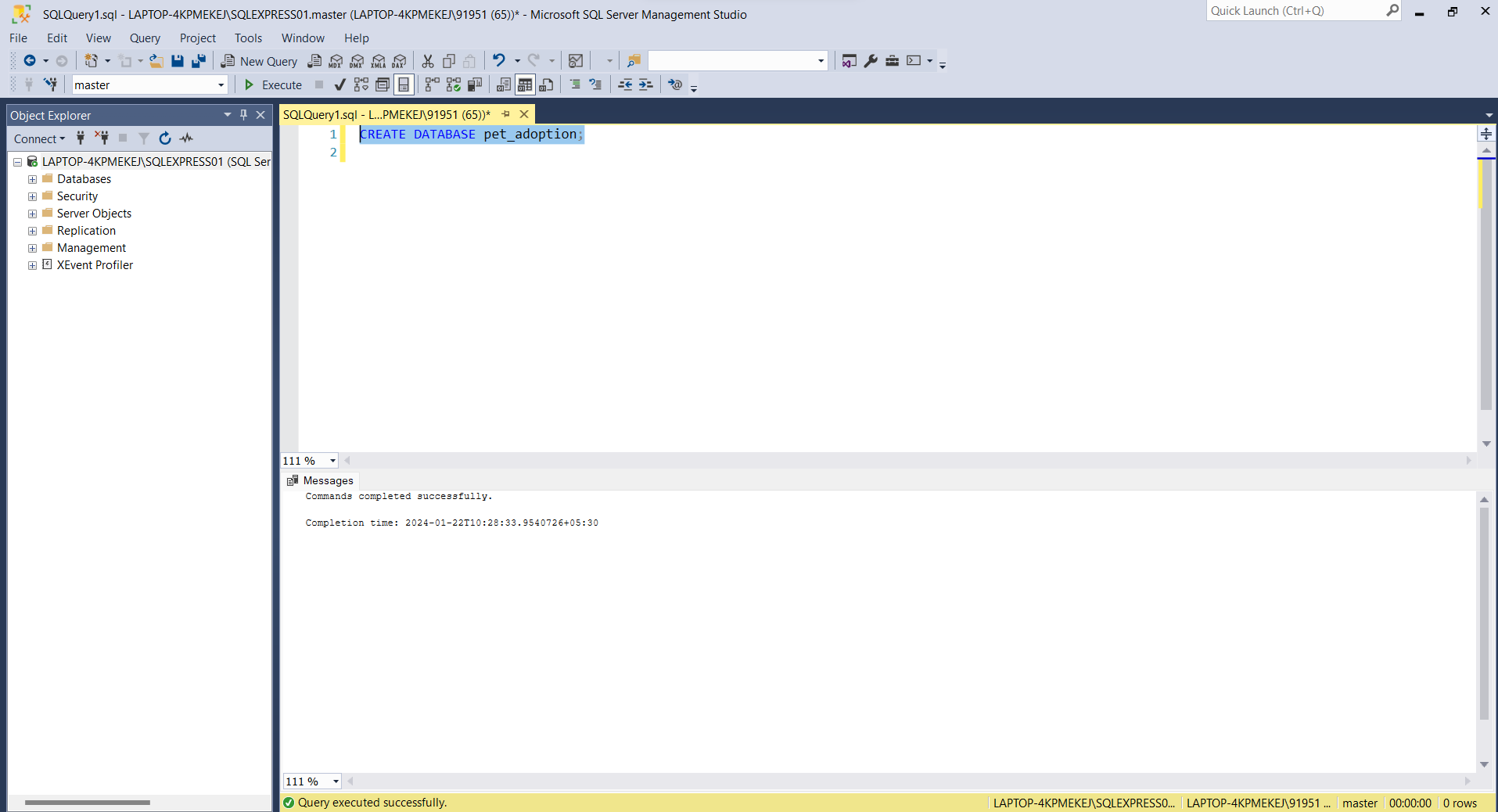
**DATA DEFINITION LANGUAGE ( DDL) :**

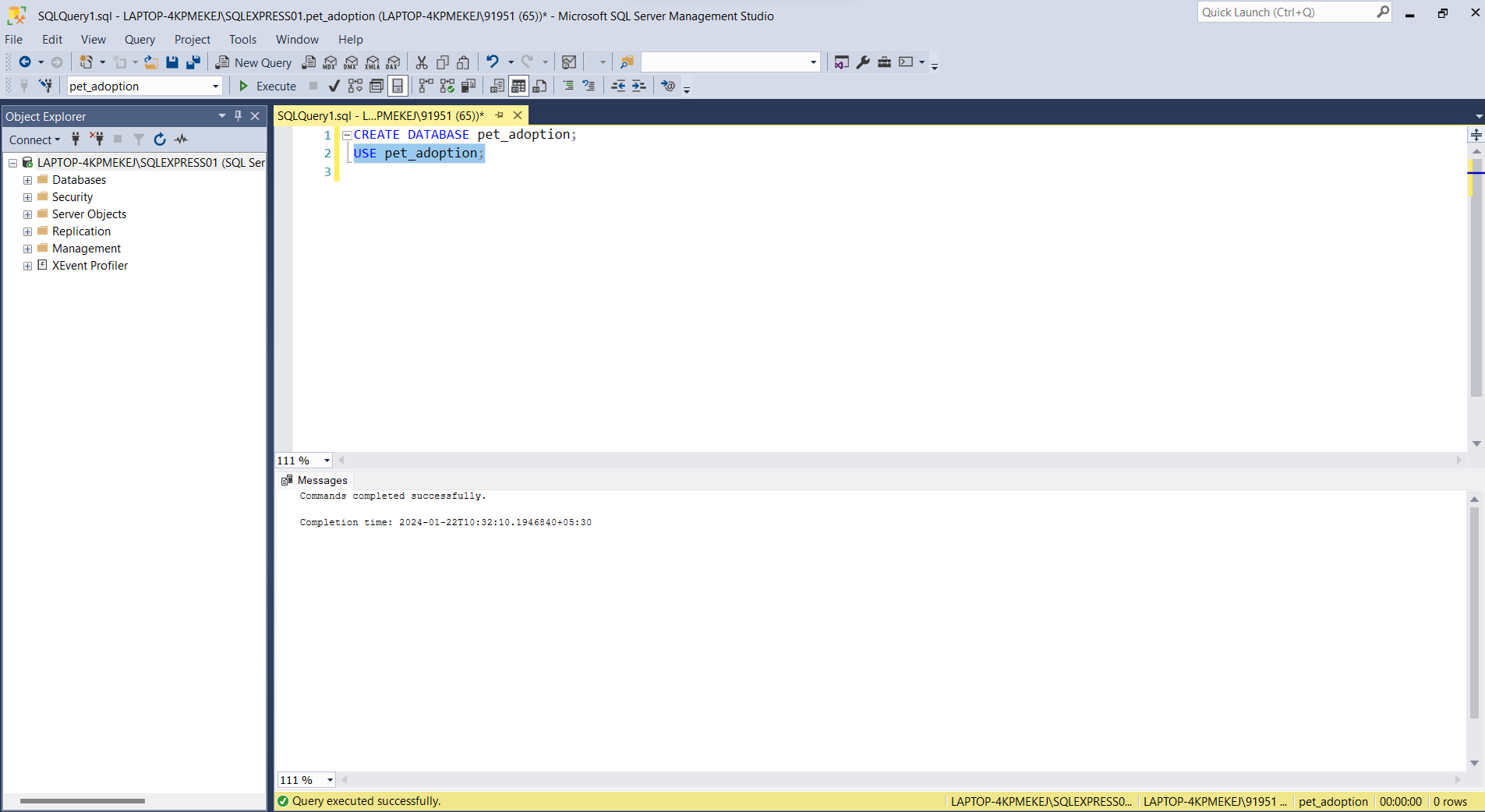
A data definition language (DDL) is a computer language used to create and modify the structure of database objects in a database. These database objects include views, schemas, tables, indexes, etc.

