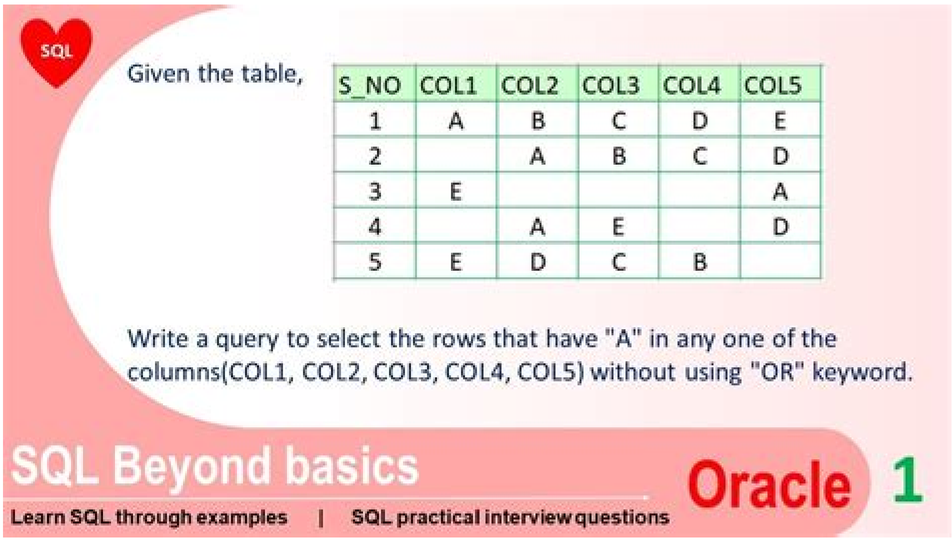
Sql Practice Exercises With Solutions



SQL

Given the table,

S_NO	COL1	COL2	COL3	COL4	COL5
1	A	B	C	D	E
2		A	B	C	D
3	E				A
4		A	E		D
5	E	D	C	B	

Write a query to select the rows that have "A" in any one of the columns(COL1, COL2, COL3, COL4, COL5) without using "OR" keyword.

SQL Beyond basics

Learn SQL through examples | SQL practical interview questions

Oracle 1

SQL practice exercises with solutions are essential for anyone looking to improve their database management skills. Structured Query Language (SQL) is a powerful tool used for managing and manipulating relational databases. Whether you are a beginner wanting to learn the basics or an experienced developer looking to sharpen your skills, practicing with real-world scenarios can significantly enhance your understanding. In this article, we will cover various SQL exercises ranging from simple queries to more complex operations, along with their solutions to help you learn effectively.

Understanding SQL Basics

Before diving into practice exercises, it’s crucial to have a solid understanding of SQL basics. Here are some fundamental concepts:

- Databases: A structured set of data held in a computer.
- Tables: Data is organized into tables, consisting of rows and columns.
- Queries: Instructions used to interact with the database (e.g., selecting data, inserting new records).
- Primary Key: A unique identifier for a record in a table.
- Foreign Key: A field that links to the primary key of another table.

Setting Up a Sample Database

To start practicing, we'll use a sample database that includes two tables: `employees` and `departments`. Here’s how the tables are structured:

Table: employees

- employee_id (INT, Primary Key)
- first_name (VARCHAR)
- last_name (VARCHAR)
- department_id (INT, Foreign Key)
- salary (DECIMAL)

Table: departments

- department_id (INT, Primary Key)
- department_name (VARCHAR)

You can create these tables using the following SQL commands:

```sql
CREATE TABLE departments (
department_id INT PRIMARY KEY,
department_name VARCHAR(100)
);

CREATE TABLE employees (
employee_id INT PRIMARY KEY,
first_name VARCHAR(100),
last_name VARCHAR(100),
department_id INT,
salary DECIMAL(10, 2),
FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

SQL Practice Exercises

Now that we have our sample database set up, let’s dive into some practice exercises.

Exercise 1: Inserting Data

Task: Insert the following records into the `departments` table:

1. Sales
2. Marketing
3. IT
4. HR

Solution:

```sql
INSERT INTO departments (department_id, department_name) VALUES
(1, 'Sales'),
(2, 'Marketing'),
(3, 'IT'),
(4, 'HR');
```

Task: Insert the following records into the `employees` table:

1. John Doe, Sales, 60000
2. Jane Smith, Marketing, 50000
3. Alice Johnson, IT, 70000

4. Bob Brown, HR, 55000

Solution:

```sql
INSERT INTO employees (employee_id, first_name, last_name, department_id, salary) VALUES
(1, 'John', 'Doe', 1, 60000),
(2, 'Jane', 'Smith', 2, 50000),
(3, 'Alice', 'Johnson', 3, 70000),
(4, 'Bob', 'Brown', 4, 55000);
```

Exercise 2: Selecting Data

Task: Retrieve all records from the `employees` table.

Solution:

