1. (6 Points) The following relations are given (primary keys are underlined):

\[
\begin{align*}
\text{USER}(\text{Uid}, \text{Name}, \text{Surname}, \text{City}, \text{State}, \text{BirthDate}) \\
\text{PHOTO}(\text{Pid}, \text{Uid}, \text{Resolution}, \text{KBs}) \\
\text{UPLOAD}(\text{Pid}, \text{Date}, \text{Time}, \text{Description}) \\
\text{TAG}(\text{Pid}, \text{Tag})
\end{align*}
\]

Assume the following cardinalities:

- \(\text{card(USER)} = 10^6\) tuples, \\
  \(\text{MIN}(\text{BirthDate}) = 1-1-1941, \ \text{MAX}(\text{BirthDate}) = 31-12-1990,\)
  number of \text{City} \simeq 10^2, \\
  number of \text{State} \simeq 10,

- \(\text{card(PHOTO)} = 10^8\) tuples, \\
  number of \text{Resolution} \simeq 10, \\
  \(\text{MIN}(\text{KBs}) = 10^2, \ \text{MAX}(\text{KBs}) = 2 \cdot 10^3,\)

- \(\text{card(UPLOAD)} = 5 \cdot 10^8\) tuples, \\
  \(\text{MIN}(\text{Date}) = 01-01-2010, \ \text{MAX}(\text{Date}) = 31-12-2010,\)

- \(\text{card(TAG)} = 10^9\) tuples

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

- \(\text{having count(distinct tag)} \leq 10 \simeq \frac{99}{100}.\)

Consider the following SQL query:

\[
\begin{align*}
\text{select } \text{Uid}, \text{Name}, \text{Surname} \\
\text{from USER U, PHOTO P} \\
\text{where U.UId=}P.\text{Pid} \text{ and BirthDate } > 1-1-1981 \\
\text{and City } = \text{'Rome'} \\
\text{and PId NOT IN (select UP.PId} \\
\text{from UPLOAD UP, TAG T, PHOTO P1} \\
\text{where UP.PId=}T.\text{Pid} \text{ and UP.PId=}P1.\text{Pid} \\
\text{and Date } \geq 01-06-2010 \text{ and Date } \leq 30/06/2010 \\
\text{and Resolution } <> \text{'1280x720'} \text{ group by UP.PId} \\
\text{having count(distinct Tag) } \leq 10);\end{align*}
\]

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.).

Join and group by discussion:

(1) Hash Join
(2) Hash Join
(3) GB No Hash
(4) Hash Join or Nested loop
(5) Hash Join

Indexes:
- **Table Upload**: B+ Tree on Date
- **Table User**: Hash on City