

Test Results and Interview Guide

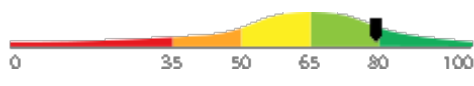
Candidate: **Elizabeth Wantsajob**
Assessment: SQL - Advanced
Completed: June 28, 2026
Prepared for: Sara Maple
Example Company

What's Included

- Overall Score
- Competency Summary Table
- Comparison Matrix
- Detailed Competency Results with Interview Guide

Important Note: The SQL - Advanced assessment measures key factors related to high performance and tenure in this job. Attribute types measured vary by test, but can include cognitive ability, skills, knowledge, personality characteristics, emotional intelligence, and past behavioral history. This report includes a one page summary, followed by detailed results with an embedded interview guide. Note that these results should always be used as a part of a balanced candidate selection process that includes independent evaluation steps, such as interviews and reference checks.

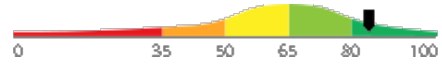
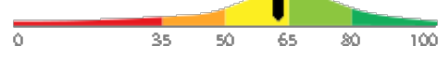


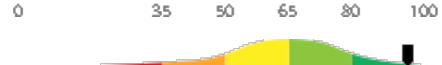
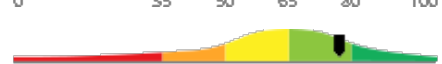
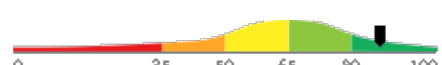

Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com SQL - Advanced June 28, 2026 <p>The candidate demonstrates a solid and broad understanding of advanced SQL, showing proficiency across most key areas including subqueries, window functions, joins, set operations, and transaction control. Minor gaps may exist in specialized topics such as execution plan interpretation, correlated subqueries, or advanced schema design, but overall competence is strong for most advanced database programming responsibilities.</p>	<div style="background-color: #008000; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">79</div>	

Key


- Candidate Score
- Higher Risk
- Lower Risk

Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
DDL: Tables, Constraints, Indexes, and Views	84	
Joins, Set Operations, and Grouping (Coding Tasks)	62	
Subqueries, CTEs, and Correlated Queries (Coding Tasks)	62	
Joins, Set Operations, and Grouping	91	
Query Performance and Execution Plans	75	
Subqueries, CTEs, and Correlated Queries	93	
Transaction Control and Data Integrity	77	
Window Functions	86	

Comparison

Percentile scores indicate how the candidate compares to other test-takers within various groups. The candidate scored equal to or better than the fraction of test-takers indicated by the percentile.

Test-Taker Group	Percentile	0	10	20	30	40	50	60	70	80	90	100	
Global	79th												
North America	65th												
United States	65th												
Example Company	73rd												

Artificial Intelligence (AI) Generated Scores

This table includes one or more scores derived from a large language model AI query. AI-derived scores are non-deterministic. That is, they are not precisely repeatable. Therefore, these scores should always be treated as supplementary information and should never be used exclusively or compared to hard cutoff values.

