



I Introduction to SQL

CERTIFICATION OBJECTIVES

- | | | | |
|------|---|------|--|
| I.01 | The Exam: An Overview | I.05 | Understand the Unique Role of SQL in Modern Software Systems |
| I.02 | Define and Understand the Basics of the RDBMS | I.06 | Confirm Appropriate Materials for Study |
| I.03 | Define and Understand the Basics of SQL | ✓ | Two-Minute Drill |
| I.04 | Understand the Oracle RDBMS and Oracle SQL | Q&A | Self Test |

Oracle Corporation's implementation of the Structured Query Language, or SQL, is arguably the most powerful and most significant computer language used in the world of government and business today. This chapter begins the process of preparing you to successfully take and pass the Oracle 1Z0-047 exam, titled "Oracle Database SQL Expert". First we'll discuss a few particulars about the exam itself, how it's different from other Oracle certification tests, and what you can expect when you take it. Then we'll begin to address the SQL language; we'll introduce some background information and prepare you to go through a comprehensive analysis and review of the language in order to successfully study for the test. We will explore the background of SQL and its role in the world of computer languages and software development.

If you are a veteran Oracle professional and are taking a bare-bones approach to exam preparation, you might want to read the first section, which provides an overview of the exam, and then perhaps skim through the rest of the chapter, looking for the Exam Watch sections along the way. However, I encourage you to review all of this material, for it will help you to position your thinking with regard to the exam, as well as your career. If you aren't absolutely crystal clear on how prominent SQL is in the marketplace, or how important it is that you have a comprehensive and thorough understanding of all of the capabilities of SQL, then read on. If nothing else, this chapter will help galvanize your career development by informing (or reminding) you of how increasingly crucial it is that you maintain your SQL skills at the highest level, as this is what organizations in the world today require—and this requirement grows with each day, as databases grow, and as the use and potential of the data they contain continue to increase in significance.

CERTIFICATION OBJECTIVE 1.01

The Exam: An Overview

A typical Oracle professional doesn't generally begin his or her career by taking the advanced "1Z0-047 Oracle Database SQL Expert" exam. Since you're reading this book, chances are you've probably taken some of Oracle's other certification tests, such as "1Z0-051 SQL Fundamentals I". The "SQL Expert" exam builds on

the “SQL Fundamentals I” exam in terms of both subject matter and complexity. But “SQL Expert” is a very different and unique exam. It is demanding and asks questions that test the full breadth of your knowledge of SQL syntax and processing, and its application to business rules.

A typical question on the SQL Expert exam might go something like this:

- You’ll be asked to review an exhibit, which could be a set of data output in a half-dozen columns and perhaps 20 or 30 rows—or it might be an entity-relationship diagram (ERD) containing as many as a half-dozen entities or more.
- Next you’ll need to review a set of SQL statements that are intended to operate on the exhibit you were just shown, with a number of SQL statements, in which there might be a series of nested scalar functions, aggregate functions, multitable indices, subqueries of various forms, and/or the use of different statements and clauses showcasing features such as complex timezone usage, very large datatypes, complex join conditions, etc.
- Some of the code may be correct—some may not, and you’ll need to recognize the difference.
- With the sample exhibit and SQL code in front of you, you may be asked to identify the resulting status of the database after the SQL statements execute.
- You may be asked to identify the internal workings of the Oracle database, in order, in accordance with the SQL statements you’ve been shown.
- The list of possible answers may include more than one correct response, and you must identify each of them.

Does that sound like a lot to do for a single question? Then consider this: for the entire exam, you are allowed 120 minutes to answer 70 questions. That’s an average of 1.71 minutes per question.

Think you can handle it? Do you have what it takes to be a formally recognized and officially certified Oracle Database SQL Expert?

Whether you do or not remains to be seen . . . but one thing is for certain. This book will prepare you, strengthen your knowledge, fill in the gaps, and dramatically increase your odds of success.

So get ready for a fun and rewarding challenge and an important milestone in your career. Get ready to enter the world of the technical elite, to join the *crème de la crème*, to be ranked with the best of the best.

Get ready, for here starts your path to become . . . a certified Oracle Database SQL Expert.

Time’s a-wastin’ . . . let’s get started . . . first, we’ll take a detailed look at the official certification objectives of the SQL Expert exam and compare them with the SQL Fundamentals I exam.

“SQL Fundamentals I” Versus “SQL Expert”

As I just mentioned, since you’re planning on obtaining your SQL Expert certification by taking the “1Z0-047 Oracle Database SQL Expert” exam, then chances are you may have already taken another exam, titled “1Z0-051, SQL Fundamentals I”. The two exams share some common objectives, but 047 goes far beyond 051. See Table 1-1 for a comparison of those objectives, and a detailed analysis of where the two exams are similar, and where they are different.

TABLE 1-1 Comparison: 1Z0-051 and 1Z0-047 Exam Objectives	1Z0-051 SQL Fundamentals I	1Z0-047 SQL Expert	Exam Objectives
	1.0	1.0	Retrieving Data Using the SQL SELECT Statement
	1.1	1.1	List the capabilities of SQL SELECT statements
	1.2	1.2	Execute a basic SELECT statement
	—	1.3	Describe how schema objects work
	2.0	2.0	Restricting and Sorting Data
	2.1	2.1	Limit the rows that are retrieved by a query
	2.2	2.2	Sort the rows that are retrieved by a query
	2.3	—	Use ampersand substitution to restrict and sort output at runtime
	3.0	3.0	Using Single-Row Functions to Customize Output
	3.1	3.1	Describe various types of functions that are available in SQL
	3.2	3.2	Use character, number, and date functions in SELECT statements
	(*)	3.3	Describe the use of conversion functions

TABLE I-1

Comparison:
IZ0-051 and
IZ0-047 Exam
Objectives
(Continued)

IZ0-051 SQL Fundamentals I	IZ0-047 SQL Expert	Exam Objectives
4.0	(*)	Using Conversion Functions and Conditional Expressions
4.1	(*)	<i>Describe various types of conversion functions that are available in SQL</i>
4.2	(*)	<i>Use the TO_CHAR, TO_NUMBER, and TO_DATE conversion functions</i>
4.3	—	<i>Apply conditional expressions in a SELECT statement</i>
5.0	4.0	Reporting Aggregated Data Using the Group Functions
5.1	4.1	<i>Identify the available group functions</i>
5.2	4.2	<i>Describe the use of group functions</i>
5.3	4.3	<i>Group data by using the GROUP BY clause</i>
5.4	4.4	<i>Include or exclude grouped rows by using the HAVING clause</i>
6.0	5.0	Displaying Data from Multiple Tables
6.1	5.1	<i>Write SELECT statements to access data from more than one table using equijoins and nonequijoins</i>
6.2	5.2	<i>Join a table to itself by using a self-join</i>
6.3	5.3	<i>View data that generally does not meet a join condition by using outer joins</i>
6.4	5.4	<i>Generate a Cartesian product of all rows from two or more tables</i>
7.0	6.0	Using Subqueries to Solve Queries
7.1	6.1	<i>Define subqueries</i>
7.2	6.2	<i>Describe the types of problems that subqueries can solve</i>
7.3	6.3	<i>List the types of subqueries</i>
7.4	6.4	<i>Write single-row and multiple-row subqueries</i>

(Continued)

TABLE I-1

Comparison:
IZ0-051 and
IZ0-047 Exam
Objectives
(Continued)

IZ0-051 SQL Fundamentals I	IZ0-047 SQL Expert	Exam Objectives
8.0	7.0	Using the Set Operators
8.1	7.1	<i>Describe set operators</i>
8.2	7.2	<i>Use a set operator to combine multiple queries into a single query</i>
8.3	7.3	<i>Control the order of rows returned</i>
9.0	8.0	Manipulating Data
9.1	8.1	<i>Describe each data manipulation language (DML) statement</i>
9.2	8.2	<i>Insert rows into a table</i>
9.3	8.3	<i>Update rows in a table</i>
9.4	8.4	<i>Delete rows from a table</i>
9.5	8.5	<i>Control transactions</i>
10.0	9.0	Using DDL Statements to Create and Manage Tables
10.1	9.1	<i>Categorize the main database objects</i>
10.2	9.2	<i>Review the table structure</i>
10.3	9.3	<i>List the data types that are available for columns</i>
10.4	9.4	<i>Create a simple table</i>
10.5	9.5	<i>Explain how constraints are created at the time of table creation</i>
10.6 (**)	1.3 (**)	<i>Describe how schema objects work</i>
11.0	10.0	Creating Other Schema Objects
11.1	10.1	<i>Create simple and complex views</i>
11.2	10.2	<i>Retrieve data from views</i>
11.3	10.3	<i>Create, maintain, and use sequences</i>
11.4	10.4	<i>Create and maintain indexes</i>
11.5	10.5	<i>Create private and public synonyms</i>

