# 01-01 SQL introduction

**Question 1:** SQL (Structured Query Language) is a programming language used for managing and manipulating relational databases. It is used to insert, update, and retrieve data from a database. SQL is the standard language for interacting with relational databases and is used by a wide variety of software applications and technologies.

Some common tasks that can be performed using standard SQL include all but the following:

1. Creating, altering, and deleting tables and other database objects
2. Inserting, updating, and deleting data in a table
3. Retrieving data from one or more tables, and performing operations on the retrieved data (such as sorting, filtering, and joining data from multiple tables)
4. Analytical processes like logistic regression, K-means, and K-nearest neighbor models.
5. All of the above

**Answer**: D – standard SQL does not contain modeling options.

**Question 2**: True or False: SQL commands can be executed through a variety of interfaces, including command-line interfaces, graphical user interfaces, and programming languages such as Python and Java.

**Answer**: True

Some of the most basic SQL statements are SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY. In the following questions, match up the SQL statement with its. For example, the WHERE CLAUSE is used to filter the rows based on certain conditions.

1. SELECT
2. FROM
3. GROUP BY
4. HAVING
5. ORDER BY

**Question 3**: The \_\_\_\_\_\_\_\_ clause is used to filter the groups based on certain conditions.

**Answer**: D – HAVING.

**Question 4**: The \_\_\_\_\_\_\_\_\_ clause is used to specify the columns that you want to retrieve. You can list one or more columns separated by commas.

Answer A – SELECT.

**Question 5**: The \_\_\_\_\_\_\_\_\_ clause is used to sort the rows based on one or more columns.

**Answer**: E – ORDER BY.

**Question 6**: True or False: The SELECT and WHERE clauses should always be present in a select statement, FROM, GROUP BY, HAVING, and ORDER BY is optional depends on how you want to output the data.

**Answer**: False – The SELECT and FROM clauses should always be present in a select statement, where, group by, having, and order by is optional depends on how you want to output the data.

# 01-02 Predicates, Operators, and Case Expressions

Predicates are conditions that evaluate to either true or false, and they are used in SQL statements to filter, sort, or group data. Predicates are typically used in the WHERE clause of a SELECT, UPDATE, or DELETE statement to specify which rows to include or exclude based on some condition.

**Question 7**: In SQL, predicates include:

1. Comparison operators
2. Logical operators
3. LIKE operator
4. BETWEEN operator
5. All of the above

**Answer**: E – All of the above

**Question 8**: True or False: Comparison operators are used to compare two values and return a boolean result. This include Equal to, not equal to, greater than, less than, greater than or equal to, and less than or equal to.

**Answer**: True.

# 01-03 String Functions

**Question** **9**: In SQL, the process of joining two or more strings or character values into a single string value is referred to as \_\_\_\_\_\_\_\_. It is frequently used in SQL to build dynamic strings, such as generating custom messages or constructing complex SQL queries that depend on variable inputs.

1. Arithmetic operator
2. Right Function
3. Concatenation
4. Substring Function
5. Length Function

**Answer**: C – Concatenation

**Question 10**: True or False – The substring function in PostgreSQL is used to separate a substring from a provided string. The function will return the complete string if the length parameter is more than the length of the string while the function will return an empty string if the length parameter is negative or zero.

**Answer**: True

# 01-04 String Functions Part 2

**Question 11**: True or False – The PostgreSQL replace() function performs a case-sensitive match when searching for the substring replaced.

**Answer**: False – The PostgreSQL replace() function performs a case-insensitive match when searching for the substring replaced.

**Question 12**: Which string function creates a sequence of a string in repetition?

1. Replace
2. Upper
3. Repeat
4. Lower
5. Initcap

**Answer**: C

# 01-05 String Functions Part 3

**Question 13**: True or False: You can only remove characters on the right side of a string by using RTRIM.

**Answer**: False. TRIM can also be used with the keyword TRAILING to remove characters from the right.

**Question 14:** Which string functions can be used to remove characters on both ends of the string?

1. LTRIM
2. TRIM
3. RTRIM
4. REPEAT

**Answer**: B. TRIM

# 01-06 Working with Date & Time

**Question 15:** When storing a date value, PostgreSQL uses the \_\_\_\_\_\_\_\_ format, which is also the same format used for inserting data.