Estimated Value	Score	Confidence	Interpretation
Knowledge, Skills, and Abilities Summary	-	-	<p>Summary Points (AI):</p> <ul style="list-style-type: none"> (Generic Text for Sample Report) Strong performer in Drag and Drop Files tasks, indicating comfort with file management and basic computer interactions. Demonstrates solid numerical accuracy in Recognizing and Confirming Numbers, a valuable asset in detail-oriented roles. Moderate overall performance in Analytical Thinking and Attention to Detail, with adequate grammar skills but room for improvement. Struggles with Reading and Analyzing Problems, which may limit effectiveness in roles requiring critical reading and complex problem-solving. Lowest performance in Navigating Between Screens, suggesting difficulty with multi-screen software workflows that could impact productivity in computer-intensive roles. <p>Narrative (AI): Elizabeth Wantsajob demonstrates a mixed profile of knowledge, skills, and abilities across the assessed competencies.</p> <p>Elizabeth shows a strong aptitude in Drag and Drop Files, performing well on this technical task and suggesting she is comfortable with this type of computer interaction. This is a notable strength that would translate well into roles requiring file management and basic computer navigation tasks.</p> <p>In the area of Analytical Thinking and Attention to Detail, Elizabeth performs at a moderate level. She demonstrates solid ability in Recognizing and Confirming Numbers, which suggests she is careful and accurate when working with numerical data — a valuable skill in detail-oriented work environments. Her Grammar performance is adequate but leaves room for improvement, indicating she may occasionally make written communication errors. Her weakest area within this competency is Reading and Analyzing Problems, where she struggled to consistently interpret and work through written problem scenarios. This may impact her effectiveness in roles that require critical reading, written comprehension, or complex problem-solving.</p> <p>Elizabeth's most significant area for development is Navigating Between Screens, where she scored considerably lower than the other competencies. This suggests she may have difficulty efficiently moving through software interfaces or multi-screen workflows, which could slow productivity in roles that rely heavily on navigating computer applications or data entry systems.</p> <p>Overall, Elizabeth brings some useful technical strengths, particularly in file management and numerical accuracy, but would benefit from targeted development in software navigation and analytical problem-solving to be fully effective in roles that demand these skills.</p> <p>Computed on: April 2, 2026, 11:09:49PM EDT</p>

Detail

Candidate: Elizabeth Wantsajob, beth.wantsajob@gmail.com
 Assessment: SQL - Advanced
 Authorized: June 28, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
 Started: June 28, 2026, 9:16:07AM EDT
 Completed: June 28, 2026, 9:16:07AM EDT
 Overall Score: 79

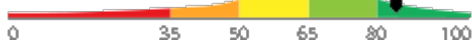
Knowledge and Skills Detail

This section contains a list of job-related knowledge areas and skills that have been evaluated. Low scores in these areas often indicate that additional learning may be required before top performance can be achieved.

Detail
Interview Guide

DDL: Tables, Constraints, Indexes, and Views

Score: 84



Description:

Covers writing Data Definition Language (DDL) statements to create, alter, and drop tables, as well as defining constraints such as PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, and CHECK. Includes creating indexes to improve query performance and defining views to simplify or secure access to underlying data.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate demonstrates a comprehensive and advanced mastery of DDL, including precise use of table management statements, the full range of constraint types, and strategic implementation of indexes to optimize performance. They show a strong ability to design and utilize views to both simplify and secure access to underlying data, reflecting expert-level proficiency in SQL data definition.

You are designing a table for a high-traffic application and need to decide which columns to index. What factors would you consider, and what are the trade-offs of adding too many indexes?



1

Cannot explain indexing strategy or is unaware of trade-offs like slower writes and increased storage.



2

Mentions indexing frequently queried or joined columns but gives only a surface-level explanation of trade-offs.



3



4

Discusses query patterns, selectivity, covering indexes, and clearly explains write overhead and maintenance trade-offs.



5

What is the difference between a PRIMARY KEY and a UNIQUE constraint? Can you give an example of when you would use each?



1

Cannot distinguish between the two or gives an incorrect explanation.



2

Correctly identifies that both enforce uniqueness but cannot explain the NULL behavior difference or practical use cases.



3



4

Clearly explains that PRIMARY KEY implies NOT NULL and uniqueness, while UNIQUE allows NULLs, with practical examples.

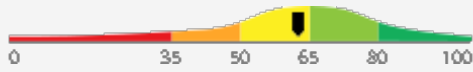


5

Detail Interview Guide

Joins, Set Operations, and Grouping (Coding Tasks)

Score: 62



Description:

Covers the use of pointers to reference and manipulate memory addresses, along with dynamic memory allocation and deallocation using malloc, calloc, realloc, and free. Includes pointer arithmetic, dereferencing, and avoiding common issues like memory leaks and dangling pointers.