TABLE I-1

Comparison:
IZ0-051 and
IZ0-047 Exam
Objectives
(Continued)

IZ0-051 SQL Fundamentals I	IZ0-047 SQL Expert	Exam Objectives
—	11.0	Managing Objects with Data Dictionary Views
—	11.1	<i>Use the data dictionary views to research data on your objects</i>
—	11.2	<i>Query various data dictionary views</i>
—	12.0	Controlling User Access
—	12.1	<i>Differentiate system privileges from object privileges</i>
—	12.2	<i>Grant privileges on tables</i>
—	12.3	<i>View privileges in the data dictionary</i>
—	12.4	<i>Grant roles</i>
—	12.5	<i>Distinguish between privileges and roles</i>
—	13.0	Managing Schema Objects
—	13.1	<i>Add constraints</i>
—	13.2	<i>Create indexes</i>
—	13.3	<i>Create indexes using the CREATE TABLE statement</i>
—	13.4	<i>Creating function-based indexes</i>
—	13.5	<i>Drop columns and set column UNUSED</i>
—	13.6	<i>Perform FLASHBACK operations</i>
—	13.7	<i>Create and use external tables</i>
—	14.0	Manipulating Large Data Sets
—	14.1	<i>Manipulate data using subqueries</i>
—	14.2	<i>Describe the features of multitable INSERTs</i>
—	14.3	<i>Use the following types of multitable INSERTs (Unconditional, Conditional and Pivot)</i>
—	14.4	<i>Merge rows in a table</i>
—	14.5	<i>Track the changes to data over a period of time</i>

(Continued)

TABLE I-1

Comparison:
IZ0-051 and
IZ0-047 Exam
Objectives
(Continued)

IZ0-051 SQL Fundamentals I	IZ0-047 SQL Expert	Exam Objectives
—	15.0	Generating Reports by Grouping Related Data
—	15.1	<i>Use the ROLLUP operation to produce subtotal values</i>
—	15.2	<i>Use the CUBE operation to produce crosstabulation values</i>
—	15.3	<i>Use the GROUPING function to identify the row values created by ROLLUP or CUBE</i>
—	15.4	<i>Use GROUPING SETS to produce a single result set</i>
—	16.0	Managing Data in Different Time Zones
—	16.1	<i>Use various datetime functions</i>
—	17.0	Retrieving Data Using Subqueries
—	17.1	<i>Write a multiple-column subquery</i>
—	17.2	<i>Use scalar subqueries in SQL</i>
—	17.3	<i>Solve problems with correlated subqueries</i>
—	17.4	<i>Update and delete rows using correlated subqueries</i>
—	17.5	<i>Use the EXISTS and NOT EXISTS operators</i>
—	17.6	<i>Use the WITH clause</i>
—	18.0	Hierarchical Retrieval
—	18.1	<i>Interpret the concept of a hierarchical query</i>
—	18.2	<i>Create a tree-structured report</i>
—	18.3	<i>Format hierarchical data</i>
—	18.4	<i>Exclude branches from the tree structure</i>
—	19.0	Regular Expression Support
—	19.1	<i>Using meta characters</i>
—	19.2	<i>Regular expression functions</i>
—	19.3	<i>Replacing patterns</i>
—	19.4	<i>Regular expressions and check constraints</i>

* Note that conversion functions are addressed by both exams, but with more emphasis in 051 versus 047.

** This is a repeat of item 1.3.

As you can see, both exams look at many of the same features, such as the SELECT statement and its ability to sort rows and convert datatypes, together with its use of functions and expressions; the GROUP BY clause; joining tables; subqueries; the set operators (such as UNION and INTERSECT); the INSERT, UPDATE, and DELETE statements; creating database objects, and more.

All of those topics are on both exams. But SQL Expert goes far beyond this. The SQL Expert exam also addresses topics such as

- The data dictionary
- User access with roles and privileges
- FLASHBACK operations
- External tables
- Function-based indexes
- Constraints
- Multitable INSERTs
- Tracking changes over time
- Conditional INSERTs with pivots
- CUBE
- ROLLUP operations
- GROUPING SETS
- Managing data across multiple time zones
- EXISTS and NOT EXISTS
- The WITH clause
- Multi-column, scalar, and correlated subqueries
- Hierarchical SELECT
- Tree-structured output
- Regular expressions in functions and check constraints
- And more

As you can see, the SQL Expert exam goes much further than the topics addressed by the SQL Fundamentals I exam.

Both exams consist of 70 questions, and two hours are afforded the test taker of either exam. At the time of this writing, the passing scores are published differently at the Oracle.com web site: for “SQL Fundamentals I” it is 60 percent, while “SQL Expert” requires 66 percent. However, note that passing score requirements

are subject to change without notice. Oracle Corporation reserves the right to substitute a particular version of the test with another version, and depending on the complexity of the specific questions included in a new version, the passing score may be adjusted accordingly. In fact, Oracle publishes this notice on its web site with regarding to the required passing score for any given exam:

The passing scores provided on the Oracle Certification Program website are for informational purposes only. Oracle does not recommend an exam preparation strategy targeting the passing score, because passing scores are subject to change without notice.

In other words: study well, and don't plan on trying to achieve the minimal passing score requirement. Instead do the best you possibly can in order to increase your chances of victory.

What to Expect

I've taken the test. Let me share a little with you about what you can expect.

Test Logistics

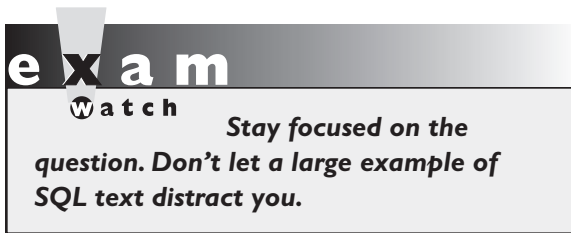
In my case, I went to the Oracle Corporation web site (oracle.com), clicked the Certification link, and looked for the 1Z0-047 exam page. From there I clicked the link asking me to "register" for the exam. This took me to the Prometric web site (www.prometric.com) where I located a local university that was hosting proctored exams. I found an available time, provided my credit card information for the \$125 payment, and a few days later arrived at the testing facility.

When I arrived, I was asked to turn my mobile phone off and give it to the staff, who locked it into a small container. The staff retained the key but handed the container containing my mobile phone to me. I was told I would be able to take the container with me into the testing room, unable to access it inside the locked container. I was told I could recover my phone after the exam.

After providing two forms of ID, I was shown into a large room filled with computers and taken to one that was already logged in to the Prometric automated testing system. I sat down and began. I stepped through a series of disclosures and agreements and finally was presented with the first of what I knew would be 70 questions. The 120-minute timer started with the first question. It clicked off each second in the upper-right corner, and I could monitor it throughout the exam.

All of the questions were multiple-choice. Most of them required me to click a button to display an exhibit, which popped up in a separate window, sized just big enough to show whatever the exhibit was displaying. Generally the exhibit was an entity-relationship diagram, but sometimes it was a listing of data that could've been the contents of a table or the output of a report. The exhibit didn't indicate what it was intended for, but the question would eventually get around to explaining how you were supposed to treat and interpret the exhibit for the question.

Some questions will throw a lot of material at you. You won't necessarily need to analyze every bit of it to answer the question. Be careful with your time—keep an eye on the clock, remember that you have less than two minutes per question on average, and don't get distracted. Remember—your goal is to answer the question, not necessarily to evaluate every line of the code and data element that is presented to you. Some questions will be about one simple concept, and that concept will be shown in the context of a series of several SQL statements or data listings. When you're asked to “look at the exhibit” and “evaluate the following SQL code”, you might want to temporarily ignore all of that and glance ahead at the actual question first, so that you know what you'll be looking for in the exhibit and the code. Otherwise, you'll waste time studying some ERD containing over a half-dozen entities with multiple relationships, plus a half-dozen lines of SQL code, when the question might really center on just one or two of those entities and how they should be joined in a multitable query. So don't get distracted—stay focused on the question, and use your time judiciously.



exam
Watch **Stay focused on the question. Don't let a large example of SQL text distract you.**

The questions were presented on the screen one at a time. I clicked Next to advance to the next question. I wasn't required to answer each question before advancing.

Each of the questions had an optional check box in the upper-left corner labeled “Mark”. I could “mark” any question for future reference, whether I had answered it or not.

When I eventually reached the final question, answered it, and clicked Next, I found myself looking at a summary screen showing the number corresponding to every question of the exam in a singular tabular listing. The questions were identified by number only, and next to each was the letter—or letters—of the answers I had provided. Any question I had “marked” showed a highlighted M next to it. Any question I had not fully answered showed a highlighted I—for

“incomplete”—next to it. I was easily able to review and complete the answers and review any questions, including those I had marked, before completing the exam.