1. yyyy-mm-dd
2. yyyy-dd-mm
3. mm-dd-yyyy
4. dd-mm-yyyy

**Answer:** A – yyyy-mm-dd format

**Question 16**: The PostgreSQL TO\_CHAR function, the syntax TO\_CHAR(expression, format), converts a date into a string and can utilize the following into a string:

1. A timestamp
2. An interval
3. An integer
4. A numeric value
5. All of the above

**Answer**: E – all of the above

**Question 17**: Which date & time function can generate the complete date with time zone?

1. Age
2. Current\_Date
3. Now
4. Both B & C
5. None of the above

**Answer**: C – Now.

# 01-07 Working with Date & Time P2

**Question 18**: The DATE PART function\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

1. Extracts a subfield from a date or time value.
2. Allows the user to round off the date field to any timestamp instance.
3. Converts a date into a string and can utilize the following into a string.
4. Performs a case-insensitive match when searching for the substring replaced.

**Answer**: A

**Question 19**: True or False: The date\_part works differently with that of the extract function.

**Answer**: False. They work the same, just that extract complies with SQL standard & date\_part only works in postgres.

**Question** **20**: The date trunc function, DATE\_TRUNC, of Postgres \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. So instead of simply cutting off a part of the value, it rounds the date to the value based on the datetime identifier specified.

1. Extracts a subfield from a date or time value.
2. Performs a case-insensitive match when searching for the substring replaced.
3. Allows the user to round off the date field to any timestamp instance.
4. Converts a date into a string and can utilize the following into a string.

**Answer**: C

# 01-08 Converting Data Types

**Question** **21**: The TO\_DATE function does what?

1. Converts a character string to a numeric value.
2. Converts a string data to a date & timestamp with timezone according to the specified format.
3. Converts string literal to a date value.
4. Converts any datatype to text.

**Answer**: C – Converts string literal to a date value.

**Question 22**: The PostgreSQL TO\_TIMESTAMP function does what?

1. Converts a character string to a numeric value.
2. Converts a string data to a date & timestamp with time zone according to the specified format.
3. Converts string literal to a date value.
4. Converts any datatype to text.

**Answer**: B – Converts a string data to a date & timestamp with timezone according to the specified format.

**Question** **23**: What does the To\_number function do:

1. Converts a character string to a numeric value.
2. Converts a string data to a date & timestamp with timezone according to the specified format.
3. Converts string literal to a date value.
4. Converts any datatype to text

**Answer**: A

**Question 24**: True or False: The second way of converting data types is by using the Cast operator which can convert a value of one datatype into another.

**Answer**: True.

# 01-09 Joins P1

**Question 25**: PostgreSQL supports the basic type of joins including:

1. Inner Join
2. Left Outer Join
3. Right Outer Join
4. Cross Join
5. All of the above joins.

**Answer**: E – All of the above joins.

**Question 26**: True or False: PostgreSQL join is used to combine columns from 1 or more tables based only on the values of the foreign key of both tables.

**Answer**: False. PostgreSQL join is used to combine columns from 1 or more tables based on the values of the common columns between related tables. The common columns are typically the primary key columns of the first table and foreign key columns of the second table.

**Question 27**: An INNER JOIN is the most common type of join and an OUTER JOIN is an extension of the INNER JOIN. SQL standard defines three types of OUTER JOINs, which are all but the following:

1. LEFT
2. RIGHT
3. FULL
4. TRUNC

**Answer**: D – TRUNC is a function.

**Question 28**: True or False: The last type of the basic joins is the FULL OUTER JOIN, which returns a result set that contains all rows from both the left and right tables.

**Answer**: True.

# 01-10 Joins P2

**Question 29**: A natural join combines two or more tables based on their common columns. A natural join can be all but the following:

1. Inner join
2. Left join
3. Right join
4. Cross
5. All of the above.

**Answer**: D - Cross

**Question 30**: True or False – If you do not specify a join explicitly e.g., INNER JOIN, LEFT JOIN, RIGHT JOIN, PostgreSQL will apply the LEFT JOIN by default.

