

DDR Data Science Interview Practice Test (Sample)

Study Guide



Everything you need from our exam experts!

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

- Copyright** 1
- Table of Contents** 2
- Introduction** 3
- How to Use This Guide** 4
- Questions** 5
- Answers** 8
- Explanations** 10
- Next Steps** 16

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 - 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

- 1. Which is an example of referencing in MongoDB for a one-to-many relationship?**
 - A. Embed orders in the customer document**
 - B. Reference orders in a separate collection**
 - C. Embed orders in the parent document**
 - D. Reference orders via a foreign key in a separate system**

- 2. Why are hash tables efficient for lookups?**
 - A. It uses binary search to locate entries.**
 - B. It allows direct access to elements in $O(1)$ average time.**
 - C. It stores data in sorted order for fast range queries.**
 - D. It compresses data to reduce memory.**

- 3. What does collation define?**
 - A. The layout of tables on disk.**
 - B. The rules for how string data is sorted and compared.**
 - C. The default numeric precision for calculations.**
 - D. The maximum length of a string.**

- 4. What are scalar functions?**
 - A. Functions that operate on a single value and return a single value.**
 - B. Functions that operate on a set of rows and return a summary.**
 - C. Functions that modify the database schema.**
 - D. Functions that require a GROUP BY clause to work.**

- 5. In MongoDB, embedding stores related data as a nested document. Which of the following options best describes embedding?**
 - A. Referencing stores related data as a nested document.**
 - B. Embedding stores related data in a separate collection.**
 - C. Embedding stores related data as a nested document.**
 - D. Referencing uses document IDs**

- 6. What does the USING clause do in a JOIN?**
- A. It joins tables on a column with the same name, including only one copy of the join column.**
 - B. It changes the join type to a cross join.**
 - C. It forces a natural join regardless of syntax.**
 - D. It duplicates all join columns in the result.**
- 7. What SQL operation retrieves combined data from related tables?**
- A. SELECT to retrieve combined data**
 - B. JOIN to retrieve combined data**
 - C. MERGE to retrieve combined data**
 - D. UNION to retrieve combined data**
- 8. What is a Cross-Join?**
- A. A Cartesian product**
 - B. A natural join**
 - C. An inner join**
 - D. A left join**
- 9. What is a Self-Join?**
- A. A join of a table with two different tables**
 - B. A join using aggregate**
 - C. A join with a subquery**
 - D. A self-join is a join of a table with itself using aliases**
- 10. In MongoDB, embedding is storing related data as a nested document. Which option describes embedding?**
- A. Referencing stores related data as a nested document**
 - B. Embedding stores related data as a nested document**
 - C. Embedding stores data in a separate collection**
 - D. Referencing stores data in the same document**

Answers

SAMPLE

1. B
2. B
3. B
4. A
5. C
6. A
7. B
8. A
9. D
10. B

SAMPLE

Explanations

SAMPLE

1. Which is an example of referencing in MongoDB for a one-to-many relationship?

- A. Embed orders in the customer document**
- B. Reference orders in a separate collection**
- C. Embed orders in the parent document**
- D. Reference orders via a foreign key in a separate system**

In MongoDB, for a one-to-many relationship you decide between embedding the many-side inside the one-side document or keeping them in separate collections and linking them with identifiers. Referencing—storing orders in their own collection and linking to them from the customer (for example, by keeping order IDs or by having each order reference the customer)—is the better approach when the “many” side can grow large. This keeps the customer document small and avoids hitting the 16MB document size limit, makes updates to orders more scalable, and allows efficient indexing and querying of orders independently. Embedding all orders inside the customer would create very large documents and complicate updates as the number of orders grows. Referencing via a separate system, or with an external foreign key, isn’t how MongoDB typically handles relationships and can lead to consistency and integrity issues, since MongoDB does not enforce cross-collection or cross-system foreign keys. You can still retrieve related data with lookups or separate queries when needed.

2. Why are hash tables efficient for lookups?

- A. It uses binary search to locate entries.**
- B. It allows direct access to elements in $O(1)$ average time.**
- C. It stores data in sorted order for fast range queries.**
- D. It compresses data to reduce memory.**

Hash tables achieve fast lookups by providing direct access to elements in average constant time. A hash function maps a key to an index in an underlying array, so the search for a value can go straight to its bucket instead of scanning many items. In practice, with a good hash function and a reasonable load factor, most lookups take constant time on average, even as the dataset grows. When collisions occur, they’re handled with methods like separate chaining or open addressing, but the overall average remains constant-time. This is why hash tables are prized for lookups. Binary search, by contrast, requires sorted data and yields logarithmic time rather than constant time. Storing data in sorted order helps range queries and ordered iteration but adds overhead that hash tables avoid. Compression focuses on reducing memory usage, not on accelerating lookups.

3. What does collation define?

- A. The layout of tables on disk.
- B. The rules for how string data is sorted and compared.**
- C. The default numeric precision for calculations.
- D. The maximum length of a string.

Collation defines how text is compared and ordered. It sets the rules that govern string comparisons, including locale-specific conventions, case sensitivity, and whether accents are considered. This means it directly affects how strings are sorted in ORDER BY clauses and how equality checks behave in WHERE clauses or GROUP BY operations. For example, a case-insensitive collation would treat 'Apple' and 'apple' as the same in comparisons, while an accent-sensitive one would distinguish 'café' from 'cafe'. Collation decisions influence query results and sorting behavior, but they do not dictate how data is stored on disk, nor do they set numeric precision or maximum string length, which are properties of data types and storage.