One factor worth noting regarding questions that require more than one correct answer to be identified: some of these questions will tell you exactly how many correct answers you must choose from the set presented to you. For example, one question may have a total of four possible answers, of which two are correct, and the question will tell you to choose two answers. In this example, if you were to choose a third answer, the automated testing system wouldn’t let you do it, but would pop up a small message window telling you to de-select another answer first.

But let’s say you don’t catch the fact that there were two correct answers, and you only click one, and you move on. Nothing in the system will stop you from moving ahead and leaving the question incomplete with—in this example—only one of the two required answers. In fact you’re always allowed to advance and leave any question “incomplete”. But the good news is this: once you reach the summary screen at the end, any incomplete questions will be flagged clearly and you’ll be able to go back and review.

So in case you think you answered everything, don’t be too sure and exit the test prematurely—be sure you take a good look at the summary screen at the end and check for any highlighted “I” markings next to your answers. If you see any and weren’t expecting to, it’s quite possible that you’re dealing with a question that had more correct answers than you thought.

When I was done with my questions and was satisfied that I had answered everything, I clicked Exit on the summary screen, the test score was instantly evaluated, and I was shown my score and passing grade on the screen. In addition, a nearby printer produced a written summary of my performance as well.

I picked up my papers and went to the front desk of the testing center, where a clerk made copies of my papers and kept a set. They unlocked the container containing my mobile phone, which I retrieved, and I left a happy and certified SQL Expert.

Subject Areas

The certification objectives for 1Z0-047 are shown in Table 1-1. They are taken directly from the Oracle Corporation web site as of this writing. They were the same certification objectives at the time that I took the exam.

Warning: the emphasis on the exam is on those objectives not included in the 1Z0-051 exam. That makes sense, of course, but it’s not obvious from the published literature—until now. While all topic areas are addressed in some fashion, and

Oracle Corporation reserves the right to change anything and everything about the exam with no warning at any time, be aware that I've taken this exam, and of the 19 stated certification objective categories for 047, my exam provided a greater emphasis on the nine certification objective categories that are not covered by 051. The test does include questions from all categories—including those categories that are included in 051. But the nine unique certification categories not included in 051 are emphasized in 047.

In addition, you'll need to have an ability to read entity-relationship diagrams in order to take this exam, something that's not specifically mentioned in the list of certification objectives. A large number of questions will ask you to look at an "exhibit", and more often than not the exhibit will be some sort of entity-relationship diagram.

Many questions challenge your knowledge of several facts at once. For example, I encountered one question that presented several nested scalar functions in a series of SELECT statements. I had to understand clearly what each individual scalar function did, recognizing syntax issues, understand the data type transformations as one function passed on results to another, confirm whether the parameter positioning was accurate, and identify two facts about the process and end result, all within the concept of a given ERD.

The moral to the story: study this book well, understand everything listed in the certification objectives, pay special attention to those areas that are not included in 051 but are unique to 047, and get all of your facts down cold. And on test day: show up rested and on time, and don't get distracted. Pay attention to the real question and keep track of your time.

You'll be glad you did.

CERTIFICATION OBJECTIVE I.02

Define and Understand the Basics of the RDBMS

Now that you've taken a look at the exam's certification objectives, let's get started with reviewing the subject of the Structured Query Language (SQL). This section isn't specifically analyzed on the exam, but the information presented here is foundational to an understanding of everything else in the book. If you're a veteran SQL developer, you might want to just skim the rest of this chapter, looking for the

“Exam Watch” sections, and moving on to the next chapter. Otherwise, stay tuned and get your thinking well positioned to focus on the remainder of the book.

We’ll start with a cursory review of the reason we have SQL: the relational database management system (RDBMS). This section reviews the history and fundamental principles of the RDBMS.

Relational Databases and Dr. E.F. Codd

Before the advent of the RDBMS, software developers found themselves frequently creating applications that used data. These applications needed features to store, change, and retrieve data in various forms. The data was different in every application, but the required functionality was the same—store it, change it, retrieve it. In spite of this common need for functionality, there was no common approach for getting the job done. There was no standard approach to database design, nor a standard set of logic for the storage, changing, and retrieval—each programmer recreated this logic in every application. The result was a slow development effort and proprietary data structures. Programmers found it difficult at best to share each other’s data. Even if they wanted to do it, it was often an arduous, time-consuming effort. Something better was needed.

The concept of a relational database management system was first formally introduced in 1970 in a paper published by an IBM engineer named Dr. E.F. Codd. That paper was titled “A Relational Model of Data for Large Shared Data Banks”, and Codd’s work revolutionized the software industry.

Codd envisioned a system within which programmers could build their own individual databases, using standard methods and functions, with built-in support for common functions to add, modify, and extract data from the database. In an RDBMS, data is stored in *tables*, each of which consists of one or more *columns* of information.

Consider Figure 1-1, which shows a list of ships with the fictional Codd Cruise Lines. The database table has two columns, which are “Ship ID” and “Ship Name”. Each *row* of data includes a bit of information that serves as a unique identifier, which in this example is the “Ship ID” column. For example, the first row has a unique identifier of “1”, the second row has a unique identifier of “2”, and so on. This unique identifier is a *key* to identifying a particular ship’s record. The values found under the “Ship ID” column uniquely identify each ship. This column is considered to be a *primary key* column.

FIGURE 1-1

List of “Codd
Cruise Lines”
ships

Ship ID	Ship Name
1	Codd Crystal
2	Codd Elegance
3	Codd Champion
4	Codd Victorious

Next is Figure 1-2, which is a database table of employees. In this case, we have three columns of data, “Employee ID”, “Name”, and “Ship ID”. The unique identifier here is “Employee ID”.

Now, if I were to ask you to identify the ship to which Mike West was assigned, what would you say?

Naturally you would (or should) say it was the “Codd Victorious”, and you would determine this by looking in the employees table, finding the record for Mike West, then “relating” that record’s Ship ID value to the ship table, and finding that Ship ID 4 “relates” to the ship named “Codd Victorious”.

This is an example of the sort of data that an RDBMS might contain, and the sort of processing it does to “relate” data in one table to another table.

A typical database consists of any number of tables, many of which contain key information that is used to *relate* rows of one table to rows of another table. In the example you just saw, each ship record can theoretically relate to multiple records in the “employee” table. In other words, for every *one* ship, there might be *many* employees. These two tables are said to have a *one-to-many* relationship.

A properly structured relational database system will consist of several tables, each of which contains data that uniquely identifies each record, and then “relates” records to each other, using those unique identifiers, according to the needs of the business rules that the database is intended to support.

FIGURE 1-2

List of “Codd
Cruise Lines”
employees

Employee ID	Name	Ship ID
1	Joe Smith	3
2	Mike West	4
3	Alice Lindon	3

Database Normalization

A full analysis of the concept of database normalization is beyond the task of this book, whose intent is to prepare you for the exam. But it's worth noting the *rules of normalization*, which are a set of rules that drive the design of any set of tables that compose a relational database.

The most common levels of normalization are summarized in Table 1-2.

Normalization is a standard method used by database designers intended to reduce the risk of errors in the database. By eliminating, for example, the existence of unnecessary duplication of data and other design weaknesses, the process of normalization can help minimize the occurrence of conflicting data and improve the quality of the data contained within the database.

A database adheres to the first normal form (1NF) when tables are structured in a one-to-many relationship. For example, in our earlier example of “ships” and “employees”, we would have been in violation of first normal form if we had instead placed all the ship names in the “employees” table and repeated them and any associated data, such as ship length, with each record of each employee that might happen to be assigned to that ship. By separating “ship” and “employee” data, we established the requirement to be, at a minimum, in first normal form.

TABLE 1-2

Levels of
Normalization

Level of Normal Form	Abbreviation	Characterized by
First Normal Form	1NF	No repeating groups, all tables are two-dimensional.
Second Normal Form	2NF	1NF plus each data element is identified by one corresponding unique identifier—a <i>primary key</i> —that is not a composite and therefore cannot be subdivided into smaller bits of data.
Third Normal Form	3NF	2NF plus all tables contain no data other than that which describes the intent of the primary key—extraneous data is placed in separate tables.
Boyce-Codd	BCNF	A slightly modified version of 3NF designed to eliminate structures that might allow some rare logical inconsistencies to appear in the data.
Fourth Normal Form	4NF	BCNF plus additional logic to ensure that every multivalued dependency is dependent on a superkey.
Fifth Normal Form	5NF	4NF plus every join dependency for the table is a result of the candidate keys.

e x a m**W a t c h**

Nothing in the exam objectives, nor in my experience of having taken the exam, indicates that the test involves anything other than typical third normal form scenarios.