DDL comprises the primary commands listed below:

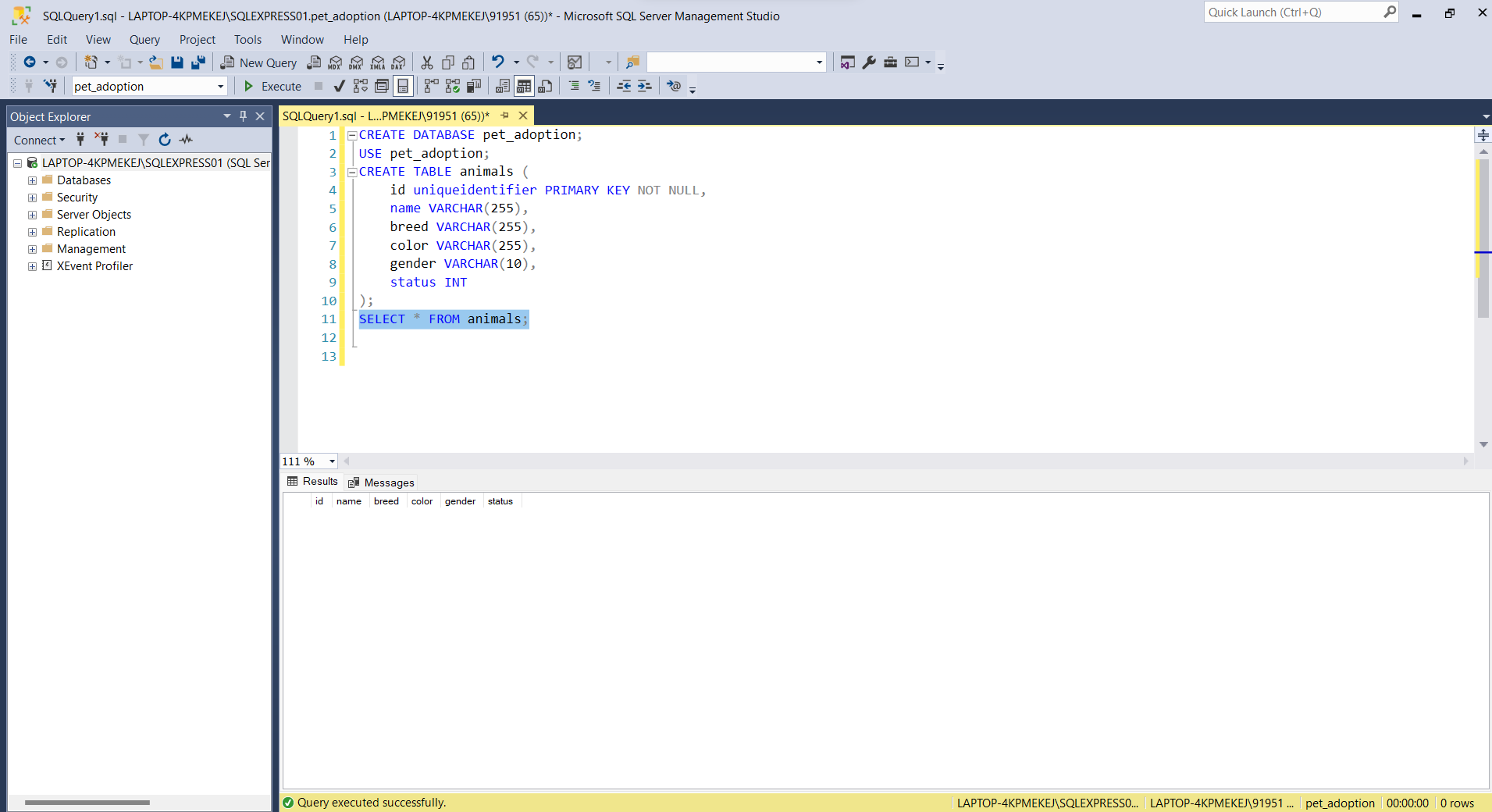
**CREATE**: This “CREATE” command is used to create database objects like tables, indexes, views, and stored procedures