Interpretation:

The candidate exhibits average writing skills, which can hinder high performance in some jobs.

The candidate possesses a moderate working knowledge of C programming, demonstrating familiarity with core concepts including data types, control flow, functions, and basic file I/O. They may require some guidance when working with more advanced topics such as dynamic memory allocation, modular design, or debugging complex logic.

Overall AI Score:	65.0
Lines of Code:	15.0
Syntax Errors:	5.0
AI Confidence Level:	50
Match with Ideal Response (AI):	30.0
Structure:	50.0
Syntax:	30.0

Please see below to view the essay submitted.

Walk me through how you would dynamically allocate memory for an array of 10 integers, use it, and then properly release it. What issues might arise if you don't follow best practices?



1

Cannot write correct allocation code; unaware of free() or memory leak risks.



2

Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.



3



4

Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.



5

Can you explain what a pointer is in C and describe a situation where you would use one?



1

Vague or incorrect definition; cannot describe a practical use case.



2

Correct basic definition; gives a simple but valid use case with some gaps.



3



4

Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.



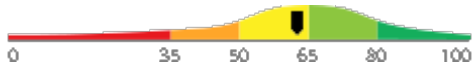
5

Detail

Interview Guide

Subqueries, CTEs, and Correlated Queries (Coding Tasks)

Score: 62



Description:

Covers the use of pointers to reference and manipulate memory addresses, along with dynamic memory allocation and deallocation using malloc, calloc, realloc, and free. Includes pointer arithmetic, dereferencing, and avoiding common issues like memory leaks and dangling pointers.

Interpretation:

The candidate exhibits average writing skills, which can hinder high performance in some jobs.

The candidate possesses a moderate working knowledge of C programming, demonstrating familiarity with core concepts including data types, control flow, functions, and basic file I/O. They may require some guidance when working with more advanced topics such as dynamic memory allocation, modular design, or debugging complex logic.

Overall AI Score:	65.0
Lines of Code:	15.0
Syntax Errors:	5.0
AI Confidence Level:	50
Match with Ideal Response (AI):	30.0
Structure:	50.0
Syntax:	30.0

Please see below to view the essay submitted.

Walk me through how you would dynamically allocate memory for an array of 10 integers, use it, and then properly release it. What issues might arise if you don't follow best practices?



1

Cannot write correct allocation code; unaware of free() or memory leak risks.



2

Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.



3



4

Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.



5

Can you explain what a pointer is in C and describe a situation where you would use one?



1

Vague or incorrect definition; cannot describe a practical use case.



2

Correct basic definition; gives a simple but valid use case with some gaps.



3



4

Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

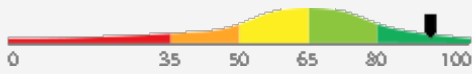


5

Detail Interview Guide

Joins, Set Operations, and Grouping

Score: 91



Description:

Covers the use of INNER, LEFT, RIGHT, and FULL joins to combine data from multiple tables, as well as set operations like UNION, INTERSECT, and EXCEPT. Includes grouping data with GROUP BY, filtering groups with HAVING, and using aggregate functions like COUNT, SUM, AVG, MIN, and MAX.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate exhibits a comprehensive and expert-level mastery of advanced SQL keywords, syntax, and design patterns across all major subject areas, including recursive queries, window functions, performance optimization, normalization, triggers, and stored procedures. They are well-equipped to independently design, implement, debug, and optimize complex database solutions at a high level of proficiency.

You need to find all departments that have more than 10 employees earning above \$50,000. How would you write that query, and what clauses would you use to filter the grouped results?



1

Cannot construct the query or confuses WHERE and HAVING for filtering grouped data.



2

Writes a mostly correct query using GROUP BY and HAVING but may have minor syntax errors.