Second normal form (2NF) exists when no non-key attribute is dependent upon a part of a composite key.

Third normal form (3NF) is the most commonly used form of normalization for transaction-based database applications. In this form, lookup data is moved to separate tables.

These descriptions are merely a refresher and are not intended to be an exhaustive analysis of database normalization. For that,

I refer you to other books in the Oracle Press line that deal with the fundamentals of database design.

Database Design Considerations

Databases may be used for different purposes, and not all purposes require the same sort of design. Furthermore, there are conflicting priorities that any database designer is forced to consider in the design of any database.

For example, the ideal database shouldn't store any data more than once. This way, errors are minimized considerably. After all, when you enter the current phone number for a customer, it should only be located in one place in the database, so you don't end up with a database that requires the end users to update information in more than one location. What if the users forget? Then you'll end up with conflicting information in the database, and that's a bad situation. With only one location for the "customer's current phone number", any changes made to the customer's current phone number will be done easily, and the results will be clear to future users of the database.

Now—as you've already seen, any record in the database should have a unique identifier. So that would imply that the customer's current phone number should have one unique identifier. That's a reasonable requirement. But what happens if the database design becomes complex, and stores a great deal of information in many tables? And what happens if the complexity of the design requires that unique identifier to relate to many tables in order to finally extract the customer's current phone number, buried down inside a set of tables? Depending on your computer hardware, you might experience slow performance if the database grows in size to tens of millions of records, with many interrelated tables.

And now imagine that your database is required to reply instantly to any incoming phone call, in such a way that the office telephone system identifies the incoming phone number via caller ID and sends that phone number to the database. The customer's name must appear immediately on the screen, so that your call center can answer "Hello, Mr. Codd, nice to hear from you again today, your order is on its way".

That means your database has to reply instantly to that particular query. That might require some special consideration, depending on the circumstances. Perhaps you'll choose to copy the set of "customer current phone number" records in a second table to support speedy lookups if the overall system is too slow and cumbersome to handle it.

There are purists out there who would argue this is bad design. But try to explain that to your boss when the hardware budget is tight and your incoming customer phone calls are being passed off anonymously to the call center in violation of corporate policy.

The bottom line is that the database exists to serve the organization, not the other way around, and you'll often find yourself in a situation where your design choices and trade-offs require you to bend the rules of what is considered good design.



It's important to note that there is not necessarily a single right or wrong way to model every system out there. Some design decisions involve trade-offs of performance (speed of response) versus reduction of duplicate data and complexity of the resulting application. These are the challenges to any data modeler and to any SQL developer.

Now that you have a basic idea of how a database should be structured, how do you actually build it? To accomplish this, you need a tool. This is why the Structured Query Language was created.

CERTIFICATION OBJECTIVE 1.03

Define and Understand the Basics of SQL

The Structured Query Language is often abbreviated with the letters SQL. Some people pronounce it by spelling out the letters, as in "ESS-CUE-ELL".

Others pronounce it as “sequel”. Both pronunciations are fine, and both are used by respected professionals in the industry. Whatever you do, just don’t call it “squeal”.

SQL is a language to

- Create databases and the objects within them.
- Store data in those databases.
- Change and analyze that data.
- Get that data back out in reports, web pages, or virtually any other use imaginable.

Let’s look at a very simple example: consider the ships listed in Figure 1-1. A valid SQL command to create a table in which we could store that information might look like this:

```
CREATE TABLE SHIPS
(SHIP_ID          NUMBER,
 SHIP_NAME        VARCHAR2(20),
 CAPACITY         NUMBER,
 LENGTH           NUMBER );
```

I say “might” look like this because there are a number of options that you might include here, including primary or foreign key declarations, data filtering, storage assignment, and other options that go beyond our simple example. We’ll look at many of those options later in the book. But this code definitely works in an Oracle SQL database.

Next, here’s a SQL command to add our sample record to this table:

```
INSERT INTO SHIPS (SHIP_ID, SHIP_NAME, CAPACITY, LENGTH)
VALUES (1, 'Codd Crystal', 2052, 855);
```

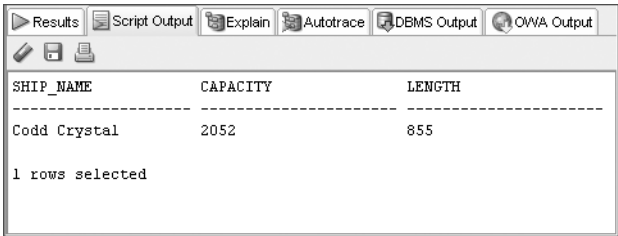
Again, this is a valid command, albeit a simplified version. It inserts one record of information about one ship into our new table SHIPS.

Finally, let’s create a SQL command to display the contents of our newly populated SQL table:

```
SELECT  SHIP_NAME, CAPACITY, LENGTH
FROM    SHIPS;
```

FIGURE 1-3

Output of
our sample
SQL SELECT
statement



SHIP_NAME	CAPACITY	LENGTH
Codd Crystal	2052	855

1 rows selected

If all has gone correctly, we should get a display that appears something like the display shown in Figure 1-3. (Note: Figure 1-3 shows the output as seen in the Oracle tool known as SQL Developer.)

As you can see, our data is stored in the table, and it is still there—SELECT merely displays the data; it doesn’t change the data at all.

At its simplest level, this is what SQL is all about—writing statements to create database objects, and then working with those objects to store and retrieve data.

There are many SQL commands. Some of the more commonly used SQL commands are shown in Table 1-3.

There are more commands than are shown here. For the commands that are shown here, there are many, many clauses, parameters, and other additional features for each one. Later in the book we will look in great detail at each command that is covered by the exam.

TABLE 1-3

Some of the
More Commonly
Used SQL
Commands

SQL Command	Description
SELECT	Retrieves data from a table.
INSERT	Adds new data to a table.
UPDATE	Modifies existing data in a table.
DELETE	Removes existing data from a table.
CREATE object_type	Creates a new database object, such as a table.
ALTER object_type	Modifies the structure of an object, such as a table.
DROP object_type	Removes an existing database object, such as a table.

CERTIFICATION OBJECTIVE 1.04

Understand the Oracle RDBMS and Oracle SQL

So where does Oracle enter the picture? Let's revisit Dr. Codd's story.

After Codd's paper was published in 1970, it eventually created quite a reaction in the software industry and triggered an effort within IBM to create a commercial database product. But as IBM engineers worked on their RDBMS concept and spoke about their efforts at conferences and throughout the industry, they inadvertently inspired a young entrepreneur named Larry Ellison to create an RDBMS product of his own, one that might complement IBM's product and, in the end, actually aligned more closely with Codd's original theories.

That competing product is known today as Oracle, and was officially released first, a few weeks before IBM's product, which today is known as DB2. There are many commercially available RDBMS products on the market. But only one is the dominant and unquestioned leader in the field: Oracle.

Oracle Is the Market Leader

The Oracle relational database management system (RDBMS) is the preeminent data management platform in use throughout the world today. It is the most reliable, comprehensive, robust, scalable, extensible, secure, and dynamic system for managing any amount of information, in any form, on any platform, for any number of end users, in any implementation of business rules available today. Its dominant market share is clear evidence that the marketplace recognizes the established and growing power and significance of Oracle and shows no sign of reversing this trend.

Do we have to elaborate any more on this? Does anyone dare to disagree? I'm very tempted to quote Marvel Comics legend Stan Lee with his signature "nuff said!" And yet . . . there really is so much more to say here.

I think it's fair to say that a lot of people in the world of technology, finance, and American culture see Microsoft as the number one software maker in the world. Part of the reason is that Bill Gates, the founder of Microsoft, has been the richest man in the world for quite some time now. Not too far behind him, though, is Larry Ellison, the legendary founder of the Oracle Corporation, a company that is often considered the second largest software maker in the world.

None of this is probably new to you, but consider this: Microsoft is behind a wide array of products: the operating system, integrated office software suites, a variety of development tools, games for your PC, the Xbox line, even the MSNBC cable news channel and web site, which is their joint venture with NBC.

Oracle Corporation, on the other hand, is built on the foundation of their flagship database product and the complement of tools that support it. That's really about it. In other words, you could argue that Oracle is the world's number two software company on the strength of what is really just one product, and its associated tools.

It's an amazing testament to the significant position in the world today that's held by the Oracle RDBMS.

on the
job

Two of the leading competitors to Oracle's RDBMS are Microsoft SQL Server, and IBM's DB2. Both are respectable products. But Microsoft's database only runs on the Microsoft operating systems. It doesn't run on Linux or Unix or other platforms. DB2, on the other hand, does run on multiple platforms and offers some competition to Oracle, but as evidenced by market share, Oracle is still seen as an overall superior product in terms of interoperability, ease of maintenance, integration with other tools, and more.

