*

```
select distinct r1.publication
from wrote r1, wrote r2
where r1.publication = r2.publication
and not r1.author = r2.author
```

$$\mathsf{elim}\ \pi_{\#2}(\sigma_{\#2=\#4}(\texttt{WROTE} \times \texttt{WROTE}) - \sigma_{\#1=\#3}(\texttt{WROTE} \times \texttt{WROTE}))$$

NOTE: Either semantics for set difference can be assumed.

---

3. *All author-publication ids for all publications except books and journals.*

```
select * from wrote
where publication not in (
   (select pubid from book)
   union
   (select pubid from journal))
```

$\texttt{WROTE} - (\pi_{\#1,\#2}(\sigma_{\#2=\#3}(\texttt{WROTE} \times \texttt{BOOK})) \cup \pi_{\#1,\#2}(\sigma_{\#2=\#3}(\texttt{WROTE} \times \texttt{JOURNAL})))$

NOTE: Either semantics for set difference can be assumed. Also, execution with the alternative semantics for set difference will be fully pipelined with no need for an unbounded amount of temporary store.

Question 4.
Assume the DBA group has chosen the following standard physical design (as defined in Module 10) for the relational database schema for bibliography information given on Slide 10 of Module 2:

- A primary index called T-A is chosen for each table T, where a is the attribute name of the first column of T. The chosen data structure in each case is a Btree on a. For example, the primary index of table `WROTE` is called `WROTE-AUTHOR` and is a Btree on `author`, and, for querying, has arity 3:

  `WROTE-AUTHOR / (rid, author, publication).`

  Like all indices, recall that attribute `rid` is prepended to the list of attributes for `WROTE-AUTHOR`.

- A secondary index called `WROTE-PUBLICATION` is chosen for table `WROTE` that is a Btree on attribute `publication`, and, for querying, also has arity 3

  `WROTE-PUBLICATION / (rid, publication, wrote-author-rid).`

  Recall that there is a foreign key on attribute `wrote-author-rid` to attibute `rid` of primary key table `WROTE-AUTHOR`.

- A secondary index called `ARTICLE-APPEARS-IN` is chosen for table `ARTICLE` that is a Btree on attribute `appears-in`, and, for querying, has arity 3

  `ARTICLE-APPEARS-IN / (rid, appears-in, author-pubid-rid).`

Translate each of the following queries in SQL to an RA expression over this physical design, that is, in which all relation names are names of indices. Your RA expression should be as efficient as possible, assuming the implementation of operations and simple cost model given in Module 10. For example, you should consider using nested index scans. Recall from Slide 39 of Module 10 that these are expressed with the syntax

$$E \times \sigma_{\#i=\#j\ell}(R),$$

where $\#j$ refers to a column of a subexpression E and where R is an index for which $\#i$ is its `rid` column or its search key column (i.e., either column $\#1$ or $\#2$ for the primary index `WROTE-AUTHOR`).

1. *Names of all authors of books published in 2019 last year.*

```
select distinct name
from AUTHOR a
where exists (
   select * from WROTE w, BOOK b
   where w.publication = b.pubid and w.author = a.aid
   and b.year = 2019 )
```

$$
\begin{array}{l}
\mathsf{elim}\ \pi_{\#3}(\\
\quad \pi_{\#2}(\\
\quad\quad \pi_{\#3}(\pi_{\#2}(\sigma_{\#4=\#5}(\texttt{BOOK-PUBID} \times 2019)) \\
\quad\quad\quad\quad \times \sigma_{\#2=\#1\ell}(\texttt{WROTE-PUBLICATION})) \\
\quad\quad \times \sigma_{\#1=\#1\ell}(\texttt{WROTE-AUTHOR})) \\
\quad \times \sigma_{\#2=\#1\ell}(\texttt{AUTHOR-AID}))
\end{array}
$$