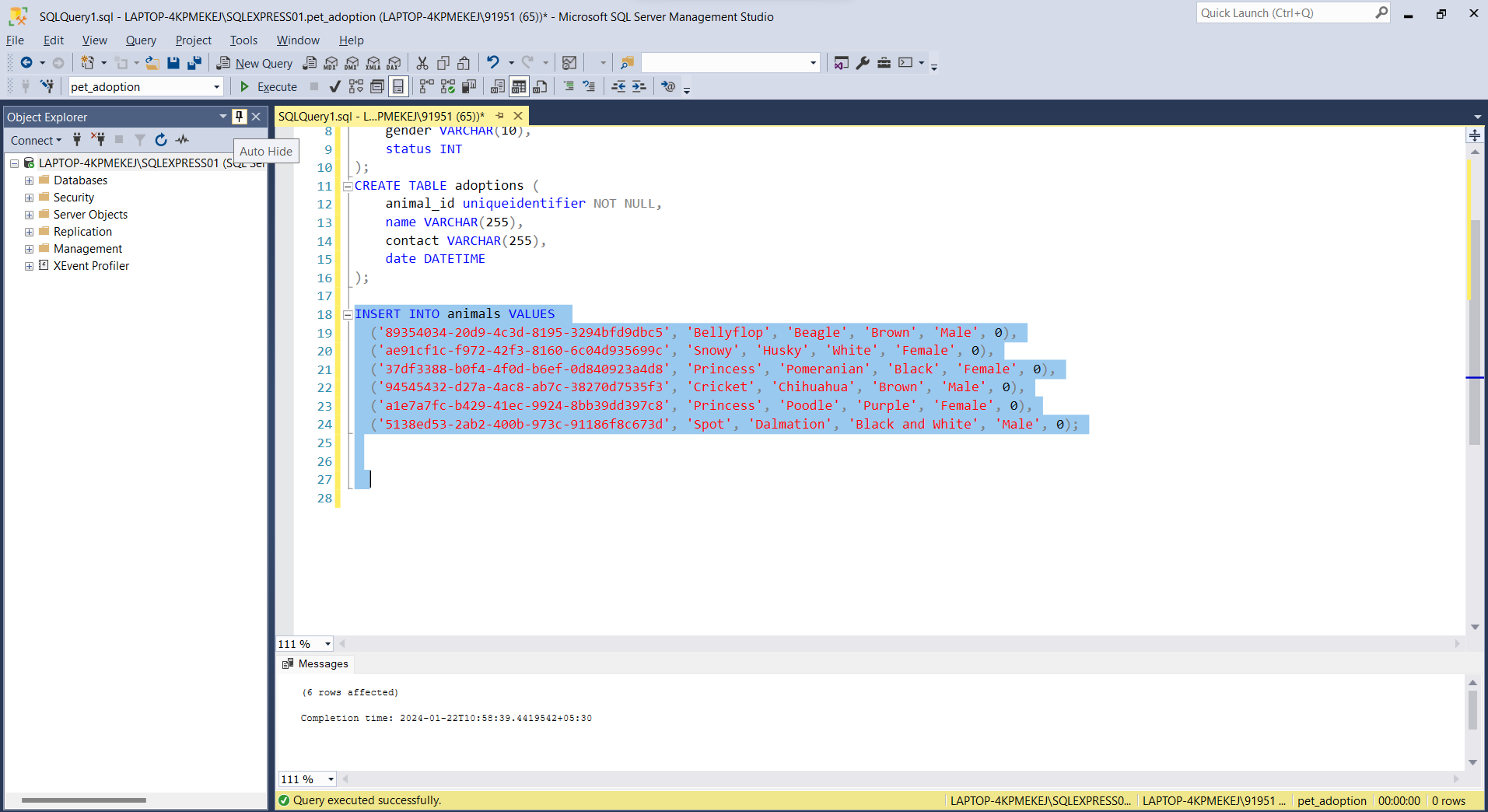
**USE:** The "USE" command in SQL is used to set the current database context, specifying the database on which subsequent queries and operations will be performed.