4. What are scalar functions?

- A. Functions that operate on a single value and return a single value.**
- B. Functions that operate on a set of rows and return a summary.
- C. Functions that modify the database schema.
- D. Functions that require a GROUP BY clause to work.

Scalar functions are operations that take a single input value and return a single output value. They are used to transform data at the row level, applying to each value in a column and producing one result per row. For example, converting text to uppercase or measuring the length of a string are scalar operations, because each input value yields one corresponding output value. This per-row, one-to-one behavior is what defines a scalar function. This differs from aggregate functions, which combine values from multiple rows into a single summary number, such as a total, average, or count, often produced per group when you use grouping. Scalar functions don't change the database schema and don't rely on grouping to produce their results. So, a function that operates on a single value and returns a single value best captures what a scalar function is.

5. In MongoDB, embedding stores related data as a nested document. Which of the following options best describes embedding?

- A. Referencing stores related data as a nested document.
- B. Embedding stores related data in a separate collection.
- C. Embedding stores related data as a nested document.**
- D. Referencing uses document IDs

Embedding is when you store related data inside the same document as a nested structure. In practice, that means the parent document includes a subdocument or an array of subdocuments representing the related data. This lets you read the whole unit in one go and update related fields atomically, since everything lives together in one document. It's especially useful for tightly related data that you usually access together and that stays reasonably bounded in size, so the document doesn't become unwieldy. For example, a user document might include an embedded address subdocument or an array of embedded order subdocuments. This is different from referencing, where related data lives in separate collections and you store IDs to link them. Referencing is chosen when the related data is large, grows independently, or you need to access the pieces separately.

6. What does the USING clause do in a JOIN?

- A. It joins tables on a column with the same name, including only one copy of the join column.**
- B. It changes the join type to a cross join.
- C. It forces a natural join regardless of syntax.
- D. It duplicates all join columns in the result.

The key idea is that the USING clause specifies a shared column name that exists in both tables and uses equality on that column to perform the join. This is like saying the join condition is the column you named, but you write it once instead of repeating it for each table. It requires that the column exists in both sides and has compatible types. The practical effect is that you get only one copy of that join column in the result set, avoiding duplicate columns for the same data. If you list more than one column in USING, the join uses all those columns as the join condition (combined with AND), and each of those join columns is deduplicated in the output. This makes the USING form concise when you're joining on columns that share the same names across tables.

7. What SQL operation retrieves combined data from related tables?

- A. SELECT to retrieve combined data
- B. JOIN to retrieve combined data**
- C. MERGE to retrieve combined data
- D. UNION to retrieve combined data

To pull together data from two related tables, you use a join. A join links rows from one table to rows in another based on a shared key (like a customer ID) and returns a single result set that includes columns from both tables. This is what lets you see related information side by side, such as a customer's details alongside their orders. Other options don't fit because: selecting alone fetches data from one table or requires separate queries to combine data; union stacks results from separate queries with the same columns rather than linking related rows; and merge is a data-modification operation used to upsert or synchronize tables, not to retrieve combined data.

8. What is a Cross-Join?

- A. A Cartesian product**
- B. A natural join
- C. An inner join
- D. A left join

A cross join returns the Cartesian product of two tables, meaning every row from the first table is paired with every row from the second. If table A has m rows and table B has n rows, the result contains $m \times n$ rows. There's no join condition involved; it simply combines all possible row pairs. In SQL this is written as CROSS JOIN or, in older syntax, listing both tables in the FROM clause with no WHERE clause. This differs from a natural join, which automatically matches on columns with the same names and only returns rows where those values align (and it also drops duplicate columns). An inner join requires an explicit join condition (ON or USING) to determine which rows to combine. A left join returns all rows from the left table, with matching rows from the right where the condition holds, or nulls when there is no match. So the cross join is the Cartesian product.

9. What is a Self-Join?

- A. A join of a table with two different tables
- B. A join using aggregate
- C. A join with a subquery
- D. A self-join is a join of a table with itself using aliases**

A self-join is a join of a table with itself, usually done by giving two different aliases to the same table in the FROM clause. This lets you treat two references to the same data as if they were two separate tables, so you can relate rows within that one table. The key is using aliases to distinguish the two roles, and writing the join condition to compare columns from each alias—for example, linking an employee's record to their manager's record by matching the employee's manager_id with the manager's employee_id. This technique is exactly what the correct option describes. The other ideas describe different concepts—joining two different tables, applying an aggregate, or using a subquery—none of which capture the idea of pairing rows within the same table via aliases.

10. In MongoDB, embedding is storing related data as a nested document. Which option describes embedding?

- A. Referencing stores related data as a nested document**
- B. Embedding stores related data as a nested document**
- C. Embedding stores data in a separate collection**
- D. Referencing stores data in the same document**

Embedding means placing related data inside the same document as a nested structure. In MongoDB, this is done by storing a sub-document (or an array of sub-documents) directly within the parent document. This makes reading and updating the related data faster and atomic since you can treat it as one unit without needing to fetch multiple documents or collections. It also improves data locality and can simplify data modeling when the related data is tightly coupled. However, embedding can lead to larger documents and is constrained by the 16 MB document size limit, so very large or unbounded growth of embedded data is not suitable. The option that describes embedding is the one that states related data is stored as a nested document; the other choices describe storing data in separate documents or collections or using references, which are alternatives to embedding.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://ddrdatascienceinterview.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE