EXAM ANSWER KEY

Computer Science - Advanced Topics
Date: December 18, 2025

Question 1 [10 marks]

Question: Explain the concept of ACID properties in database transactions. Discuss how each property ensures data consistency and provide real-world scenarios where violation of these properties could lead to problems.

Evaluation Rubric:

Trait	Weight	Description
Concept Coverage	40%	Comprehensive explanation of all 4 ACID properties (Atomicity, Consistency, Isolation
Real-World Application	30%	Clear examples of transactions and scenarios where violations cause problems
Logical Flow	20%	Well-organized answer with clear connections between properties
Clarity & Language	10%	Clear writing, appropriate terminology usage

Question 2 [8 marks]

Question: Write a SQL query to find the top 5 departments by average salary, excluding departments with fewer than 10 employees. Include department name and average salary in results.

Expected Answer:

```
SELECT d.dept_name, AVG(e.salary) AS avg_salary
FROM departments d
INNER JOIN employees e ON d.dept_id = e.dept_id
GROUP BY d.dept_id, d.dept_name
HAVING COUNT(e.emp_id) >= 10
ORDER BY avg_salary DESC
LIMIT 5;
```

Key Points:

- Correct JOIN syntax connecting departments and employees tables
- GROUP BY clause with both dept_id and dept_name
- HAVING clause for filtering groups by employee count (COUNT >= 10)
- ORDER BY with DESC for descending average salary
- LIMIT 5 to restrict result set
- Correct aggregation function AVG()

Question 3 [12 marks]

Question: Discuss the differences between supervised and unsupervised learning. Provide at least two examples for each category and explain why certain problem types are better suited to each approach.

Evaluation Rubric:

Trait	Weight	Description
Fundamental Differences	35%	Clear explanation of labeled vs unlabeled data, training approach differences
Examples Provided	35%	Minimum 2 valid examples each (supervised: classification, regression; unsupervised:

Problem Suitability Analysis	20%	Reasoning for why certain problems fit each approach
Organization & Clarity	10%	Well-structured response with clear delineation between sections

Question 4 [7 marks]

Question: Given an array of integers, implement an algorithm to find if there exists a subarray with sum equal to a target value. Time complexity should not exceed O(n).

Expected Answer Approach:

```
def find_subarray_sum(arr, target):
    seen_sums = {0} # Set to track cumulative sums
    current_sum = 0

    for num in arr:
        current_sum += num
        if (current_sum - target) in seen_sums:
            return True
        seen_sums.add(current_sum)
    return False
```

Key Points:

- Use of hash set/dictionary for O(1) lookup of cumulative sums
- Cumulative sum approach to track running total
- Logic: if (current_sum - target) exists, then subarray found
- Single pass through array = O(n) time complexity
- O(n) space complexity for set storage
- Handles negative numbers and edge cases
- Alternative: Sliding window for positive integers only

Question 5 [9 marks]

Question: Analyze the impact of network latency and bandwidth constraints on distributed system design. How would you architect a system to handle high-latency and low-bandwidth environments?

Evaluation Rubric:

Trait	Weight	Description
Impact Analysis	30%	Explanation of latency/bandwidth effects on system performance and reliability
Design Strategies	40%	Multiple strategies (caching, compression, async communication, local processing, ba
Trade-offs Discussion	20%	Understanding of consistency vs performance trade-offs
Coherence & Examples	10%	Well-articulated with practical examples

Question 6 [6 marks]

Question: Write a MongoDB aggregation pipeline to calculate the monthly revenue for each product category, sorting by revenue in descending order.

Expected Answer:

```
db.orders.aggregate([
    {
        $group: {
            _id: {
                category: "$product.category",
                month: { $dateToString: { format: "%Y-%m", date: "$order_date" } }
            },
            monthly_revenue: { $sum: "$total_amount" }
        }
    },
    {
        $sort: { monthly_revenue: -1 }
    }
])
```

Key Points:

- Correct $group stage with nested _id for category and month
- $dateToString for formatting date to YYYY-MM format
- $sum aggregation to calculate total revenue
- $sort stage with -1 for descending order
- Proper MongoDB syntax and nested field access ($product.category)
- Handles multiple documents per category-month combination
- Alternative: Using $month and $year functions instead of $dateToString

GRADING GUIDELINES

Question	Type	Max Marks	Grading Method
1	DESCRIPTIVE	10	Rubric-based with weighted traits
2	TECHNICAL	8	Correctness + Code quality + Keywords
3	DESCRIPTIVE	12	Rubric-based with weighted traits
4	TECHNICAL	7	Algorithm correctness + Complexity analysis
5	DESCRIPTIVE	9	Rubric-based with weighted traits
6	TECHNICAL	6	Query correctness + Syntax + Stage usage
		<b>Total: 52</b>	

Note: This answer key is for instructor reference only. Students should demonstrate understanding of concepts, proper methodology, and clear communication. Partial credit may be awarded for partially correct answers.