SS24 name_____
(neatly)

Final Exam
CS 5300

Instructions

1.) Read **all** instructions carefully.
2.) **Important:** Assume that there are **NO** typos in the test.
3.) Place answers in blanks if they are provided. If you are asked to write a short answer, make it literate.
4.) If you have a question about a problem/question (besides how to solve it), be sure to ask.
5.) Multiple choice questions will have ONLY one correct answer unless stated otherwise.
6.) Good Luck!

Score: _____ / 100 total possible points

1. Which of the following Relational Algebra expressions are equivalent to $R - \sigma_c (S)$? Select ALL the correct answers. [5 points]

   a. $\sigma_c (R) - S$
   b. $\sigma_c (R - S)$
   c. $\sigma_c (S - R)$
   d. $\sigma_c (R) - \sigma_c (S)$
   e. None of the above

2. Given the following relational database schema related to the "Squid Game" TV show and the specific SQL query, construct the corresponding query tree in canonical form (i.e., pre-heuristic optimization). [10 points]

   PLAYERS (PlayerID, Name, Age, Gender)
   GAMES (GameID, GameName, DifficultyLevel)
   PLAYER_PERFORMANCE (PlayerID, GameID, Score, Status)

   Query: The following query finds the average score of female players for each age group who participated in and passed the "Tug of War" game. Only those age groups where the average score is above 8.0 should be displayed.

   ```
   SELECT    P.Age, AVG (PP.Score) AS AverageScore
   FROM      PLAYERS AS P, GAMES AS G, PLAYER_PERFORMANCE AS PP
   WHERE     P.PlayerID = PP.PlayerID AND
             G.GameID = PP.GameID AND
             G.GameName = 'Tug of War' AND
             PP.Status = 'Passed' AND
             P.Gender = 'Female'
   GROUP BY  P.Age
   HAVING    AVG (PP.Score) > 8.0;
   ```

   Draw the initial query tree for this SQL query. Your tree should clearly show the Relational Algebra operations like Cartesian Products, Selections, Projections, etc.

3. Based on the initial query tree you produced in the previous question, construct an equivalent, optimized query tree by applying the heuristic optimization rules to it. Explain the optimization steps you've applied and why they improve the query performance. [20 points]

4. Suppose that the PLAYERS table from the previous questions has 2000 records stored in 100 disk blocks and the following access paths:

   - A primary index on PlayerID, with levels $x_{PlayerID} = 2$.
   - A clustering index on Gender, with levels $x_{Gender} = 4$. There are a total of 3 distinct values for Gender.

- A bitmap index on Age that is stored in 2 disk blocks. There are a total of 10 distinct values for Age.

For each of the following selection operations, compute the cost of performing the operation using the brute force approach and the available indices. Justify which method is more cost-effective for each operation.

- OP1: $\sigma_{PlayerID = 007}$ *(PLAYERS)* [5 points]
- OP2: $\sigma_{Gender != 'Female'}$ *(PLAYERS* [5 points]
- OP3: $\sigma_{Gender = 'Female' AND Age IN (20, 30, 40)}$ *(PLAYERS)* [5 points]

5. Suppose that the following JOIN operation is performed on relations PLAYERS and PLAYER_PERFORMANCE from the previous questions. PLAYERS has 2000 records stored in 100 disk blocks and PLAYER_PERFORMANCE has 5300 records stored in 300 disk blocks.

   Query:

```
SELECT    *
FROM      PLAYERS AS P, PLAYER_PERFORMANCE AS PP
WHERE     P.PlayerID = PP.PlayerID;
```

   Assume there are 20 main memory buffers available and that the blocking factor for writing the join result back to disk is 37 records per block. The following access paths are also available:

   - A primary index on P.PlayerID, with levels $x_{PlayerID} = 2$.
   - A primary index on the composite key (PP.PlayerID, PP.GameID) with levels $x_{(PlayerID, GameID)} = 3$.
   - In addition, the relation PP is sorted on PlayerID in ascending order. The PLAYERS table is also sorted on PlayerID in ascending order.

   a. Estimate the cost using the Simple Nested-Loop Join. Assume that one buffer is reserved for reading from the INNER relation, another one for writing to the output file, and the remaining for reading from the OUTER relation. Justify your decision on choosing which relation should be the outer or inner. [5 points]

   b. Estimate the cost using the Simple Nested-Loop Join. Assume that one buffer is reserved for reading from the OUTER relation, another one for writing to the output file, and the remaining for reading from the INNER relation. Justify your decision on choosing which relation should be the outer or inner. [5 points]

   c. Estimate the cost using the Index-based Nested-Loop Join with the provided available indices. Justify your decision on choosing which relation should be the outer or inner, and how the buffers should be best allocated. [10 points]

   d. Estimate the cost using the Sort-Merge Join. [10 points]

   e. Estimate the cost using the Partition-Hash Join. [10 points]

   f. Determine the optimal join method from your computations in options a through e. [5 points]

6. Select ALL the following statements that are FALSE on multi-relation queries. [5 points; 1 point each]

   a. The number of equivalent query trees grows rapidly as the number of aggregation operators in a multi-relation query increases.
   b. A query that joins *n* relations will have at most (*n-1*) join operations that could be rearranged in *n!* different join orders.
   c. A left-deep join tree is a binary tree in which the left child of each non-leaf node is always a base relation.
   d. It is easier to apply pipelining on a left/right-deep join tree than a bushy join tree.
   e. The order in which joins are executed in a query is NOT something that the database's query optimizer concerns with.