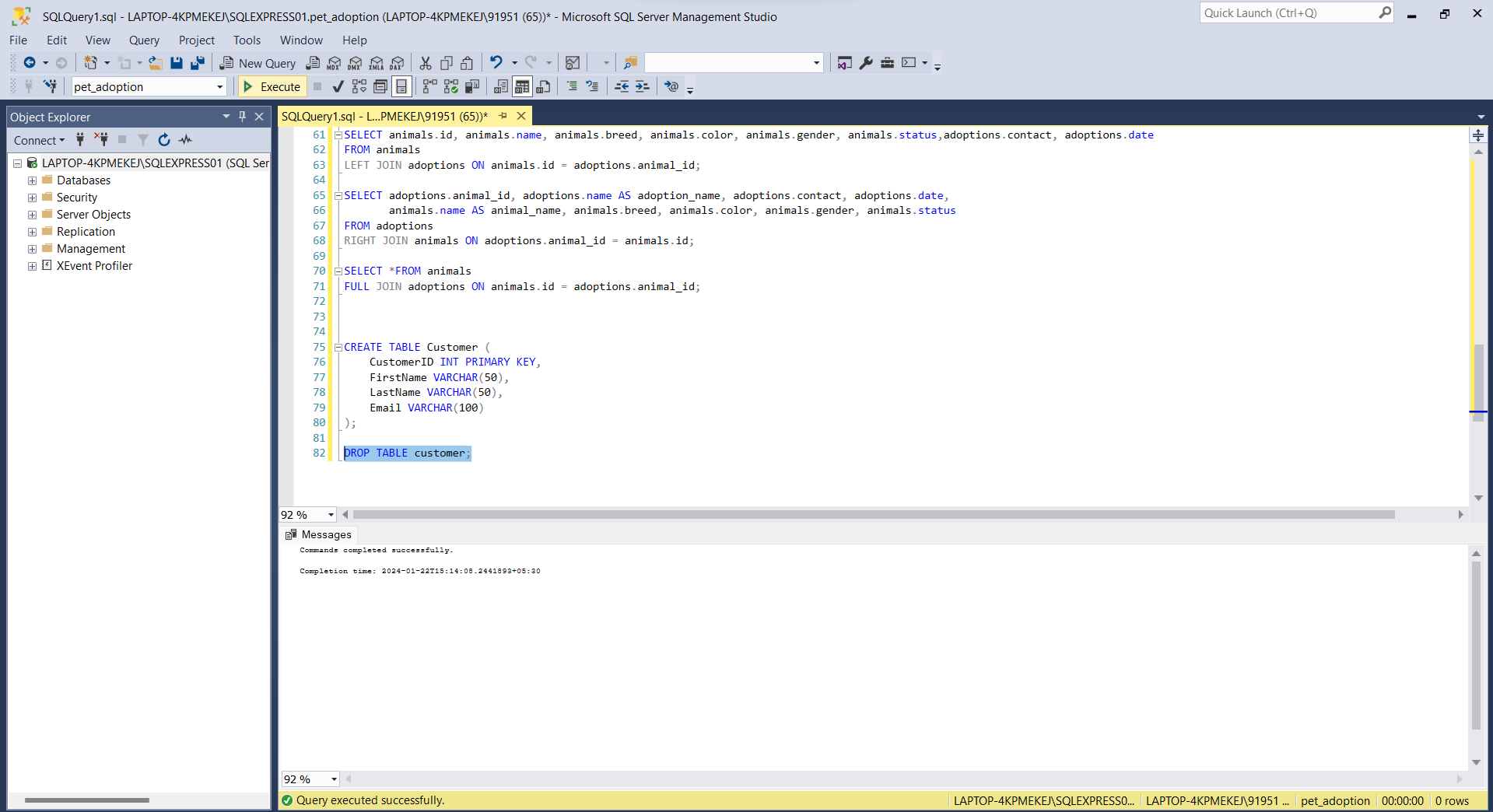
**SELECT:** The "SELECT" command in SQL is used to retrieve data from one or more tables in a database.