3



4

Writes a clean, correct query using JOIN, GROUP BY, HAVING, and aggregate functions with clear explanation.



5

Can you explain the difference between an INNER JOIN and a LEFT JOIN? Can you give an example of when you would use each?



1

Confuses join types or cannot explain the difference clearly.



2

Correctly explains the difference but gives a vague or generic example.



3



4

Clearly explains both, gives concrete practical examples, and mentions NULL behavior in LEFT JOIN.

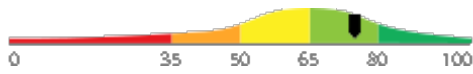


5

Detail Interview Guide

Query Performance and Execution Plans

Score: 75



Description:

Covers identifying and resolving slow queries by analyzing execution plans, recognizing common performance anti-patterns such as full table scans and inefficient subqueries, and applying strategies like query restructuring, index optimization, and avoiding unnecessary computations to improve performance.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid and competent understanding of query performance analysis and execution plan interpretation. They are generally proficient in recognizing anti-patterns such as full table scans and inefficient subqueries, and can apply effective optimization strategies including index improvements and query restructuring in most scenarios.

A query that joins three large tables and uses several subqueries is taking too long to run. Walk me through how you would diagnose and optimize it.



1

Offers only vague suggestions like 'add an index' without a structured diagnostic approach.



2

Checks the execution plan and suggests indexing join columns and replacing subqueries with CTEs or joins.



3



4

Systematically diagnoses using execution plans, addresses join order, index coverage, subquery replacement, and statistics updates.



5

What is a query execution plan, and how would you use it to identify why a query is running slowly?



1

Cannot explain what an execution plan is or how to access and interpret one.



2

Knows what an execution plan is and can identify a full table scan but cannot interpret cost estimates or suggest fixes.



3



4

Explains execution plans clearly, identifies key indicators like scans vs. seeks, and describes actionable optimization steps.



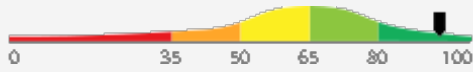
5

Detail

Interview Guide

Subqueries, CTEs, and Correlated Queries

Score: 93



Description:

Covers writing subqueries in SELECT, FROM, and WHERE clauses, as well as common table expressions (CTEs) using the WITH keyword for readable, reusable query logic. Includes correlated subqueries and the use of EXISTS and IN for filtering rows based on related data in another table.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate exhibits an advanced and comprehensive mastery of subqueries, CTEs, and correlated queries in SQL. They are highly proficient in crafting readable, efficient, and reusable query logic using the WITH keyword, nested subqueries across all clause types, and sophisticated filtering techniques including correlated subqueries with EXISTS and IN.

How would you rewrite a complex nested subquery as a CTE, and what are the advantages of doing so? Can you walk through an example?



1

Cannot explain CTEs or how to refactor a subquery into one.



2

Explains the CTE syntax correctly but gives a simple example without discussing readability or reuse benefits.



3



4

Demonstrates CTE syntax fluently, explains readability and reuse advantages, and walks through a meaningful example.



5

What is a subquery, and can you describe a situation where you would use one instead of a join?



1

Cannot define a subquery or describe a meaningful use case.



2

Defines a subquery correctly but struggles to articulate when to prefer it over a join.



3



4

Clearly defines subqueries, explains correlated vs. non-correlated, and gives a practical use case with reasoning.



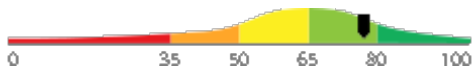
5

Detail

Interview Guide

Transaction Control and Data Integrity

Score: 77



Description:

Covers the use of COMMIT, ROLLBACK, and SAVEPOINT to manage database transactions and ensure data consistency. Includes understanding of isolation levels (READ COMMITTED, REPEATABLE READ, SERIALIZABLE, etc.) and how locking behavior affects concurrent access and prevents issues like dirty reads and phantom reads.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid and proficient understanding of SQL transaction control and data integrity, including the practical use of transaction commands and a working knowledge of isolation levels and their effects on concurrent access. They are generally capable of identifying and addressing common concurrency issues in a database environment.