---

2. *All publication identifiers of articles appearing in volume 1 number 1 of the journal ACM TODS.*

```
select distinct pubid
from ARTICLE a
where exists (
   select * from JOURNAL j, PUBLICATION p
   where j.pubid = p.pubid and p.title = 'ACM TODS'
   and j.volume = 1 and j.number = 1
   and a.appears-in = j.pubid )
```

$$
\begin{array}{l}
\pi_{\#3}(\\
\quad \pi_{\#3}(\\
\quad\quad \pi_{\#3}(\sigma_{\#4=\#5}((\pi_{\#2}(\sigma_{\#4=\#7}(\sigma_{\#3=\#6}(\texttt{JOURNAL-PUBID} \times 1) \times 1)) \\
\quad\quad\quad\quad\quad\quad\quad\quad \times \sigma_{\#2=\#1\ell}(\texttt{PUBLICATION-PUBID})) \\
\quad\quad\quad\quad\quad\quad\quad\quad \times \texttt{'ACM TODS'}) \\
\quad\quad \times \sigma_{\#2=\#1\ell}(\texttt{ARTICLE-APPEARS-IN})) \\
\quad \times \sigma_{\#1=\#1\ell}(\texttt{ARTICLE-PUBID}))
\end{array}
$$

Note: Primary and foreign keys imply an elim operation is not required.

Question 5.
Determine whether or not each of the following four transaction execution schedules is conflict serializable. If a schedule is conflict serializable, specify a serial order of transaction execution to which it is equivalent.

$$S_1 = r_1[x], r_2[y], w_2[x], r_1[z], r_3[z], w_3[z], w_1[z]$$
$$S_2 = w_1[x], w_1[y], r_2[u], w_2[x], r_2[y], w_2[y], w_1[z]$$
$$S_3 = w_1[x], w_1[y], r_2[u], w_1[z], w_2[x], r_2[y], w_1[u]$$
$$S_4 = w_1[x], w_2[u], w_2[y], w_1[y], w_3[x], w_3[u], w_1[z]$$

- S_1 is not conflict serializable.
- S_2 is conflict serializable with only the serial schedule of operations for transactions in the order T_1 then T_2.
- S_3 is not conflict serializable.
- S_4 is conflict serializable with only the serial schedule of operations for transactions in the order T_2 then T_1 then T_3.

Question 6.
Suppose that after a system failure, the transaction log looks as shown below (the log tail is at the bottom). Describe what the database system must do to recover from the system failure. Indicate which objects must be modified, and in what order those modifications occur. Indicate which transactions are committed and which are aborted after the failure recovery is complete.

T_1, BEGIN
T_1, UNDO, x, 0
T_1, REDO, x, 10
T_2, BEGIN
T_2, UNDO, y, 10
T_2, REDO, y, 20
T_3, BEGIN
T_1, COMMIT
T_3, UNDO, x, 10
T_3, REDO, x, 400
T_4, BEGIN
T_3, UNDO, z, 0
T_3, REDO, z, 100
T_5, BEGIN
T_4, UNDO, a, 0
T_4, REDO, a, 1
T_6, BEGIN
T_4, ABORT
T_5, UNDO, a, 0
T_5, REDO, a, 2
T_3, COMMIT
T_6, UNDO, x, 400
T_6, REDO, x, 0
T_5, COMMIT
T_6, UNDO, a, 2
T_6, REDO, a, 3

- A simple recovery from failure can proceed as follows:

  Step 1. Scan the log from the tail to the head remembering which transactions committed and applying the UNDO logs.

  Step 2. Scan the log from the head to the tail applying the REDO logs for committed transactions.

  One improvement is to not apply UNDO logs for committed transactions.

- Based on the simple recovery scheme, objects would be updated as follows:

  (during the first step)

  1. $a \leftarrow 2$
  2. $x \leftarrow 400$
  3. $a \leftarrow 0$
  4. $a \leftarrow 0$
  5. $z \leftarrow 0$
  6. $x \leftarrow 10$
  7. $y \leftarrow 10$
  8. $x \leftarrow 0$

  (during the second step)

  9. $x \leftarrow 10$
  10. $x \leftarrow 400$
  11. $z \leftarrow 100$
  12. $a \leftarrow 2$

  Updates 3, 5, 6 and 8 during the first step would not take place with the improvement.

- After the failure recovery is completed, transactions T_1, T_3 and T_5 are committed and transactions T_2, T_4 and T_6 are aborted.