**Answer**: False - INNER

# 01-11 - 01-12 Subqueries

Subquery is basically a query within a query or we can say nested queries. The outermost query which is the main query is called the outer query. The inner query which result set is used by the outer query is the subquery.

**Question 31**: Subqueries can be used for a variety of purposes, including:

1. Filtering data based on a condition in another table.
2. Retrieving a specific set of data based on a calculation or aggregate function.
3. Testing a condition for each row in a table.
4. None of the above.
5. All of the above.

**Answer**: E – all of the above

**Question 32**: Subquery can be either self-contained subquery or correlated subquery. Self-Contained Subquery are queries that are independent to the outer query and there are two types of them, scalar subquery and multivalued subquery.

**Answer**: True

# 01-13 Set Operators & Views

**Question 33**: Aside from using joins, we can also combine data in SQL using Set Operators. Set Operators make it easy to construct queries and filter the results of searches across multiple tables from your database. They are used to join the results of two or more SELECT statements. PostgreSQL has four types of set operators, which of the following is not a standard set operator:

1. UNION
2. UNION ALL
3. LEFT
4. INTERSECT
5. EXCEPT

**Answer**: C – LEFT IS A JOIN

**Question 34**: True or False – When using set operators, all select statements must have the same number of rows.

**Answer**: False – When using set operators, all select statements must have the same number of columns.

**Question 35**: True or False – The UNION OPERATOR is used to combine results from many SELECT statements into a result set. If the result of the SELECT statements have any duplicate rows, then those rows will not be displayed in the result set.

**Answer**: True

**Question 36**: True or False –UNION has better performance compared to the UNION ALL operator because resources are not used in filtering duplicates and sorting the results.

**Answer**: False - UNION ALL has better performance compared to the UNION operator because resources are not used in filtering duplicates and sorting the results.

# 01-14 Window Functions

**Question 37**: A window function performs a calculation across a set of table rows that are related to the current row. This is comparable to the type of calculation that can be done with an aggregate function. However, window functions do not cause rows to become grouped into a single output row like non-window aggregate calls would. Instead, the rows retain their separate identities. The advantages of using a Window Functions include all of the following but:

1. Since the rows are not aggregated into one, you can work with aggregate and non-aggregate values.
2. Simple to use and read, unlike subqueries and joins.
3. As it reduces the complexity of the script, it is a lot easier to maintain.
4. The simplicity helps with performance issues.
5. All of the above

**Answer**: E – all of the above.

**Question 38**: Window functions are permitted only in the \_\_\_\_\_\_\_\_\_\_ clause of the query because they logically execute after the processing of other clauses.

1. SELECT LIST and ORDER BY
2. GROUP BY & ORDER BY
3. HAVING & GROUP BY
4. WHERE

**Answer**: A – SELECT LIST and ORDER BY

**Question 39**: True or False – Window functions execute before non-window aggregate functions, which means it is valid to include an aggregate function call in the arguments of a window function, but not vice versa.

**Answer**: False – Window functions execute after non-window aggregate functions.

**Question 40**: True or False – The DENSE\_RANK assigns a rank to every row in each partition of a result set. This function always returns consecutive rank values.

**Answer**: True.

# 01-16 PL/pgSQL & Variables

**Question 41**: The pgsql code block is composed of the following except:

1. DECLARATION
2. BEGIN
3. SELECT
4. EXCEPTION

**Answer**: C. Select is a basic statement clause.

**Question 42**: True or False – To create an anonymous code you need to add a double dollar sign on the do statement.

**Answer**: True.

# 01-17 PL/pgSQL & Variables

**Question 43**: True or False – Loop control structures let you execute alternative commands based on certain conditions.

**Answer**: False. **Conditional control structures** let you execute alternative commands based on certain conditions.

**Question 44**: The If statement has 3 forms except:

1. IF THEN
2. IF THEN ELSIF
3. IF THEN ELSE
4. IF THEN ELSE ELSIF

**Answer**: D. IF THEN ELSE ELSIF

**Question 45**: Which loop is used to do the job repeatedly within the block of statements until the condition mentioned becomes false?

1. While
2. Basic
3. For

**Answer**: A. While loop