A colleague reports that two concurrent transactions are producing inconsistent results. How would you investigate whether the isolation level is contributing to the problem, and what changes might you make?



1

Cannot explain isolation levels or their effect on concurrency issues.



2

Identifies isolation levels as relevant and names a few but cannot precisely map them to specific concurrency problems.



3



4

Explains dirty reads, phantom reads, and non-repeatable reads, maps them to isolation levels, and recommends a specific fix.



5

Can you explain what a database transaction is and why COMMIT and ROLLBACK are important?



1

Cannot define a transaction or explain the purpose of COMMIT and ROLLBACK.



2

Correctly defines transactions and the commands but cannot explain real-world implications of missing them.



3



4

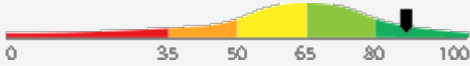
Defines transactions clearly, explains atomicity and consistency, and gives a practical example of why rollback matters.



5

Detail
Interview Guide
Window Functions

Score: 86


Description:

Covers the use of window functions such as ROW_NUMBER, RANK, DENSE_RANK, LEAD, LAG, and aggregate functions with OVER() clauses. Includes defining partitions with PARTITION BY and ordering results within windows using ORDER BY to perform calculations across rows without collapsing them into groups.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate exhibits an advanced and comprehensive mastery of SQL window functions, including proficient use of ranking, offset, and aggregate functions with OVER() clauses. They are highly skilled at defining partitions, ordering results within windows, and leveraging these techniques to perform sophisticated row-level calculations across complex datasets.

How would you use a window function to return each employee's salary along with their rank within their department, and also show the difference between their salary and the highest salary in their department?



1

Cannot write the query or misuses PARTITION BY and ORDER BY within OVER().



2

Writes a mostly correct query using RANK() and PARTITION BY but struggles with the salary difference calculation.



3



4

Writes a clean query using RANK() OVER(PARTITION BY ... ORDER BY ...) and MAX() as a window function for the difference.



5

What is a window function, and how is it different from using GROUP BY with an aggregate function?



1

Cannot explain window functions or confuses them with GROUP BY aggregations.



2

Explains the basic concept but cannot clearly articulate the key difference regarding row preservation.



3



4

Clearly explains that window functions calculate across rows without collapsing them, with a concrete example.



5

IT Coding Tasks

During the assessment, the candidate was asked to write one or more programs or scripts. Their responses are included below for review.

Question or Task	Response
<p>Complete the provided partial C program by filling in the missing sections marked with TODO comments. Your completion must use standard C keywords and libraries.</p> <p>The program declares a function called <code>duplicate_array</code> that:</p> <ol style="list-style-type: none"> 1. Takes a const int pointer to a source array and its length as parameters. 2. Uses <code>calloc</code> to allocate a new int array of the same length. 3. Returns NULL if <code>calloc</code> fails. 4. Copies each element from the source array into the new array using pointer arithmetic (not array subscript notation). 5. Returns the pointer to the newly allocated copy. <p>In main, the program:</p> <ol style="list-style-type: none"> 1. Declares and initializes a stack array of 4 integers with values 5, 15, 25, 35. 2. Calls <code>duplicate_array</code> to create a heap-allocated copy. 3. Checks for NULL and prints an error and returns 1 if the call failed. 4. Prints each element of the duplicate using a loop. 5. Frees the duplicate array. <p>Use appropriate indentation, common C coding conventions, and add brief inline comments where needed. Type your completed source code as your response.</p>	<pre>#include <stdio.h> #include <stdlib.h> int *duplicate_array(const int *src, int length) { /* TODO: Use calloc to allocate a new array of 'length' integers, return NULL if calloc fails, copy elements from src using pointer arithmetic, and return the new pointer. */ calloc(303); } int main(void) { /* TODO: Declare and initialize a stack array of 4 integers: 5, 15, 25, 35, then call duplicate_array and store the result. Check for NULL and print an error message returning 1 if it failed. */ array[4]={5,15,25,35}; int i; /* Print each element of the duplicate */ for (i = 0; i < 4; i++) { printf("duplicate[%d] = %d\n", i, *(duplicate + i)); } /* Free the duplicate array */ free(duplicate); return 0; }</pre>

