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

\[
\begin{align*}
\text{PLAY-ACTOR}(AId, \text{Name}, \text{Nationality}, \text{BirthDate}) \\
\text{COMEDY}(\text{ComId}, \text{Title}, \text{Director}, \text{SceneNumber}, \text{Year}) \\
\text{PLAY-ACTOR-IN-COMEDY}(\text{ComId}, \text{AId}, \text{Role}) \\
\text{PLANNING}(\text{ComId}, \text{Theater}, \text{Date}, \text{StartTime}, \text{LengthOfTime})
\end{align*}
\]

Assume the following cardinalities:

- \( \text{card(PLAY-ACTOR)} = 10^4 \) tuples,
  \( \text{MIN(BirthDate)} = 1-1-1960, \text{MAX(BirthDate)} = 31-12-1999, \)
- \( \text{card(COMEDY)} = 10^3 \) tuples,
  \( \text{distinct values of SceneNumber} \approx 15, \)
- \( \text{card(PLAY-ACTOR-IN-COMEDY)} = 10^6 \) tuples,
  \( \text{distinct values of Role} \approx 30, \)
- \( \text{card(PLANNING)} = 10^8 \) tuples,
  \( \text{MIN(Date)} = 1-1-2010, \text{MAX(Date)} = 31-12-2010, \)
  \( \text{MIN(LengthOfTime)} = 81, \text{MAX(LengthOfTime)} = 180, \)

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

- \( \text{having count(DISTINCT Theater)} \geq 50 \approx \frac{1}{10}. \)

Consider the following SQL query:

\[
\begin{align*}
\text{select Title, Director} \\
\text{from COMEDY C, PLANNING P, PLAY-ACTOR-IN-COMEDY AC} \\
\text{where P.ComId=C.ComId and AC.ComId=C.ComId} \\
\text{and LengthOfTime=180 and SceneNumber}>12 \\
\text{and AC.Aid in (select Aid from PLAY-ACTOR} \\
\text{where BirthDate} \geq 1996) \\
\text{group by ComId, Title, Director} \\
\text{having count(DISTINCT Theater)} \geq 50
\end{align*}
\]

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) Nested loop
(2) Hash Join
(3) Hash Join or Nested loop
(4) GB Hash

Indexes:
- **Table PLANNING**: $B^+$-Tree on LengthOfTime
- **Table PLAY-ACTOR**: $B^+$-Tree on BirthDate

Group by anticipation:

**Pay attention**: The distinct statement should be inserted into the select clause since a many-to-many relationship holds among comedy and actor tables.

Join and group by discussion:

1. Nested loop
2. Nested loop
3. Hash Join or Nested loop
4. GB Hash