Since its inception, Oracle Corporation has grown quite dramatically, with no end in sight. Through corporate acquisitions, many of the leading industry commercial application products, such as PeopleSoft, J.D. Edwards, Hyperion, and other products—that started with other companies—are now part of the Oracle family. The annual Oracle Open World conference, once a gathering of a few thousand technical developers, has exploded into a gargantuan event featuring several tens of thousands of attendees from every industry imaginable.

But each of these acquisitions, and all of Oracle's tools, have one feature in common: their dependence on, use of, or relationship to the core product of the company, which to this day is still the Oracle RDBMS. Oracle Corporation used to be fond of reminding Open World attendees that 99 of the top 100 revenue-generating web sites were all built on the Oracle database: Amazon, eBay, Google, Dell, the list goes on and on—all relying on Oracle databases. (One exception: IBM, whose online order processing system uses DB2.)

on the
job

You may be asking yourself: where is Microsoft Access in relationship to Oracle? The answer is simple: it doesn't have memory management capabilities comparable to Oracle, and is not capable of "scaling up" well, meaning that you cannot easily add large numbers of end users to a given application on an instance of Microsoft Access.

Certification: Oracle SQL Versus ANSI SQL

The certification exam will test you for Oracle SQL. Oracle SQL is close to, but not identical to, the standard established for SQL by the American National Standards Institute, also known as ANSI standard SQL. You will not be required to know the differences between them. There's a lot more I could say here, but it's not relevant to the exam, so let's move on.

Certification: Oracle SQL Versus Oracle SQL*Plus

The certification exam will test you for Oracle SQL, but not for Oracle's enhancements to SQL known as SQL*Plus.

Note, however, that SQL*Plus is a set of commands, and it's also a software tool with an interface into which you can type SQL and SQL*Plus commands and monitor their execution.

You won't be studying SQL*Plus commands in this book, but we will use the SQL*Plus Command Line Interface from time to time to demonstrate Oracle SQL commands.

Oracle's Tools for Working with SQL

Most of Oracle's various products and tools, such as Oracle Financials, Oracle Project, and others, all use SQL. Many development tools, such as Oracle JDeveloper, provide the ability to enter SQL statements and execute them. Two of the most commonly used tools for this purpose are the SQL*Plus Command Line Interface and SQL Developer.

The SQL*Plus Command Line Interface

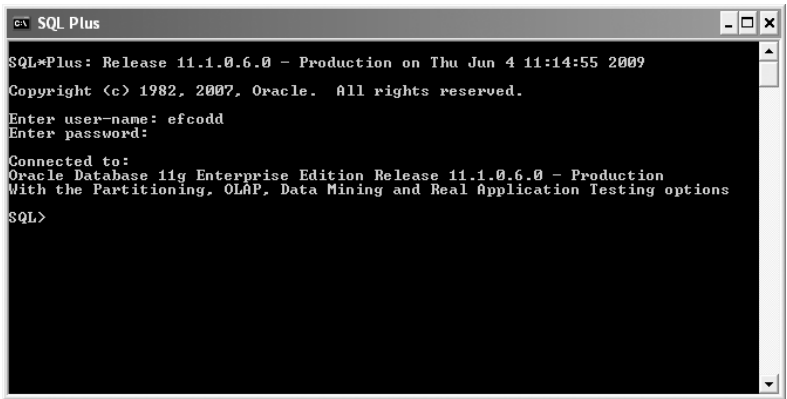
The SQL*Plus command line interface is a simple way to type SQL commands, execute them, and observe the result. It's a universal system that operates the same way in every operating system.

See Figure 1-4 for an example of what the command line interface looks like.

The advantage to the command line interface is that it functions identically in Windows, Unix, and Linux operating systems. That's one of the many advantages that Oracle has always offered—ease of use in any operating system.

FIGURE I-4

The SQL*Plus command line interface

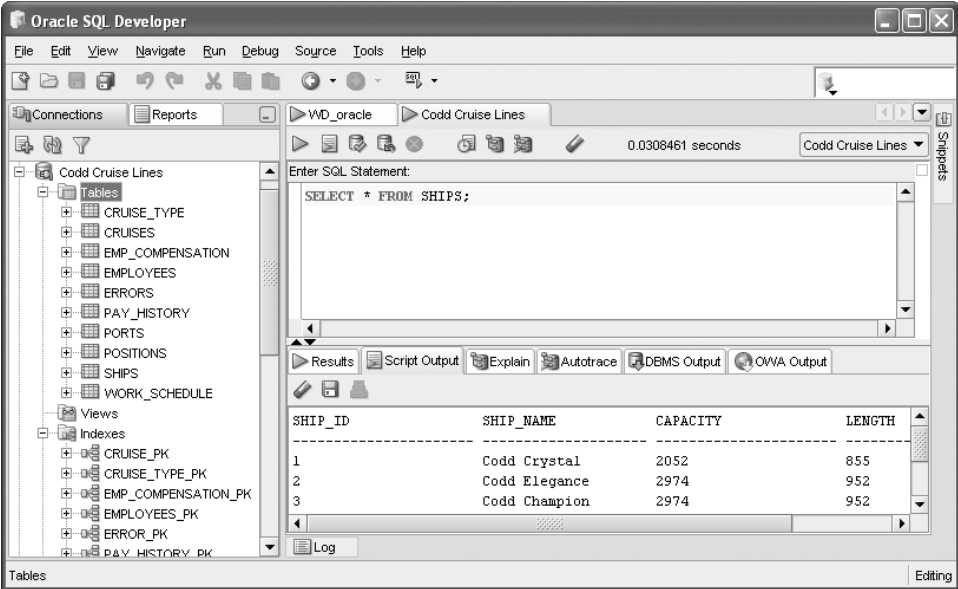


SQL Developer

The SQL Developer interface is a GUI-style interactive point-and-click menu-driven interface that's very powerful and gives the developer a quick overview of the entire database. Some commands may be entered by either typing them in or using point-and-click-style interaction with a graphic menu. See Figure 1-5 for an example of what SQL Developer looks like.

FIGURE I-5

The SQL Developer tool



There are other tools that process SQL statements:

- Oracle JDeveloper
- Oracle Application Express
- SQL Workshop
- . . . and others

For purposes of the certification exam, your choice of interface is irrelevant. SQL statements execute correctly in all Oracle interfaces.

exam

Watch

The exam will test for your knowledge of the syntax of SQL, not your ability to point and click your way through a GUI. In other words, the fact that you might be able to create a SQL table using a code generator through a point-and-click interface will not help you during the exam. Furthermore, if you are a serious applications architect / programmer, there will eventually come a time—probably frequently—when you need to design and/or program such a feature as part of a larger application, and design and embed SQL code into other programming languages that have no access to the nice GUI tools during application run time. Furthermore, as we'll see in numerous

instances in this book, there are many types of SQL statements in which you can combine features and clauses in such a way that they appear to be correct, execute without error, and produce lots of output—all of which can be totally erroneous. A trained eye glancing at the SQL code will recognize the mistake; an untrained eye will not even realize there is a problem. In other words: there is no substitution for comprehensive knowledge of the syntax of SQL, neither in the world of the serious software developer, nor on the certification exam. Know your syntax. As you study for this exam, type your commands and make sure they are done correctly.

Oracle's Documentation for SQL

Oracle Corporation offers a wealth of documentation at its web site, particularly through the Oracle Technology Network, accessible at <http://otn.oracle.com>. The amount of documentation is almost overwhelming, particularly to a newcomer.

But one volume in particular is of interest to us for the purpose of the certification exam, and that volume is the *SQL Language Reference Manual*. It is a huge book, at close to 1,500 pages long. The size of the PDF version is 22MB. Its syntax charts are complex and go far beyond the needs of the exam. The book contains far more information than what you'll need to pass, all of which is yet another reason why you're brilliant to have obtained this book you now have in your hands. I will refer to the *SQL Language Reference Manual* from time to time, but I will only focus on the parts that are relevant to pass the exam. Other useful references of relevance to this book and the exam include Oracle's *Advanced Application Developer's Guide*, *Concepts*, *Security Guide*, *Globalization Support*, and the *Administrator's Guide*. We will refer to the set of manuals for the Oracle database version 11g, Release 1, which is to say version 11.1. The questions for exam 047 have been tested against database versions 10g and 11g.