```sql
SELECT FROM employees;
```

Task: Retrieve the first names and last names of employees who work in the IT department.

Solution:

```sql
SELECT first_name, last_name FROM employees
WHERE department_id = 3;
```

Exercise 3: Updating Data

Task: Update the salary of John Doe to 65000.

Solution:

```sql
UPDATE employees
SET salary = 65000
WHERE first_name = 'John' AND last_name = 'Doe';
```

Exercise 4: Deleting Data

Task: Delete the record of Bob Brown from the employees table.

Solution:

```sql
DELETE FROM employees
WHERE first_name = 'Bob' AND last_name = 'Brown';
```

Exercise 5: Aggregate Functions

Task: Find the average salary of employees.

Solution:

```sql
SELECT AVG(salary) AS average_salary FROM employees;
```

Task: Count the number of employees in each department.

Solution:

```sql
SELECT department_id, COUNT() AS employee_count
FROM employees
GROUP BY department_id;
```

Exercise 6: Joining Tables

Task: Retrieve a list of employees along with their department names.

Solution:

```sql
SELECT e.first_name, e.last_name, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id;
```

Advanced SQL Exercises

Once you feel comfortable with the basics, you can try some advanced exercises.

Exercise 7: Subqueries

Task: Find employees whose salary is above the average salary of all employees.

Solution:

```sql
SELECT first_name, last_name, salary

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);
```

Exercise 8: Using CASE Statements

Task: Create a query that categorizes employees based on their salary.

Solution:

```sql
SELECT first_name, last_name,

CASE

WHEN salary < 50000 THEN 'Low'

WHEN salary BETWEEN 50000 AND 70000 THEN 'Medium'

ELSE 'High'

END AS salary_category

FROM employees;
```

Exercise 9: Ordering Results

Task: Retrieve a list of employees ordered by their salary in descending order.

Solution:

```sql
SELECT first_name, last_name, salary
FROM employees
ORDER BY salary DESC;
```

Exercise 10: Limiting Results

Task: Retrieve the top 2 highest-paid employees.

Solution:

```sql
SELECT first_name, last_name, salary
FROM employees
ORDER BY salary DESC
LIMIT 2;
```

Conclusion

Practicing SQL through exercises with solutions is an effective way to solidify your understanding of database management. From simple data manipulation to advanced queries, each exercise helps you grasp the intricacies of SQL. As you continue to practice, you will find that your ability to write complex queries and manage databases will improve significantly. Make sure to explore different scenarios and challenges to further enhance your SQL skills. Happy querying!

Frequently Asked Questions

What are some effective SQL practice exercises for beginners?

Effective SQL practice exercises for beginners include tasks like retrieving data from a single table, filtering results using WHERE clauses, sorting data using ORDER BY, and using aggregate functions like COUNT, SUM, AVG.

How can I practice SQL joins through exercises?

You can practice SQL joins by creating exercises that require combining data from multiple tables. Examples include inner joins to find matching records, left joins to include all records from one table, and full outer joins to retrieve all records from both tables.

Are there any online platforms that offer SQL exercises with solutions?

Yes, platforms like LeetCode, HackerRank, SQLZoo, and Codecademy offer a variety of SQL exercises along with solutions and explanations to help you understand the concepts better.

What are some common SQL queries that I should practice?

Common SQL queries to practice include SELECT statements, INSERT, UPDATE, DELETE commands, as well as using GROUP BY, HAVING, and subqueries for advanced data manipulation.

How can I build a custom SQL exercise set for practice?

You can build a custom SQL exercise set by defining a small database schema, populating it with sample data, and then creating a series of questions that require different SQL operations to retrieve or manipulate that data.

What types of SQL exercises can help improve performance tuning skills?

Exercises that focus on optimizing queries, analyzing execution plans, and rewriting inefficient SQL statements can significantly improve your performance tuning skills.

Can I find SQL exercises specific to certain database management systems?

Yes, many resources offer SQL exercises tailored to specific database management systems such as MySQL, PostgreSQL, SQL Server, and Oracle, focusing on their unique features and functions.

What is the benefit of solving SQL exercises with provided solutions?

Solving SQL exercises with provided solutions allows you to verify your answers, understand different approaches to the same problem, and learn best practices in SQL query writing.

How can I track my progress while practicing SQL exercises?

You can track your progress by maintaining a log of exercises completed, noting the concepts learned, and revisiting challenging queries to measure improvement over time.

Find other PDF article:
https://soc.up.edu.ph/19-theme/Book?docid=EHY67-0904&title=edgenuity-biology-unit-test-answers.pdf

Sql Practice Exercises With Solutions

□□□□*SQL*□ - □□
SQL□□□□□□□□□□□□□□□□□□□□□□□□□□ □□SQL□□□□□□□□□□□□□□□□□ SQL□□□□□□□□□□□□sql□□□□□□□□□□□□□□□□□□

What does <> (angle brackets) mean in MS-SQL Server?

Nov 8, 2013 · What does <> (angle brackets) mean in MS-SQL Server? Asked 11 years, 8 months ago Modified 3 years, 11 months ago Viewed 80k times

sql - Not equal <> != operator on NULL - Stack Overflow

Apr 14, 2011 · 11 In SQL, anything you evaluate / compute with NULL results into UNKNOWN This is why SELECT * FROM MyTable WHERE MyColumn != NULL or SELECT * FROM ...

□□□□ SQL □□□ - □□

SQL□□□ 6000□□□□□□□□□□□□ SQL □□□ □□□ □□ □□□□ SQL □□□□ □□□□□□□□□□SQL□~□□□□□~ PYTHON□□□ □□□□□□□□□Python□□□ ...

What does the "@" symbol do in SQL? - Stack Overflow

The @CustID means it's a parameter that you will supply a value for later in your code. This is the best way of protecting against SQL injection. Create your query using parameters, rather than ...

What does SQL Select symbol || mean? - Stack Overflow

Apr 29, 2014 · sql server: + (infix operator), concat (vararg function) Edit : Now Azure SQL also supports ANSI SQL standard || operator for string concatenation. Docs link.

sql□□□□□□□□□□□□□□□□ - □□

SQL□□ SQL□□□□□□□□□□ S Q L □□□□□□□□□□□ Structured Query ...

SQL: IF clause within WHERE clause - Stack Overflow

Sep 18, 2008 · Is it possible to use an IF clause within a WHERE clause in MS SQL? Example: WHERE IF IsNumeric(@OrderNumber) = 1 OrderNumber = @OrderNumber ELSE ...

Should I use != or <> for not equal in T-SQL? - Stack Overflow

Apr 6, 2009 · Yes; Microsoft themselves recommend using <> over != specifically for ANSI compliance, e.g. in Microsoft Press training kit for 70-461 exam, "Querying Microsoft SQL ...

What does the colon sign ":" do in a SQL query?

May 9, 2017 · What does ":" stand for in a query? A bind variable. Bind variables allow a single SQL statement (whether a query or DML) to be re-used many times, which helps security (by ...

□□□□SQL□ - □□

SQL□□□□□□□□□□□□□□□□□□□□□□□□□ □□SQL□□□□□□□□□□□□□□□□□ SQL□□□□□□□□□□□□sql□□□□□□□□□□□□□□ □□□□

What does <> (angle brackets) mean in MS-SQL Server?

Nov 8, 2013 · What does <> (angle brackets) mean in MS-SQL Server? Asked 11 years, 8 months ago Modified 3 years, 11 months ago Viewed 80k times

sql - Not equal <> != operator on NULL - Stack Overflow

Apr 14, 2011 · 11 In SQL, anything you evaluate / compute with NULL results into UNKNOWN This is why SELECT * FROM MyTable WHERE MyColumn != NULL or SELECT * FROM ...

□□□□ SQL □□□ - □□

SQL□□□ 6000□□□□□□□□□□□□ SQL □□□ □□□ □□ □□□□ SQL □□□□ □□□□□□□□□□SQL□~□□□□□~ PYTHON□□□ □□□□□□□□□Python□□□ ...

What does the "@" symbol do in SQL? - Stack Overflow
The @CustID means it's a parameter that you will supply a value for later in your code. This is the best way of protecting against SQL injection. Create your query using parameters, rather than …

What does SQL Select symbol || mean? - Stack Overflow
Apr 29, 2014 · sql server: + (infix operator), concat (vararg function) Edit : Now Azure SQL also supports ANSI SQL standard || operator for string concatenation. Docs link.

sql□□□□□□□□□□□□□□□□ - □□
SQL□□□ SQL□□□□□□□□□□ S Q L □□□□□□□□□□□ Structured Query …

SQL: IF clause within WHERE clause - Stack Overflow
Sep 18, 2008 · Is it possible to use an IF clause within a WHERE clause in MS SQL? Example: WHERE IF IsNumeric(@OrderNumber) = 1 OrderNumber = @OrderNumber ELSE …

Should I use != or <> for not equal in T-SQL? - Stack Overflow
Apr 6, 2009 · Yes; Microsoft themselves recommend using <> over != specifically for ANSI compliance, e.g. in Microsoft Press training kit for 70-461 exam, "Querying Microsoft SQL …

What does the colon sign ":" do in a SQL query?
May 9, 2017 · What does ":" stand for in a query? A bind variable. Bind variables allow a single SQL statement (whether a query or DML) to be re-used many times, which helps security (by …

Boost your SQL skills with our comprehensive SQL practice exercises with solutions. Discover how to enhance your coding expertise—start practicing today!

[Back to Home]