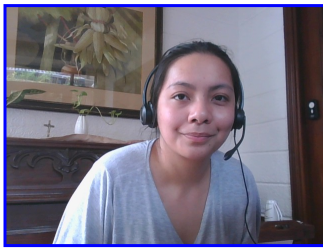
Comments (AI): The code segment has several syntax errors and incomplete implementation. The `duplicate_array` function does not correctly allocate memory or copy elements. The main function has syntax errors and does not properly call the `duplicate_array` function. However, the structure and intent of the code are somewhat clear, and the code attempts to follow the requirements.

Identity Confirmation Photos

The following photos of the candidate and any identification were uploaded during the assessment session.

Photo Analysis Results

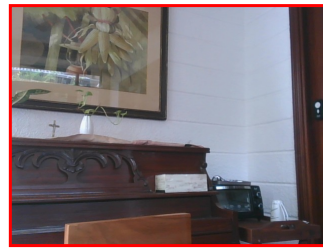
- Risk:	Medium risk of cheating based on image inconsistencies
- Percent match among processed faces	100%
- Total images processed	17
- Total images with valid faces	14 (82%)
- Total pairs of faces compared	13
- Pairs in which faces matched	13 (100%)



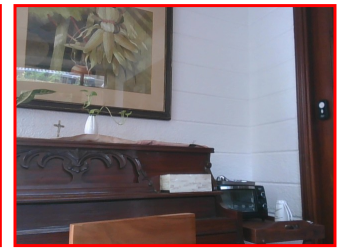
Pre/Post-Test Photo



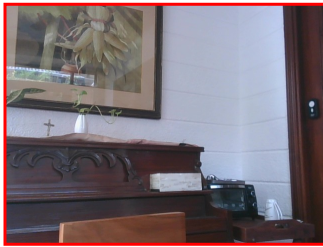
ID Photo



In-Test Error Detected (No Face Detected)



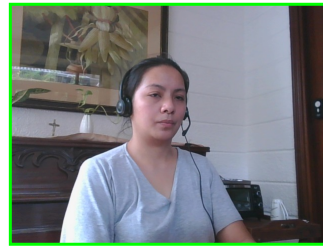
In-Test Error Detected (No Face Detected)



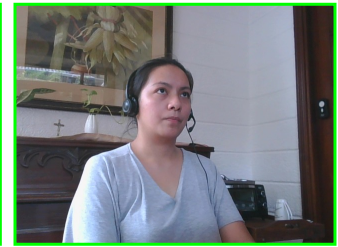
In-Test Error Detected (No Face Detected)



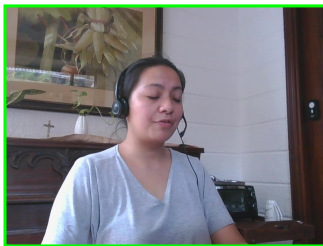
In-Test Photo



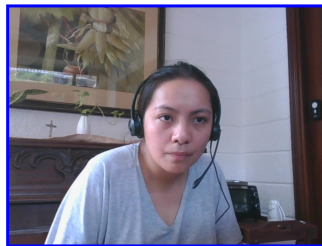
In-Test Photo



In-Test Photo



In-Test Photo



Pre/Post-Test Photo

Resume or CV

[Summary](#)[Updated on](#)

Motivated career professional with extensive experience in office administration and management. Proven track record of improving efficiency, reducing costs, and enhancing office operations through strategic initiatives and technology implementation.

