1. Explain the vectorized execution model. Compare it briefly with the Tuple-at-a-Time and the Operator-at-a-Time processing model. Describe the impact of the vector size.

2. Consider the following DBMSs:
   - MonetDB/MIL (Operator-at-a-Time)
   - MySQL (Tuple-at-a-Time)
   - MonetDB/X100 (Vector-at-a-Time)

   Assign the different DBMSs to the following performance diagram of the TPC-H Query 1 (Data size: 1 GB):

   What unit is used on the x axis? Explain the performance difference of the different systems.

3. For which of the following scenarios would you use a tuple-at-a-time or an operator-at-a-time processing model?
   - Disk-based systems with small main memory.
   - Maximize instructions-per-cycle.
   - Nested-loops join on very large tables.
   - In-memory database with maximum CPU utilization.
   - Avoid big intermediate results.
   - Minimize instruction cache misses

4. Describe the two different storage models presented in the lecture. Especially, consider the following aspects:
   (a) Usability for OLTP/OLAP
   (b) Compression techniques
   (c) Query execution
5. Consider the following table Shops:

<table>
<thead>
<tr>
<th>ID (int)</th>
<th>Shop (Char 10)</th>
<th>Revenue (Double)</th>
<th>City (Char 20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H&amp;M</td>
<td>1357,68</td>
<td>Berlin</td>
</tr>
<tr>
<td>2</td>
<td>C&amp;A</td>
<td>2766,12</td>
<td>Dresden</td>
</tr>
<tr>
<td>3</td>
<td>McDonalds</td>
<td>30000,23</td>
<td>Magdeburg</td>
</tr>
<tr>
<td>4</td>
<td>PizzaHut</td>
<td>11999,99</td>
<td>Berlin</td>
</tr>
<tr>
<td>5</td>
<td>H&amp;M</td>
<td>24135,76</td>
<td>Dortmund</td>
</tr>
</tbody>
</table>

Explain how the data would be stored within the cache (Cache line: 64 byte, Cache size: 6 lines).

How many cache lines must be loaded for the following queries:

- SELECT * FROM Shops where revenue >10.000;
- INSERT INTO Shops VALUES(6,'New Yorker', 16724.35,'Stuttgart');
- SELECT SUM(revenue) FROM Shops WHERE City='Berlin';
- SELECT Shop FROM Shops GROUP BY Shop HAVING COUNT(*) >1;

6. Why is lightweight data compression so important, especially for main-memory DBMSs? Explain the basic principle of dictionary encoding? Use a self-chosen example. Explain how dictionary encoded data can be processed without decompressing it!

7. Explain and compare different materialization strategies for column-stores. What are their advantages?

Good Luck!