Section 7: Relational Algebra, Query Execution

1. Write the logical plan for the queries below.

(a) Product(pid, name, price)
    Purchase(pid, cid, store)
    Customer(cid, name, city)

(b) SELECT z.city, sum(x.price)
    FROM Product x, Purchase y, Customer z
    WHERE x.pid = y.pid and y.pid = z.pid
    and y.store = 'Walmart'
    GROUP BY z.city
    HAVING count(*) > 100

(c(i)) SELECT DISTINCT z.name
    FROM Customer z
    WHERE z.city='Seattle' AND
    not exists (select *
        from Product x, Purchase y
        where x.pid = y.pid
        and y.cid = z.cid
        and x.price < 100)
(b) [10 points] Write a SQL query that is equivalent to the logical plan below:

```
SELECT DISTINCT r.a, t.g
FROM R(a, b, f) S(b, c, h) T(c, d, g)
WHERE r.a > 5 AND c.g < 9 AND
      r.b = s.b AND s.c = t.c
```

(a) [10 points] Consider two tables \( R(A, B, C) \) and \( S(D, E, F) \), and the query plan \( P \) below. Indicate which of the four query plans \( P_1, P_2, P_3, P_4 \) are equivalent to \( P \). The symbol \( \bowtie \) in \( P_4 \) represents the right semi-join, i.e. \( R \bowtie S = S \bowtie R \).

\[
P = \sigma_{A > 0}(\gamma_{A, \text{sum}(F)}(R \bowtie_{C = D} S))
\]

\[
P_1 = \gamma_{A, \text{sum}(F)}(\sigma_{A > 0}(R) \bowtie_{C = D} \gamma_{D, \text{sum}(F)} S)
\]

\[
P_2 = \gamma_{A, \text{sum}(F)}(\sigma_{A > 0}(R) \bowtie_{C = D} \gamma_{D, E, \text{sum}(F)} S)
\]

\[
P_3 = \gamma_{A, \text{sum}(F)}(\sigma_{A > 0}(R) \bowtie_{C = D} \gamma_{D, E, \text{sum}(F)}(\sigma_{A > 0}(R) \bowtie_{C = D} S))
\]

\[
P_4 = \gamma_{A, \text{sum}(F)}(\sigma_{A > 0}(R) \bowtie_{C = D} \gamma_{D, E, \text{sum}(F)}(\sigma_{A > 0}(R) \bowtie_{C = D} S))
\]

\* Try performing query plans on relations below:

\[
\begin{array}{c|ccc}
S & D & E & F \\
\hline
C_1 & e_1 & f_1 \\
C_2 & e_2 & f_2 \\
C_3 & e_1 & f_3 \\
\end{array}
\]

\[
\begin{array}{c|ccc}
R & A & B & C \\
\hline
a_1 & b_1 & c_1 \\
a_2 & b_1 & c_1 \\
\end{array}
\]
3. Start with the B+ tree below and perform the actions indicated in sequence. (D = 1)

a) Insert Record 60

b) Insert Record 15
c) Delete Record 50

---

d) Delete Record 20

* For this example, there is a possibility to collapse the node with 20 into the node with 60. This solution shows the choice of doing a rotation.

---

e) Delete Record 10