Objective

I am seeking a role where I can use my many skills and my exceptional judgment and empathy for customers to make a difference to a growing company.

Education

- Associate of Applied Science in Office Administration, Portland Community College, 2020

Experience

- General Office Clerk, Paramount Office Management, 09/2023 – Present
- Administrative Assistant, Global Enterprises Inc., 04/2021 – 08/2023
- Administrative Assistant, Innovative Business Solutions Ltd., 07/2019 – 03/2021

Other Qualifications

- Microsoft Office Specialist (MOS) Certification
- Certified Administrative Professional (CAP)
- International Association of Administrative Professionals (IAAP) Certification

Report Preparation Notes

- Hiring decisions should never be based on a single source of information. The most effective use of this assessment report is as a part of a multi-faceted program of candidate evaluation that includes resume review, interviews, and reference checks.
- Overall vs Percentiles Scores: The overall score reflects the success in the test, based on the mean (average) and standard deviation of the test scores. The percentile score reflects the percentage of test-takers who scored equal or below this overall score. We recommend you use the Overall Score as your primary evaluation criteria. However, percentile scores can often be useful in comparing specific candidates against one another and with a group, such as for test takers in a certain organization or within a certain account.
- Note that comparison information is calculated based on completed instances of this assessment at that time the assessment is scored. As additional instances are completed, the comparative data may change. You can always update a report to the current values by clicking on 'Recalculate Percentiles' within the online results viewing pages at www.hravatar.com.
- Most competency scores are norm-based, which means that they can be interpreted in terms of their distance from the average or mean score. For all scales, a score equal to the mean receives a score of 65 and scores above and below this value are set so that a score change of 15 equals one standard deviation.
- For linear competencies, higher is better across the entire scale. For these scales a score between 65 and 80 (light green) represents 0 to 1 standard deviation above the mean and a score above 80 (dark green) represents more than one standard deviation above the mean. Similarly, a score of 50 - 65 (yellow) represents 0 to 1 standard deviation below the mean, while a score of 35 - 50 (orange) equates to 1 to 2 standard deviations below the mean, and a score below 35 represents more than 2 standard deviations below the mean.
- Sim ID: 20806-1, Key: 0-0, Rpt: 68, Prd: 9628, Created: 2026-06-28 09:16 EDT
- UA: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; Touch; rv:11.0) like Gecko

Score Calculation Detail

The following table provides a summary of how the overall score was calculated from each of the individual competency scores. First, all competency scores are calculated on a scale of 0-100. Note that some competencies use their color category rather than their actual numeric score in the overall calculation. For these, a standard score associated with the assigned color category is used in the overall score calculation rather than the actual numeric score. This is reflected in the "Score Value Used" column. Next, a weighted average of scores is computed using individual competency weights, typically set using job analysis data provided by the US Government Occupational Information Network (O*Net).

Competency	Score	How applied to overall	Score Value Used	Weight (%)
DDL: Tables, Constraints, Indexes, and Views	84.5007	Numeric Score	84.5007	12.5000
Joins, Set Operations, and Grouping	91.6704	Numeric Score	91.6704	12.5000
Joins, Set Operations, and Grouping (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Query Performance and Execution Plans	75.2152	Numeric Score	75.2152	12.5000
Subqueries, CTEs, and Correlated Queries	93.5184	Numeric Score	93.5184	12.5000
Subqueries, CTEs, and Correlated Queries (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Transaction Control and Data Integrity	77.5844	Numeric Score	77.5844	12.5000
Window Functions	86.9373	Numeric Score	86.9373	12.5000
Weighted Average:				79.4229
Final Overall Score:				79

Notes

(This area is intentionally blank - it's reserved as space for your notes.)