Database Management Systems

1. The following relations are given (primary keys are underlined):

```
SINGER(SCode, SName, City, DateofBirth)
RECORD_LABEL(RLCode, RLName, Address, City)
DISC(DCode, Title, SCode, RLCode, Type, Price)
SALE(DCode, Date, SoldCopyNumber)
```

   Assume the following cardinalities:

   - card(SINGER)= 10^4 tuples,
     MIN(DateofBirth) = 1-1-1969, MAX(DateofBirth) = 31-12-1998,
   - card(RECORD_LABEL)= 10^4 tuples,
     number of City $\simeq 100$,
   - card(DISC)= 10^6 tuples,
     MIN(Price) = 10, MAX(Price) = 29,
   - card(SALE)= 10^8 tuples for year 2010.

   Furthermore, assume the following reduction factor for the group by condition:

   - having sum(SoldCopyNumber)$\geq$10.000 $\simeq \frac{1}{100}$.

   Consider the following SQL query:

```
select RLName, City
from RECORD_LABEL RL
where City='Rome' and RLCode IN (select D.RLCode
                      from SALE S, DISC D, SINGER SI
                      where S.DCode=D.DCode and SI.SCode=D.SCode
                      D.Price ≤ 11 and SI.DateofBirth ≤ 31/12/1983
                      and S.Date ≤ 31/01/2010
                      group by S.DCode, D.RLCode
                      having sum(SoldCopyNumber)≥10.000)
```

   For the SQL query:

   (a) Report the corresponding algebraic expression and specify the cardinality of each node (representing an intermediate result or a leaf). If necessary, assume a data distribution. Also analyze the group by anticipation.

   (b) Select one or more secondary physical structures to increase query performance. Justify your choice and report the corresponding execution plan (join orders, access methods, etc.).