25F - Data Engineering (IMEN891K/IDSC501)

Final Exam (Total: 25 points)

Name: _____ **Student ID:** _____

Part I: Multiple-Choice (8 points)

Select the correct answer from the choices. Each question is worth 1 point.

1. Which NoSQL data store is most suitable for storing user activity logs?
   a. Key-Value Store b. Wide-Column Store c. Graph Store d. Document Store
2. What is the data model of a document store?
   a. Array b. Set c. Priority Queue d. Tree
3. How are NoSQL databases typically designed to scale?
   a. Vertically (scale-up) b. Horizontally (scale-out)
4. Which MapReduce stage groups all intermediate values by key?
   a. Map b. Merge c. Shuffle d. Reduce
5. Which of the following Spark functions is NOT a lazy operator?
   a. map() b. filter() c. join() d. count()
6. What is the primary query language for BigQuery?
   a. SQL b. RDD c. DataFrame d. Dremel
7. Which system should store data in a row-oriented manner?
   a. Database (OLTP) b. Data Warehouse (OLAP)
8. Which tool should be chosen to implement a real-time application?
   a. BigQuery b. Airflow c. Kafka d. DataLake

Part II: Short Answer (8 points)

Fill in the blanks. Each blank is worth 1 point.

1. What allows Spark to be fast yet fault-tolerant?
   Answer: RDD = _____ (speed) + _____ (fault tolerance)
2. In Apache Kafka, a (1) is a logical category for messages, whereas a (2) is the physical unit of storage distributed across the cluster for parallel processing.
   Answer: (1) _____ (2) _____
3. In version control (e.g., Git), ML models are typically developed in a dev _____ and deployed in a main _____ after testing.
   Answer: _____

4. Which technique allows LLMs to utilize external knowledge without retraining?

   Answer: _____

5. Which two components are integrated in a ReAct agent, as the name implies?

   Answer: ReAct = _____ + _____

Part III: Coding (9 points)

Fill in the blanks. Each blank is worth 1 point.

1. Align each line of MongoDB code with the corresponding SQL syntax.
   Hint: SELECT, FROM, WHERE, JOIN, GROUP BY, HAVING, DISTINCT

```
db.inventory.find(                                          (1) _________________
    {tags: "red"},                                          (2) _________________
    {_id: 0, instock: 0})                                   (3) _________________

db.zips.aggregate([
    {$group: {_id: "$state", totalPop: {$sum: "$pop"}}}, (4) _________________
    {$match: {totalPop: {$gte: 15000000}}}                  (5) _________________
])
```

2. Complete the Spark RDD code based on the SQL query for `R(A,B)` and `S(A,C)`.
   Note: `RB` should be backed up for later use.

```
(SQL code)
SELECT count(*) FROM R, S
WHERE R.B > 200 AND S.C < 100 AND R.A = S.A

(Spark RDD code)
RB = R._________________(t -> t.b > 200)._________________
SC = S.... (omit)
J = RB._________________(SC)
J._________________
```