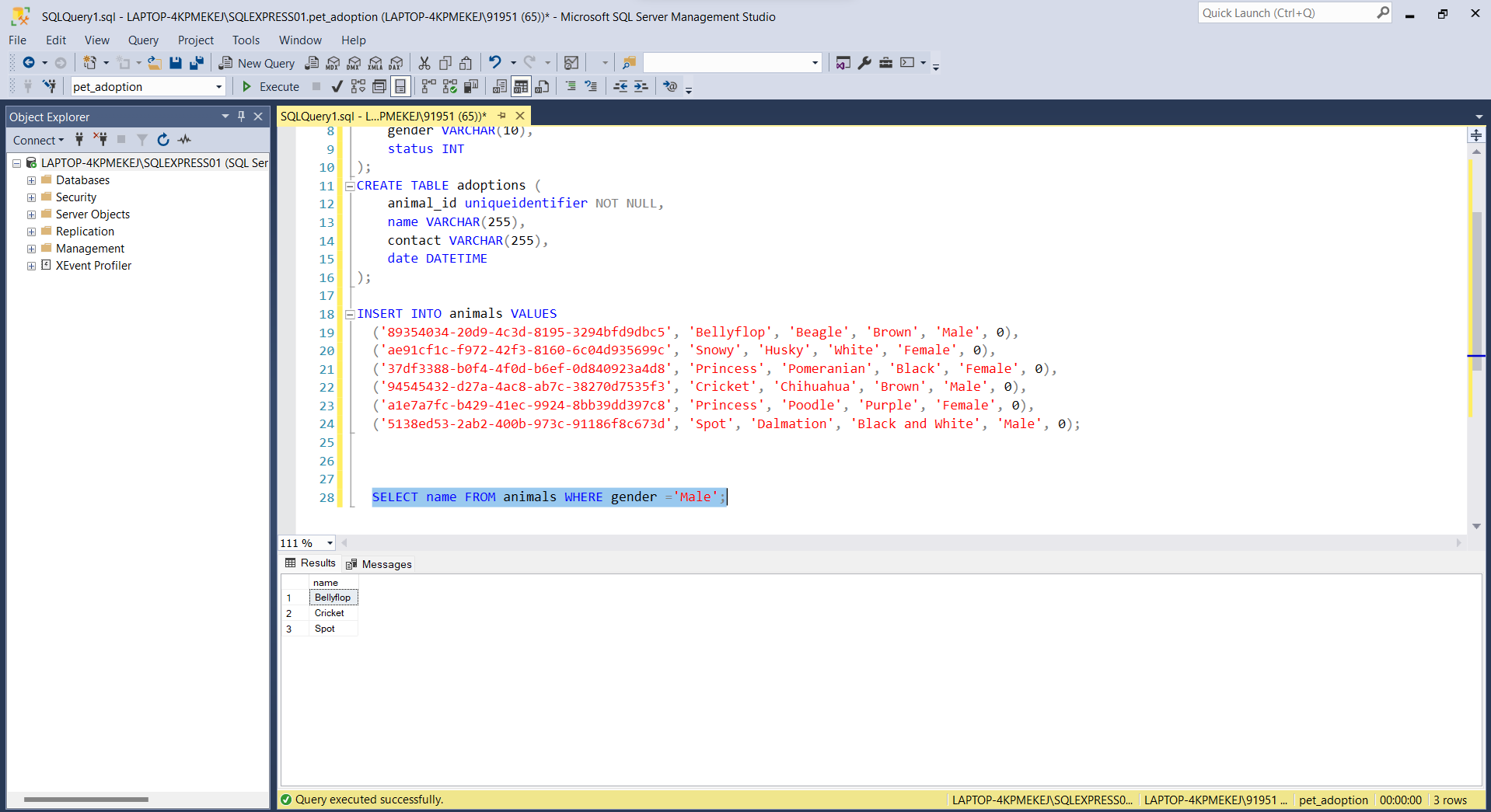
**INSERT:** The "INSERT" command in SQL is used to add new records (rows) into a table.



