

Solution: Relational algebra and SQL

SQL queries and query results

1. `SELECT DISTINCT a, b FROM r1 WHERE b > 1 OR c = 3 UNION SELECT a, b FROM r4;`

a	b
1	2
2	0
1	1

2. `SELECT DISTINCT r3.c, d, e, a FROM r3 CROSS JOIN r4 WHERE d=1;`

c	d	e	a
1	1	0	1
4	1	1	1
2	1	0	1

3. `SELECT DISTINCT r1.b, r1.c, r4.a, r4.b FROM r1 CROSS JOIN r4 WHERE r1.c=3;`

b	c	a	b
0	3	1	2
0	3	1	1

4. Option 1: `SELECT a, c FROM r1 UNION SELECT a,c FROM r4;`
 Option 2: `SELECT DISTINCT a, c FROM (SELECT * FROM r1 UNION SELECT * FROM r4);`

a	c
1	0
1	2
2	3
1	4
1	1

5. `SELECT DISTINCT a, d FROM r3 NATURAL JOIN r2 WHERE (a=0 OR a=2) AND d=1;`

a	d
0	1
2	1

6a. `SELECT DISTINCT * FROM r1 LEFT OUTER JOIN r2 USING(a);`

a	b	c	d
1	2	4	1
1	1	1	1
2	0	3	1

6b. `SELECT DISTINCT * FROM r1 RIGHT OUTER JOIN r2 USING(a);`

a	b	c	d
1	2	4	1
0			1
1	1	1	1
2	0	3	1

6c. `SELECT DISTINCT * FROM r1 FULL OUTER JOIN r2 USING(a);`

a	b	c	d
1	2	4	1
0			1
1	1	1	1
2	0	3	1

6d. `(SELECT a as r1.a", b, c, NULL as r2.a", NULL as d FROM r1)
UNION ALL (SELECT NULL, NULL, NULL, * FROM r2);`

r1.a	b	c	r2.a	d
1	1	1		
1	2	4		
2	0	3		
			1	1
			0	1
			2	1

6e. `SELECT * FROM r1 WHERE EXISTS(SELECT a FROM r1 NATURAL JOIN r2);`

a	b	c
1	1	1
1	2	4
2	0	3

6f. SELECT * FROM r1 JOIN r2 ON (r1.a=r2.a AND b>1);

a	b	c	a	d
1	2	4	1	1

7a. SELECT DISTINCT * FROM r2 LEFT OUTER JOIN r3 USING(d);

d	a	c	e
1	2	4	1
1	1	1	0
1	1	2	0
1	2	1	0
1	1	4	1
1	2	2	0
1	0	2	0
1	0	1	0
1	0	4	1

7b. SELECT DISTINCT * FROM r2 RIGHT OUTER JOIN r3 USING(d);

d	a	c	e
1	2	4	1
1	1	1	0
3		2	3
1	1	2	0
1	2	1	0
1	1	4	1
1	2	2	0
1	0	2	0
1	0	1	0
1	0	4	1

7c. SELECT DISTINCT * FROM r2 FULL OUTER JOIN r3 USING(d);

d	a	c	e
1	2	4	1
1	1	1	0
3		2	3
1	1	2	0
1	2	1	0
1	1	4	1
1	2	2	0
1	0	2	0
1	0	1	0
1	0	4	1

- 7d. (SELECT a ,d as r2.d", NULL as c", NULL as r3.d", NULL as e FROM r2) UNION ALL (SELECT NULL, NULL, * FROM r3);

a	r2.d	c	r3.d	e
1	1			
0	1			
2	1			
		1	1	0
		4	1	1
		2	1	0
		2	3	3

- 7e. SELECT * FROM r2 WHERE EXISTS(SELECT d FROM r2 NATURAL JOIN r3);

a	d
1	1
0	1
2	1

8. (SELECT d FROM r3) EXCEPT (SELECT d FROM r2);

d
3

9. SELECT * FROM (SELECT a from r2) AS t1 NATURAL JOIN (SELECT * FROM r3 WHERE c=2) AS t2;

a	c	d	e
1	2	1	0
1	2	3	3
0	2	1	0
0	2	3	3
2	2	1	0
2	2	3	3

Appendix: Translation of the tables to SQL

```
--
-- CREATE TABLES
--

CREATE TABLE r1 (a int, b int, c int);
CREATE TABLE r2 (a int, d int);
CREATE TABLE r3 (c int, d int, e int);
CREATE TABLE r4 (a int, b int, c int);

--
-- INSERT VALUES
--

INSERT INTO r1 (a, b, c) VALUES (1, 1, 1);
INSERT INTO r1 (a, b, c) VALUES (1, 2, 4);
INSERT INTO r1 (a, b, c) VALUES (2, 0, 3);

INSERT INTO r2 (a, d) VALUES (1, 1);
INSERT INTO r2 (a, d) VALUES (0, 1);
INSERT INTO r2 (a, d) VALUES (2, 1);

INSERT INTO r3 (c, d, e) VALUES (1, 1, 0);
INSERT INTO r3 (c, d, e) VALUES (4, 1, 1);
INSERT INTO r3 (c, d, e) VALUES (2, 1, 0);
INSERT INTO r3 (c, d, e) VALUES (2, 3, 3);

INSERT INTO r4 (a, b, c) VALUES (1, 1, 0);
INSERT INTO r4 (a, b, c) VALUES (1, 2, 2);
```