Write a relational algebra expression equivalent to the following relational calculus expression:

\[
\{ t: \text{fanID}, \text{fanName} \mid \\
\exists f \in \text{fan} \left( f(\text{fanID}) = t(\text{fanID}) \land \\
\quad f(\text{fanName}) = t(\text{fanName}) \land \\
\quad \forall e \in \text{likes} \left( e(\text{fanID}) = f(\text{fanID}) \rightarrow \\
\quad \left( \forall b \in \text{band} \left( b(\text{bandID}) = e(\text{bandID}) \rightarrow \\
\quad \text{genre} \neq '\text{rock}' \right) \right) \right) \}
\]
Solution:

[Finds fans that do not like any rock band (project problem 2c)]

\[ \Pi_{f1d, \text{fanName}} \neg (\text{fan} \in \text{like} \text{ fan} \in \text{genre} = 'rock' (\text{band})) \]
2) Consider the following virtual view: (schema: R(x, y))

CREATE VIEW v AS
SELECT x
FROM R
WHERE y = '2'

and the query:

SELECT x, y
FROM R, v
WHERE v.x = R.x

Unfold this view in this query & come up with a result without nesting.
Solution:

```
SELECT x, y
FROM R, (SELECT x 
    FROM R 
    WHERE y = '2') AS V 
WHERE V.x = R.x
```

Q1: SELECT x, y
FROM R
WHERE y = '2'

\[\not\equiv\] Equivalent? NO

Q2: SELECT r1.x, r1.y
FROM R r1, R r2
WHERE r1.x = r2.x AND r2.y = '2'
Why are $Q_1$ & $Q_2$ not equivalent?

$R$ | $X$ | $Y$
---|---|---
$a$ | 1 | $X_2$
$r_1 \rightarrow a$ | 2 | $X_2$

$q_1$:


$q_2$:

The right solution is $Q_2$!

But what if $R \cdot x$ is a PK2?
Recursion:

Consider the relation:

\[ \text{parent}(x, y) : x \text{ is the parent of } y \]

Express in SQL using a recursive view or query returning all descendants of Anna.
Solution:

CREATE RECURSIVE VIEW desc AS

(SELECT * FROM parent)

UNION

(SELECT p.x, d.y
FROM parent p, desc d
WHERE p.y = d.x)

SELECT *
FROM desc
WHERE x = 'Anna'
4) Is this a rewrite rule?

\[ \text{cond} \quad \text{Ty} \quad \text{cond} \quad \text{R} \]

Solution: No

5) Under which condition would this be a rewrite rule?

Answer: When cond involves only y
b) How can you fix it in the general case?
5) Create logical query plan for the following query: \[ R(a_1, a_2), S(b_1, b_2) \] :

\[
\text{SELECT } a_1, b_1, b_2 \\
\text{FROM } R, S \\
\text{WHERE } R.a_1 = '2' \text{ AND } S.b_1 = '3'
\]

6) Create a different logical plan where you push down the selections & projections as much as possible.

7) Create a physical plan for it.
Solution:

(a) \[ \prod_{a_1, b_1, b_2} \]
\[ \times \]
\[ R \times S \]

\[ \prod_{a_1, b_1, b_2} \]
\[ \times \]
\[ \times \]
\[ R \times S \]

(b) \[ \prod_{a_1, b_1, b_2} \]
\[ \times \]
\[ \times \]
\[ R \times S \]
if we know that we have index on $R.a_1$