

CSC343 Worksheet 3 Solution

June 16, 2020

1. Exercise 6.1.1:

If there is a comma between A and B (i.e, *SELECT A, B*), we can conclude A and B are two different attributes.

If there are no commas between A and B , we can conclude B is an alias of A .

2. Exercise 6.1.2:

- a) *SELECT* address *FROM* Studio *WHERE* name = 'MGM';
- b) *SELECT* birthdate *FROM* MovieStar *WHERE* name = 'Sandra Bullock';
- c) *SELECT* starName *FROM* StarsIn *WHERE* movieYear = 1980, movieTitle *LIKE* '%Love%';

Correct Solution:

```
SELECT starName FROM StarsIn WHERE movieYear = 1980 AND movieTitle  
LIKE '%Love%';
```

- d) *SELECT* name *FROM* MovieExec *WHERE* netWorth >= 10000000;
- e) *SELECT* name *FROM* MovieStar *WHERE* gender='male' OR address *LIKE* '%Malibu%';

3. Exercise 6.1.3:

- a) *SELECT* model, speed, hd *FROM* PC *WHERE* price < 1000;
- b) *SELECT* model, speed AS gigahertz, hd AS gigabytes *FROM* PC *WHERE* price < 1000;
- c) *SELECT* maker *FROM* Product *WHERE* type='printer';
- d) *SELECT* model, ram, screen *FROM* Laptops *WHERE* price > 1500;
- e) *SELECT* * *FROM* Printer *WHERE* color=TRUE;

f) `SELECT model, hd FROM PC WHERE speed = 3.20 AND price < 2000;`

4. **Exercise 6.1.4:**

- a) `SELECT class, country FROM Classes where numGuns >= 10;`
- b) `SELECT name AS shipName FROM Ships WHERE launched < 1918;`
- c) `SELECT ship, battle FROM Outcomes WHERE result='sunk';`
- d) `SELECT name FROM Ships WHERE name = class;`
- e) `SELECT name FROM Ships WHERE name LIKE 'R%';`
- f) `SELECT name FROM ships WHERE name LIKE '% % %';`

5. **Exercise 6.1.5:**

- a) Given $a = 10$, the sets of tuples that satisfy the condition is

$(10, -MAX_INT), (10, -MAX_INT + 1), \dots, (10, 0), \dots, (10, MAX_INT - 1),$
 $(10, MAX_INT), (10, NULL)$

Given $b = 20$, the sets of tuples that satisfy the condition is

$(-MAX_INT, 20), (-MAX_INT + 1, 20), \dots, (0, 20), \dots, (MAX_INT - 1, 20),$
 $(MAX_INT, 20), (NULL, 20)$

Given $a = 10$ and $b = 20$, the set of tuple that satisfy the condition is $(10, 20)$

- b) Given $a = 10$ AND $b = 20$, the only set of (a, b) tuple that satisfy the condition is $(10, 20)$.
- c) There are three cases to consider
 - i. $a < 10$

In this case, the set of (a, b) tuples that satisfy the condition is:

$(9, -MAX_INT), (9, -MAX_INT + 1), \dots, (9, 0), \dots, (9, MAX_INT - 1),$
 $(9, MAX_INT), (9, NULL)$

$(8, -MAX_INT), (8, -MAX_INT + 1), \dots, (8, 0), \dots, (8, MAX_INT - 1),$
 $(8, MAX_INT), (8, NULL)$

...

$(-MAX_INT + 1, -MAX_INT), (-MAX_INT + 1, -MAX_INT + 1),$
 $\dots, (-MAX_INT + 1, 0), \dots, (-MAX_INT + 1, MAX_INT - 1),$
 $(-MAX_INT + 1, MAX_INT), (-MAX_INT + 1, NULL)$

$(-MAX_INT + 1, -MAX_INT), (-MAX_INT + 1, -MAX_INT + 1),$
 $\dots, (-MAX_INT + 1, 0), \dots, (-MAX_INT + 1, MAX_INT - 1),$
 $(-MAX_INT + 1, MAX_INT), (-MAX_INT + 1, NULL)$

ii. $a \geq 10$

In this case, the set of (a, b) tuples that satisfy the condition is:

$(10, -MAX_INT), (10, -MAX_INT + 1), \dots, (10, 0), \dots, (10, MAX_INT - 1),$
 $(10, MAX_INT), (10, NULL)$

$(11, -MAX_INT), (11, -MAX_INT + 1), \dots, (11, 0), \dots, (11, MAX_INT - 1),$
 $(11, MAX_INT), (11, NULL)$

...

