Database

You are given the following relational schema:

\[
\begin{align*}
\text{Movie}(\text{Title, Dname}) \\
\text{Cast}(\text{Title, Aname, Role}) \\
\text{Actor}(\text{Aname, Addr}) \\
\text{Award}(\text{Aname, Year, Title})
\end{align*}
\]

Keys are underlined.

**Problem 1**

Consider the following query:

\[
Q_1 \equiv \text{Movie} \bowtie \text{Cast} \bowtie \text{Actor} \bowtie \text{Award}.
\]

You are given the following cardinality estimates:

\[
\begin{align*}
T(\text{Movie}) &= V(\text{Movie, Title}) = 10000 \\
V(\text{Movie, Dname}) &= 1000 \\
T(\text{Cast}) &= 50000 \\
V(\text{Cast, Title}) &= 10000 \\
T(\text{Actor}) &= V(\text{Cast, Aname}) = V(\text{Actor, Aname}) = 4000 \\
T(\text{Award}) &= 500 \\
V(\text{Award, Aname}) &= 200 \\
V(\text{Award, Title}) &= 100
\end{align*}
\]

Calculate the plan of least estimated cost using the dynamic programming algorithm.

**Solution**

\[
\begin{array}{|c|c|c|}
\hline
\text{Relations} & \text{Cost} & \text{Best plan} \\
\hline
\text{Movie, Cast} & 0 & \text{Movie} \bowtie \text{Cast} \\
\text{Cast, Actor} & 0 & \text{Actor} \bowtie \text{Cast} \\
\text{Actor, Award} & 0 & \text{Award} \bowtie \text{Actor} \\
\text{Movie, Award} & 0 & \text{Award} \bowtie \text{Movie} \\
\text{Cast, Award} & 0 & \text{Award} \bowtie \text{Cast} \\
\text{Movie, Cast, Actor} & 50000 & (\text{Movie} \bowtie \text{Cast}) \bowtie \text{Actor} \\
\text{Movie, Cast, Award} & 500 & (\text{Award} \bowtie \text{Movie}) \bowtie \text{Cast} \\
\text{Cast, Actor, Award} & 500 & (\text{Award} \bowtie \text{Actor}) \bowtie \text{Cast} \\
\text{Movie, Actor, Award} & 500 & (\text{Award} \bowtie \text{Actor}) \bowtie \text{Movie} \\
\text{Movie, Cast, Actor, Award} & 501 & ((\text{Award} \bowtie \text{Movie}) \bowtie \text{Cast}) \bowtie \text{Actor} \\
\hline
\end{array}
\]

For multiple plans with the same cost only one is selected above.
**Problem 2**

Consider the following query:

\[ Q_2 \equiv \sigma_{\text{Cast.Role} = \text{"star"}}(\text{Movie} \bowtie \text{Cast}). \]

Explain how indexes can be used to make the evaluation of the query \( Q_2 \) more efficient. List all such indexes and specify whether they are primary or secondary, sparse or dense. Describe the evaluation plans that use the indexes.

**Solution**

1. Dense secondary index on \( \text{Cast.Role} \).
2. Dense secondary index on \( \text{Cast.Title} \).

Evaluation plans:

1. \( \text{HashJoin(TableScan(Movie), IndexScan(Cast, Cast.Role = \text{"star"}))} \)
2. \( \text{Filter(IndexJoin(TableScan(Movie), IndexScan(Cast, Cast.Title = Movie.Title), Cast.Role = \text{"star"}))} \)

There are other possible evaluation plans but they require joining the relation in a different order which is suboptimal (\( \text{Movie} \) is smaller than \( \text{Cast} \)).