CERTIFICATION OBJECTIVE 1.05

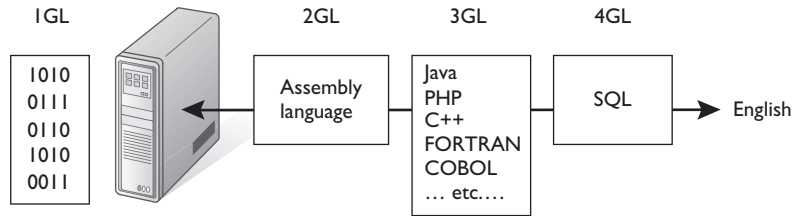
Understand the Unique Role of SQL in Modern Software Systems

By becoming a certified expert in Oracle SQL, you're establishing yourself in a very unique position in the world of software systems. Go back and look at the demand for different skills over the past few decades, and you'll see that the demand for many different computer professionals with other skill sets has spiked and fallen. But SQL has always been in steady demand for as long as it has existed.

There's a reason for that. We'll consider two in the next sections.

SQL Is a 4GL

SQL is unusual in the world of computer languages, in that it is arguably the most successful and widely used fourth-generation language, or 4GL. The term "generations" is used in describing computer languages to help to identify where a particular computer language falls on the spectrum of computer languages (see Figure 1-6) between the ones and zeros that a computer speaks versus the plain English or whatever other human language is spoken by the people who use the database. Take a look at Table 1-4.

FIGURE I-6Language
spectrum**TABLE I-4**Generations
of Computer
Languages

Generation of Computer Languages	Examples	Code Sample
First	Machine code	0011011010
Second	Assembly language	MV R1 R2
Third	FORTRAN, COBOL, C, C++, Java, PHP, Perl, etc.	IF TRUE THEN ... END IF;
Fourth	SQL, assorted artificial intelligence languages	SELECT FIRST_NAME FROM EMPLOYEES;

A first-generation language (or 1GL) is really just one single language: machine language, which is made up of the ones and zeros that computers use to communicate. All computers speak in some form of machine language, all the time. You can write a program in machine language, and if you’ve been around long enough, perhaps you have. But it’s tedious.

Second-generation languages (2GLs) include “assembly language”, which is still very detailed and tedious, albeit an improvement over machine language. For example, when writing assembly code, you don’t just write a command to add two numbers together—instead, you write code that identifies which register in the computer’s central processing unit (CPU) contains the first number, and which register contains the second number, then invoke a command to perform the arithmetic and store the result in a third location. In assembly language, the programmer is obligated to deal with very low levels of detail within the computer’s hardware to perform relatively common coding tasks. Assembly language is a bit easier to code than machine language, but not much—it’s still rather tedious and not much more efficient than machine language.

Third-generation languages (3GLs) are a significant improvement over machine or assembly language, in that their language syntax is much easier to understand for most software developers, relatively speaking. 3GLs include common language features such as variables and constants, “IF THEN ELSE” constructs, loops, error handling, and more. Most of the computer languages in common use today are 3GLs—for example, languages such as Java, C++, and PHP are in this category. The programmer still must code according to the syntax rules of the language, and eventually that code is translated to machine code for the computer to actually execute. But the coding process is far more productive.

The general idea with each generational step away from machine language is to try to get closer to the point where plain spoken human language will be sufficient to program a computer. Whether that goal will ever be fully realized is a subject for another book, but the point here is that the higher the GL number, the closer the language used by the programmer is to human language, and the further away it is from machine language.

So it’s interesting to note that all common software languages in general use today are no higher than the third generation, with one exception: SQL.

SQL is unusual in that it is categorized as a fourth generation language, or 4GL. It is the only widely used 4GL in commercial use today. (Other 4GLs include artificial intelligence languages.) As a 4GL, it’s theoretically closer to human language than any 3GL. This is good, in that it is more powerful and enables SQL developers to do more work with fewer commands. But it’s also tricky, in the sense that this increased power makes it possible to make huge mistakes at a larger degree than a typical 3GL might allow.

In other words, a 4GL carries with it the inherent obligation that its practitioner be well trained, and thoroughly knowledgeable of its power.

SQL: Gateway to the RDBMS for All Other Languages

When an application is said to be compatible with a database like Oracle, then you can be assured of one fact: there’s SQL code in that application. The overall application may be written in something else—like Java or C++. But the code that interacts with the Oracle database will be written in SQL.

SQL is the only language that can talk directly to a SQL RDBMS, such as Oracle, from within any other software system, such as a front-end application, or a web service, or even another database. (I’m making the distinction of a “SQL RDBMS” because there are some RDBMS products out there that are not SQL compliant. But the vast majority are SQL compliant.) While any software developer today

has a wide variety of tools and languages from which to choose when developing an application, there's only one choice when it comes to database interaction, and that's SQL.

"But," you might say, "I have this friend who wrote a database program in Java." And you probably do. But I guarantee you that embedded somewhere in that application is SQL.

"But," you might say, "I read that Oracle has this other language called PL/SQL for writing applications." And they do. But part of the PL/SQL language syntax is SQL code. Thus the name—"Procedural Language / SQL".

So no matter what language you choose to create an application, if that application is going to converse with an RDBMS—and most serious commercial and government applications will require some sort of RDBMS—then regardless of the language of your application, you'll still end up using SQL to interact with the database. For an illustration, see Figure 1-7.

So how many applications require the use of a database? One might argue that virtually all do.

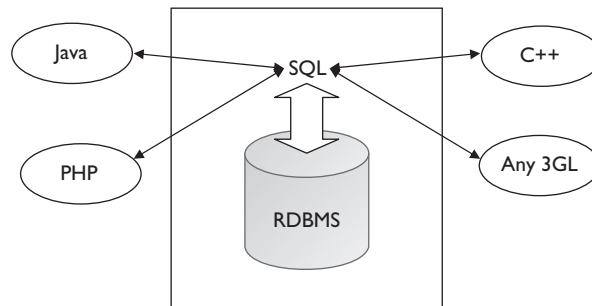
Given that, here's a question that won't be on the test; this is just between us. Here it is: which of the following answers best describes a typical database? A typical database tends to

- A. Shrink in size
- B. Stay the same size
- C. Grow in size

What do you think the answer is? I'm not sure there's any empirical evidence to back me up on this, but I should think common sense would tell us all that databases tend to grow, and often grow dramatically, all the time, *ad infinitum*.

FIGURE 1-7

All 3GLs use SQL to communicate with the RDBMS.



The result: SQL has never gone “out of style” as many other languages have. Demand for other languages ebbs and wanes, but the demand for skilled SQL professionals has been persistently high ever since the early 1980s.

Syntax Isn’t Enough

SQL is a deceptively tricky language. It’s not a language where a rookie programmer can “try stuff until it works”. Sure, that rookie might get some output, but is it the output that the query intended? And if it is correct today, will it be correct tomorrow?

One reason SQL is so tricky is the dynamic nature of the database. It’s entirely possible to test a completed SQL script against a given database, testing against every single available record in the database at a given moment in time, and for that script to pass successfully—and yet for the script to suddenly stop working correctly some time later, with no apparent warning. The reason has to do with the nature of databases. Databases aren’t static. They change and grow with use over time. It’s not enough to test your SQL scripts against existing data or sample data—you must structure your database and write your SQL code with a confident and thorough understanding of all the possible data combinations that might exist now or in the future. This cannot be accomplished with trial and error. You have to have a comprehensive command of all of SQL’s capabilities, and apply that thorough understanding to whatever business task you are facing. Anything short of that runs the risk of creating an erroneous program at best, and an unmitigated disaster at worst.

Several years ago I was brought in to clean up a database that was in use at a particular military installation. The problem: a few months before my arrival, someone very high up in the military had demanded to know how much money was being spent on a particular task. The directive went out to whoever could identify the answer. Three different not-quite-as-high-up people took on the task, and eventually all three sent inquiries to this particular installation to get the answer from the one SQL database that existed on the project. The trouble occurred when each of the three incoming requests ended up on the desks of three different SQL programmers at the installation. Each SQL programmer, unaware of the others, created a SELECT statement that he or she thought would produce the answer. Each SQL programmer created a syntactically correct report, but all were logically incorrect—they each produced a totally different number. One number was three times the amount of another. Not one was correct. When all three numbers arrived at the desk of the requesting authority, it created quite an uproar—this was a

rather visible project politically. Needless to say, the project was embarrassed, and eventually I was brought in to help clean up the system and prevent such problems from occurring in the future.

But the database system was fine. The problem was a lack of proper understanding on the part of those developers about how the database was structured, and—in this particular case—of how SQL's aggregate functions worked when joining multiple tables in the context of a GROUP BY clause.

exam

Watch

One common theme on the exam is to present you with a business case, then a series of SQL statements, and then ask you if the SQL statements will produce the intended result. In other words, the question will not be about syntax. The assumption will be that the code is correct. But does it match the business case? You'll need to recognize the complexities of the SQL language in that situation to determine if the code, which will be syntactically correct, is logically accurate.