$(MAX_INT - 1, -MAX_INT), (MAX_INT - 1, -MAX_INT + 1),$
 $\dots, (MAX_INT - 1, 0), \dots, (MAX_INT - 1, MAX_INT - 1),$
 $(MAX_INT - 1, MAX_INT), (MAX_INT - 1, NULL)$

$(MAX_INT, -MAX_INT), (MAX_INT, -MAX_INT + 1),$
 $\dots, (MAX_INT, 0), \dots, (MAX_INT, MAX_INT - 1),$
 $(MAX_INT, MAX_INT), (MAX_INT, NULL)$

iii. $a < 10$ AND $a \geq 10$

This case is not considered. No (a, b) tuples match this condition.

d) In this case the set of (a, b) tuples that satisfy this condition is

$(-MAX_INT, -MAX_INT), (-MAX_INT + 1, -MAX_INT + 1),$
 $\dots, (0, 0), \dots, (MAX_INT - 1, MAX_INT - 1),$
 (MAX_INT, MAX_INT)

Here, the case $a = NULL$ and $b = NULL$ is not considered, since $NULL \neq NULL$.

Notes:

- $NULL = NULL$ is $NULL$.

e) In this case, the set of (a, b) tuples that satisfy this condition is

$(-MAX_INT, -MAX_INT), (-MAX_INT, -MAX_INT + 1),$
 $\dots, (-MAX_INT, MAX_INT - 1),$
 $(-MAX_INT, MAX_INT),$

$(-MAX_INT + 1, -MAX_INT + 1), (-MAX_INT + 1, -MAX_INT + 2),$
 $\dots, (-MAX_INT + 1, MAX_INT - 1),$
 $(-MAX_INT + 1, MAX_INT),$

...

$(MAX_INT - 1, MAX_INT - 1), (MAX_INT - 1, MAX_INT),$
 (MAX_INT, MAX_INT)

Here, the case $a = NULL$ OR $b = NULL$ is not considered, since $a \not\leq b$.

6. SELECT * FROM Movies WHERE length;
7. (a) SELECT StarsIn.starName FROM StarsIn, MovieStar WHERE
StarsIn.starName = MovieStar.name AND MovieStar.gender = 'male';
 (b) SELECT StarsIn.starName FROM Movies, StarsIn WHERE
StarsIn.movieTitle = Movies.title AND Movies.studioName = 'MGM';
 (c) SELECT MovieExec.name FROM MovieExec, Studio WHERE MovieExec cert# =
studio.presC# AND Studio.name = 'MGM';
 (d) SELECT M2.title FROM Movies AS M1, Movies AS M2 WHERE
M1.title = "Gone With the Wind" AND M2.length > M1.length;
 (e) SELECT Mx2.name FROM MovieExec AS Mx1, MovieExec AS Mx2 WHERE
Mx1.name = 'Merg Griffin' AND Mx2.netWorth > Mx1.netWorth;
8. a) SELECT Product.maker, Laptop.speed FROM Product, Laptops WHERE
Product.type = 'laptop' AND Laptop.hd >= 30;
 b) (SELECT model, price FROM PC INNER JOIN Product ON
PC.model = Product.model WHERE maker = 'B')

UNION

(SELECT model, price FROM Printer INNER JOIN Product ON
Printer.model = Product.model WHERE maker = 'B')

UNION

(SELECT model, price FROM Laptop INNER JOIN Product ON
Laptop.model = Product.model WHERE maker = 'B')

- c) (SELECT maker FROM Product WHERE type='laptop') -
(SELECT maker FROM Product WHERE type='pc')
- d) SELECT pc1.hd FROM PC AS pc1, PC AS pc2 WHERE
pc1.model != pc2.model AND pc1.hd = pc2.hd;
- e) SELECT pc1.model FROM PC AS pc1, PC AS pc2 WHERE
pc2.model != pc1.model AND
pc2.model >= pc1.model AND
pc2.ram = pc1.ram AND
pc2.speed = pc1.speed;

9. a)

```

1  SELECT maker FROM Product WHERE model IN (
2      SELECT model FROM PC WHERE product.model = PC.model AND
3      PC.speed >= 3.0
4  );
5

```

b)

```

1  SELECT p1.model FROM Printer AS p1 WHERE \
2  p1.price >= ALL (
3      SELECT p2.price FROM Printer AS p2
4  )
5

```

Notes:

- EXISTS

- EXISTS R is a condition that is true if and only if relation R is not empty

```

1  SELECT SupplierName
2  FROM Suppliers
3  WHERE EXISTS (SELECT ProductName FROM Products WHERE
4      Products.SupplierID = Suppliers.supplierID AND Price = 22);

```

- s IN R

- is true if and only if s is equal to one of the values in R .
- s NOT IN R true if and only if s has no value in R .

```

1  SELECT name
2  FROM MovieExec
3  WHERE cert# IN
4      (SELECT producerC#
5       FROM Movies
6       WHERE (title, year) IN
7           (SELECT movieTitle movieYear
8            FROM StarsIn
9            WHERE starName = 'Harrison Ford'
10           )
11      );
12

```

- $s > \text{ANY } R$

- is true if and only if s is greater than at least one value in unary relation R .

- $s > \text{ALL } R$

- is true if and only if s is greater than at least one value in unary relation R .