CSCI 5333.3 DBMS

Fall 2021

Suggested Solution to Final Examination

(1) For example, minimally documented:

```
    if (array_key_exists('filmid1', $_GET)) {
        $filmId_1 = $_GET['filmid1'];
    }
    if (array_key_exists('filmid2', $_GET)) {
        $filmId_2 = $_GET['filmid2'];
    }

    //  Get film title and rating
    $query = <<<__QUERY
SELECT DISTINCT r1.customer_id
FROM rental AS r1 INNER JOIN inventory AS i1 ON (r1.inventory_id =
i1.inventory_id)
      INNER JOIN rental AS r2 ON (r1.customer_id = r2.customer_id)
      INNER JOIN inventory AS i2 ON (r2.inventory_id = i2.inventory_id)
WHERE i1.film_id = ?
AND i2.film_id = ?;
__QUERY;

if ($stmt = $mysqli->prepare($query)) {
    $stmt->bind_param('ss', $filmId_1, $filmId_2);
    $stmt->execute();
    $stmt->bind_result($customerId);
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        echo "<p>The two films (id: $filmId_1 and $filmId_2)";
        echo " have both been rented by the following customers
(id)\n</p>\n<ol>\n";
        while ($stmt->fetch()) {
            echo "<li>$customerId</li>\n";
        }
        echo "</ol>";
      }
      else {
        echo "<p>The two films (id: $filmId_1 and $filmId_2)";
        echo " have not been both rented by any customer.\n</p>\n<ul>\n";
      }
    $stmt->free_result();
}

$mysqli->close();
```

(2)

(a) For example:

```
 DROP VIEW f21t2;

CREATE VIEW f21t2 AS
SELECT DISTINCT fa.actor_id,
    COUNT(DISTINCT fa.film_id) as filmCount,
    COUNT(*) as copyCount
FROM film_actor AS fa INNER JOIN inventory AS i ON (fa.film_id = i.film_id )
GROUP BY fa.actor_id;
```

(b) For example:

```
SELECT DISTINCT *
FROM f21t2
WHERE filmCount = (SELECT MIN(filmCount) FROM f21t2)
ORDER BY filmCount ASC;
```

(c) For example:

```
DROP FUNCTION f21f1;

DELIMITER //

CREATE FUNCTION f21f1(categoryId TINYINT UNSIGNED) RETURNS INT
BEGIN
   DECLARE result INT DEFAULT 0;

   SELECT COUNT(*) INTO result
   FROM film_category fc INNER JOIN inventory i
            ON (fc.film_id = i.film_id)
   WHERE fc.category_id = categoryId;

   RETURN result;
END //

DELIMITER ;

SELECT f21f1(2);
```

(3) To show F = {A->B, D->E. AB->C, AC->D} implies A->E

Proof: For example:

[1] A->B (given)
[2] AB->C (given)

[3] A->C (pseudo-transitivity on [1] and [2], and simplification)
[4] AC->D (given)
[5] A->D (pseudo-transitivity on [3] and [4], and simpllification)
[6] D->E (given)
[7] A-> E (transitivity on [5] and [6])
QED.

(4) Yes, the decomposition is lossless

Given {A->C, CD->B, BC->AD, E->B} and R is decomposed into R1(A,B,C), R2(C,D,E) and R3(B,E).

You can use the algorithm for checking for lossless decomposition below.

Step 1. Create a table of 5 columns (number of columns and 3 rows (number of relations). Populate it with b(i,j).

Relation	A	B	C	D	E
R1	b(1,1)	b(1,2)	b(1,3)	b(1,4)	b(1,5)
R2	b(2,1)	b(2,2)	b(2,3)	b(2,4)	b(2,5)
R3	b(3,1)	b(3,2)	b(3,3)	b(3,4)	b(3,5)

Step 2. For each relation Ri, set all attribute Aj that appears in Ri from b(i,j) to a(j).

Relation	A	B	C	D	E
R1	*a(1)*	*a(2)*	*a(3)*	b(1,4)	b(1,5)
R2	b(2,1)	b(2,2)	*a(3)*	*a(4)*	*a(5)*
R3	b(3,1)	*a(2)*	b(3,3)	b(3,4)	*a(5)*

Step 3. For each FD X-> Y, if two rows have the common X values, for every attribute W in Y:

- If one cell is an a and the other cell is an b, change the b to the a.
- If both cells are b's, change them to the same b.

Applying A->C: no change.

Applying CD->B: no change.

Applying BC->AD: no change.

Applying E->B

Relation	A	B	C	D	E
R1	a(1)	a(2)	a(3)	b(1,4)	b(1,5)
R2	b(2,1)	*a(2)*	a(3)	a(4)	a(5)

R3	b(3,1)	a(2)	b(3,3)	b(3,4)	a(5)

Applying A->C: no change.

Applying CD->B: no change.

Applying BC->AD:

Relation	A	B	C	D	E
R1	a(1)	a(2)	a(3)	*a(4)*	b(1,5)
R2	*a(1)*	a(2)	a(3)	a(4)	a(5)
R3	b(3,1)	a(2)	b(3,3)	b(3,4)	a(5)

Since the second row is now composed of only a's, the algorithm stops and pronounces that the decomposition is lossless.

(5)

(a)	T	(b)	T	(c)	T	(d)	F	(e)	T
(f)	F	(g)	T	(h)	T	(i)	F	(j)	T

(6) {B->A, AB->C, CD->A, D->B, BC->E, E->FA}

(a) A+ = A, B+ = ABCEF, C+ = C, D+ = ABCDEF, E+=AEF, F+ =F

(b) The candidate key is D. Prime: D. Non-prime: ABCEF.

(c) For example: {B->CE, D->B, E->AF}

(d) 2NF. The FD B->C violates 3NF as B is not a superky and C is non-prime.

(e) Yes, the following decomposition satisfies all requirements:

R1(B,C,E) {B->CE}
R2(B,D) {D->B}
R3(A,E,F) {E->AF}