So the resulting numbers were all wrong—but hey, the SELECT statement returned an answer, didn't it? So that means it worked, right? Unfortunately, no.

The moral to the story: a successful SQL statement is not one that merely executes without producing an error message. It's one that executes and produces the intended result, both now and—equally important, perhaps dramatically more important—in the future, as new data is added to the system.

There is no way to confidently produce such SQL without having a comprehensive understanding of SQL, the sort of comprehensive understanding for which this certification exam tests. Testing your scripts against sample data won't do the trick by itself.

Databases change over time. New records are added, and existing records might be changed or removed. What works today might not produce the same logical result tomorrow. There is no substitute for you, the SQL professional, developing and maintaining a mastery of SQL.

CERTIFICATION OBJECTIVE 1.06

Confirm Appropriate Materials for Study

This section lists some items you may want to gather as you prepare to study for the exam. If you are a seasoned veteran in the Oracle business, you may not need any of it—this book will suffice. But if you'd like to put forth that extra bit of effort, it

might be a good idea to get your software and documentation together as listed in this section.

Software

Oracle Corporation states that they have validated the 1Z0-047 exam questions against Oracle database versions 10g and 11g. In other words, you can practice with either version; so either way, you'll be prepared for the exam.

(Note: I used version 11g, Release 1, in preparing the SQL statements for this book.)

If you don't have the software you need, you can download it from the official Oracle Corporation web site, at www.oracle.com.



If you haven't joined the Oracle Technology Network, then you should do it right away. There's no charge for it. Visit <http://otn.oracle.com> and sign up today. From there, you can download a great deal of Oracle software for evaluation and study, including the database itself.

If you install the personal version of Oracle, you'll probably get SQL*Plus and SQL Developer, either of which you can use for entering and executing SQL statements. You'll need one or both of those, or if not them, then some sort of tool for entering and executing SQL commands. Chances are you already have something that you're using anyway, or else you probably wouldn't be considering certification.

Documentation

The book you have in your hands is an outstanding reference and is all you need in the way of documentation. This book is the single best guide you could possibly get to prepare you for taking the exam.

But if you crave more documentation, you can download additional documentation from Oracle's OTN website. Remember that the exam has been tested for versions 10g and 11g of the database. I used 11g, Release 1 for this book. There is a set of documentation for each version. Any one version will do—you may as well get the latest version if you choose to download documentation.

The *SQL Language Reference Manual* is Oracle's "bible" of SQL language syntax. It's good to have on hand as a matter of practice, and to reference from time to time as we cover features about which you may have additional questions. As a study

guide for the exam, however, it's overkill: the detail goes far, far beyond the exam requirements and would be extremely difficult to use as a primary study reference. Anyone attempting to use that book alone as a guide for the exam will quickly realize it would be extremely difficult and incredibly time consuming. The manual addresses all aspects of language syntax, whereas the exam does not.

So the best thing you can do is to use this book to guide you through your exam preparation. From time to time in this book, I'll make reference to the *SQL Language Reference Manual*. You won't need it. But some readers may prefer to download a copy to have on hand for further study and exploration of the SQL language.

e x a m

W a t c h

Like many of Oracle's certification exams, the 1Z0-047 exam includes many multiple-choice questions that have more than one correct answer. In other words, a single question will require you to "select all of the following answers that are correct". The result: you must evaluate each individual answer. Once you find a correct answer, you aren't done—you have to continue checking them all. In other words, on these types of questions you can't rule out any of the answers

based on the process of elimination, since all answers are possible candidates for being correct. One question in this format is really more like several questions rolled into one. The result is a more demanding test that requires you to be more knowledgeable, and requires more of your time to answer. So study well. Review this book thoroughly. This is not a simple exam. But you can do it—equipped with this book, your odds of success increase dramatically.

Recently, Oracle began publishing an e-magazine called *Oracle Certification E-Magazine*, and it's available at no charge from the Oracle web site. At the time of this writing, you can find it quickly if you take these steps:

- Visit www.oracle.com.
- Use the upper-right corner pair of "search" boxes.
 - Enter the keywords **ORACLE CERTIFICATION EMAGAZINE**.
 - Choose Education in the pop-up list.
 - Click the magnifying glass to activate your search.

This should list the link to the Oracle Certification Magazine's summary page of current and past issues. From there, you can survey the latest and greatest information on any and all of Oracle's certification programs.

CERTIFICATION SUMMARY

The relational database management system, or RDBMS, is a structure within which database programmers can build database objects to store data. Furthermore, the RDBMS comes with built-in support for many types of objects and features that are typically needed by the programmer to perform common steps, such as adding, modifying, and removing data from the database. The tool used by database programmers to communicate with the RDBMS and its objects is the Structured Query Language, which is commonly called SQL. SQL commands such as `SELECT` and `CREATE object type` are used by SQL programmers to build database objects, store data in them, modify that data, and more.

SQL-based RDBMS products dominate the world of database products in use today. Oracle Corporation's RDBMS has always been, and continues to be, the undisputed industry leader in the field of SQL-based RDBMS products. SQL is the single gateway through which all other languages must go to access any SQL-based RDBMS. Even if an application is written in another computer language, that language will contain embedded SQL statements to interact with any SQL-based RDBMS.

Among all computer languages in use today, SQL is the only widely used fourth-generation language, or 4GL. This means it is a powerful language, but it's also a tricky language to use. You can write commands that are syntactically correct and that execute successfully but are logically incorrect. Furthermore, the database changes over time and SQL code that appears to test correctly now—both syntactically and logically—may fail tomorrow, depending on how the data changes over time. It is imperative that the SQL programmer have a comprehensive command of the SQL language syntax to ensure proper functionality and avoid potential disaster.



TWO-MINUTE DRILL

The Exam: An Overview

- ❑ This chapter provides introductory material that is important to understand in preparing for the exam.
- ❑ The 1Z0-047 Oracle Database SQL Expert exam, which is the subject of this exam guide, has 19 certification objective categories, of which ten are common to another exam, 1Z0-051 SQL Fundamentals I. While 1Z0-047 tests for all 19 of its categories, the exam tends to emphasize the nine areas that are unique to 1Z0-047 and not addressed on 1Z0-051.
- ❑ The exam includes 70 questions and allows 120 minutes to complete them. That's an average of less than two minutes per question.

Define and Understand the Basics of the RDBMS

- ❑ A relational database consists of collections of data known as tables. A table could be a list of ship names and some statistics about each ship. Another table might be a list of employees who work on different ships. The “relational” aspect to a “relational database” has to do with the common information that “relates” two tables together—for example, the list of employees might include an entry for each employee’s ship assignment, which would relate back to the list of ships and each ship’s statistics.
- ❑ A relational database management system, or RDBMS, is a system in which these relational tables and related objects can be created easily, using common functions to add, change, and remove data and database objects from the RDBMS.

Define and Understand the Basics of SQL

- ❑ The Structured Query Language, or SQL, is the language used by programmers to interact with an RDBMS.
- ❑ SQL statements can be used to create, alter, and drop database objects, such as tables.
- ❑ SQL statements can add, change, and remove data from tables and other database objects.

Understand the Oracle RDBMS and Oracle SQL

- ❑ Oracle Corporation released the first commercial RDBMS product. Today, Oracle is the industry leader in the RDBMS market.
- ❑ The American National Standards Institute publishes a set of industry-recognized standards for SQL. Oracle's implementation of SQL largely matches the ANSI standard but isn't 100 percent compliant. Oracle's competition is not fully compliant either.
- ❑ Oracle's SQL*Plus command line interface is a great tool for entering and executing SQL commands from within any operating system platform.
- ❑ Oracle's SQL Developer tool is a great GUI for entering and executing SQL commands from within the Windows operating system.
- ❑ The *SQL Language Reference Manual* is Oracle's nearly 1,500-page manual that describes the Oracle implementation of the SQL language.

Understand the Unique Role of SQL in Modern Software Systems

- ❑ SQL is most widely used fourth-generation language (4GL) in commercial use today.
- ❑ SQL is the only language for interacting with the RDBMS. Any other programming language must use embedded SQL calls to interact with the RDBMS.
- ❑ The constantly changing nature of databases makes them a tricky place to test software. If a SQL script is written and tested successfully today, it's entirely possible that it may break down and produce erroneous information later on. The solution is that the script must not only be tested, but must originally be designed and written by a capable SQL developer who understands proper database design and is thoroughly versed in the RDBMS and SQL syntax.
- ❑ The 1Z0-047 has been validated against Oracle database versions 10g and 11g, so using either to prepare for the exam will be satisfactory.

Confirm Appropriate Materials for Study

- ❑ This book will prepare you to study and successfully take and pass the exam.
- ❑ Oracle's *SQL Language Reference Manual* is overkill as an exam study guide, as it contains far more than you'll need for the exam. But it's a good reference companion to this book.
- ❑ Oracle has recently published an online magazine with the latest news and developments about Oracle's complete line of certification exams covering all of Oracle's products.

SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. While this particular material is not specifically addressed in the exam, an understanding of this material is assumed on the part of anyone studying for and taking the certification exam. Furthermore, these questions are written in the style and format of the certification exam, so it can be good practice to help you get going. As is the case with the exam, some of these self test questions may have more than one correct answer, so read carefully. Choose all the correct answers for each question.

The Exam: An Overview

1. Which of the following topics are not included in the SQL Fundamentals I exam but are addressed on the SQL Expert exam? (Choose all that apply.)
 - A. CUBE
 - B. Hierarchical retrieval
 - C. FLASHBACK
 - D. External tables
2. If you target the specific passing grade requirement of the exam, you can study more efficiently.
 - A. True
 - B. False

Define and Understand the Basics of the RDBMS

3. A database system is “relational” if it does which of the following? (Choose all that apply.)
 - A. Includes “keys” to relate records in one table to records in another table.
 - B. Uses SQL.
 - C. Stores data.
 - D. All databases are “relational”.
4. The most commonly used form of database normalization for transaction-based applications is which of the following?
 - A. 1NF
 - B. 2NF
 - C. 3NF
 - D. None of the above

5. A table consists of (choose the single best answer):
- A. Names and statements
 - B. Rows and columns
 - C. Relations
 - D. Keywords

Define and Understand the Basics of SQL

6. The language used to create objects in an Oracle database is called:
- A. RDBMS
 - B. SQL
 - C. Oracle
 - D. CREATE
7. Which of the following SQL statements is used to remove a database object, such as a table, from the database?
- A. REMOVE
 - B. DROP
 - C. KILL
 - D. DELETE

Understand the Oracle RDBMS and Oracle SQL

8. Choose all of the following statements that are true:
- A. There's only one right way to design any and every database.
 - B. Oracle is the only manufacturer of SQL-compliant databases.
 - C. SQL can be used to add data to a table, but not change it.
 - D. SQL can be used to add data to a table, and also to change that data.
9. Which of the following tools can be used to execute SQL statements against the database? (Choose all that apply.)
- A. SQL Developer
 - B. The *SQL Language Reference Manual*
 - C. The SQL*Plus command line interface
 - D. None of the above

- 10.** What can be said of the SQL*Plus command line interface? (Choose all that apply.)
- A. It is an ANSI-standard tool for executing SQL commands in the database.
 - B. It can be used to format report output in ways that pure SQL cannot do.
 - C. It only runs in Windows.
 - D. It was created by Microsoft to try to steal market share from Oracle Corporation.
- 11.** Which of the following statements are true? (Choose all that apply.)
- A. Almost all medium- to large-size government agencies, companies, and nonprofits depend on data to stay in business.
 - B. Oracle was the first commercial RDBMS product on the market.
 - C. Oracle is the leading commercial RDBMS on the market today.
 - D. Most revenue-generating web sites depend on the Oracle RDBMS for storing data.

Understand the Unique Role of SQL in Modern Software Systems

- 12.** Which of the following statements about SQL are true? (Choose all that apply.)
- A. A given SQL statement will always produce the same answer, no matter when the statement is executed.
 - B. It is not the only choice for communicating with a SQL RDBMS, but it is the best choice.
 - C. SQL is a fourth-generation language (4GL).
 - D. All commercial implementations of SQL are 100 percent ANSI standard.
- 13.** You can be sure that a SQL SELECT statement is performing as intended if it executes without producing an error message.
- A. True
 - B. False
- 14.** You can be sure that a SQL SELECT statement is performing as intended if it executes and returns data from the database.
- A. True
 - B. False
- 15.** You can be sure that a SQL statement is performing as intended if it executes and doesn't change any data.
- A. True
 - B. False

Confirm Appropriate Materials for Study

- 16.** The 1Z0-047 exam (which is the subject of this book) has been officially validated by Oracle Corporation against which of the following versions of the Oracle database? (Choose all that apply.)
- A. Every version
 - B. 9i
 - C. 10g
 - D. 11g
- 17.** The best exam guide you could possibly get for preparing to take and pass the “1Z0-047 SQL Expert” certification exam is which of the following? (Choose all that apply.)
- A. This book
 - B. The book you are holding right now
 - C. This here book
 - D. Don’t make me tell you again

SELF TEST ANSWERS

The Exam: An Overview

1. ☒ **A, B, C, and D.** See Table 1-1 for a full listing of all the topics included in either or both of the exams.
2. ☒ **B** is correct. Although, granted, it's a subjective issue, but Oracle Corporation specifically warns against this. The reason: the published minimum requirement for a passing score can be changed without notice.

Define and Understand the Basics of the RDBMS

3. ☒ **A and B.** SQL can only be used with an RDBMS. And the “keys” in an RDBMS are the feature that makes a relational database “relational”.
☒ **C and D** are incorrect. It's not true that all databases are “relational”; some are simple flat file databases, and other forms exist as well.
4. ☒ **C.** Third normal form is not the only form that's used, but it's the most common for transactional applications.
☒ **A, B, and D** are incorrect. First and second normal forms, while not totally unheard of in professional applications, are generally considered poor design and introduce potential problems in many transaction-based database application scenarios.
5. ☒ **B.** Tables consist of rows of data stored in columns. Data is added to a table in rows. Note: technically, if the table happens to be empty, one might argue that there's no data in it. Nevertheless, Oracle documentation often speaks of tables consisting of rows and columns, so we do as well.
☒ **A, C, and D** are incorrect. Tables have names, it's true, as do a table's columns, but “names” do not singularly form a table's structure. Statements are used to work with tables, not reside in them. Theoretically you could type a statement as text and store it in table, but then the table would see the result as data, and not a statement per se. Tables may optionally relate to other tables, but it's not required. Keywords represent many features throughout the database and in any programming language; they have no unique relationship to a table's structure.

Define and Understand the Basics of SQL

6. ☒ **B.** SQL is the only language that interacts with the database.
☒ **A, C, and D** are incorrect. The RDBMS is not the language, but the type of system in which the language operates. Oracle is the brand name of the particular RDBMS product from

Oracle Corporation. CREATE is one example of a SQL statement, but not a reference to the language itself.

7. ☒ **B.** The correct SQL statement is DROP, as in DROP TABLE SHIPS.
☒ **A, C, and D** are incorrect. REMOVE is not a SQL statement. DELETE is used to remove records from an existing table, but it leaves the table's structure intact, as well as any data not referenced by the DELETE statement. There is no KILL statement in SQL.

Understand the Oracle RDBMS and Oracle SQL

8. ☒ **D.** SQL can be used to create tables, to add data to tables, to modify that data, to remove that data, to remove the table from the database altogether, and more.
☒ **A, B, and C** are incorrect. There can be many good ways to design a database; there might be many reasons why a database design would change. Oracle is not the only maker of SQL-compliant databases—other vendors include IBM and Microsoft. And SQL can be used to change data.
9. ☒ **A and C.** The SQL Developer tool is a Windows-based interface, and the SQL*Plus command line interface works in any operating system.
☒ **B and D** are incorrect. The *SQL Language Reference Manual* is very helpful, but it is just a book, nothing more.
10. ☒ **B.** It provides enhancements to ANSI-standard SQL to support features not provided by the ANSI standard. Among those enhancements is formatting for report output.
☒ **A, C, and D** are incorrect. SQL*Plus is not ANSI standard. It runs in any operating system. And it's an Oracle product.
11. ☒ **A, B, C, and D.** These are all correct.

Understand the Unique Role of SQL in Modern Software Systems

12. ☒ **C.** SQL is a 4GL.
☒ **A, B, and D** are incorrect. SQL statements will not necessarily produce the same answer every time. One reason is that the database is dynamic, and as data is changed in the database, the SQL statement's results will probably change as well. And SQL is the only choice for communicating with a SQL RDBMS. And it's not true that commercial SQL implementations are all completely ANSI compliant—many are not, including Oracle.
13. ☒ **B.** The lack of an error message merely means that the statement's syntax is correct. But the logic may be incorrect. An error message does not result from inaccurate logic in the statement.

44 Chapter 1: Introduction to SQL

- 14. ☒ **B.** The fact that a query returns data does not mean that the query is performing as intended. The data returned may be different from the desired response.
- 15. ☒ **B.** Some SQL statements, such as the UPDATE statement, are intended to change data.

Confirm Appropriate Materials for Study

- 16. ☒ **C and D.** The test has been officially validated against these two versions of the database.
☒ **A and B** are incorrect. The exam tests for functionality that did not exist in earlier versions of the Oracle database.
- 17. ☒ **A, B, C, and D.** Duh.