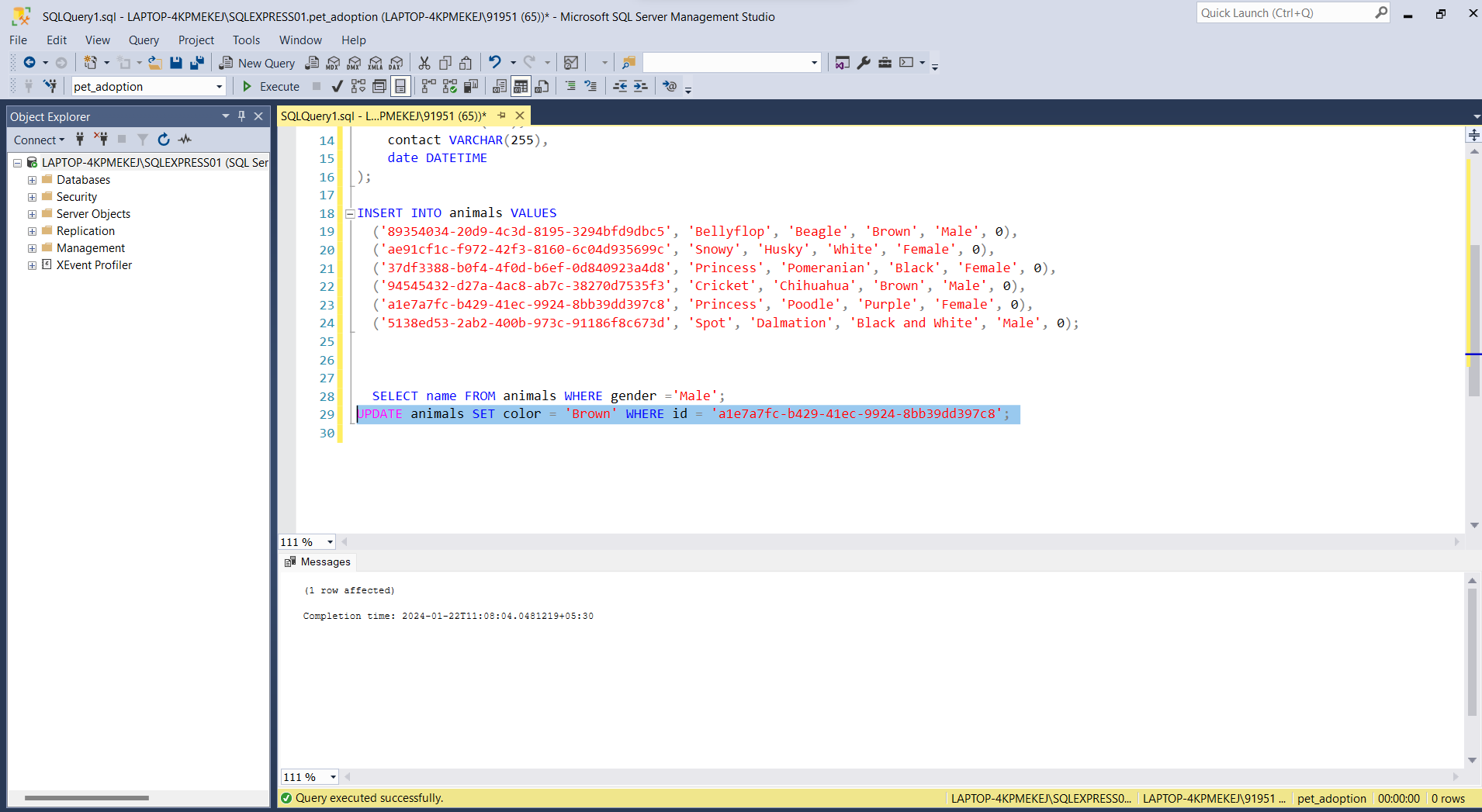
**DROP:** The "DROP" command in SQL is used to delete or remove a database object, such as a table, view, index, or schema.



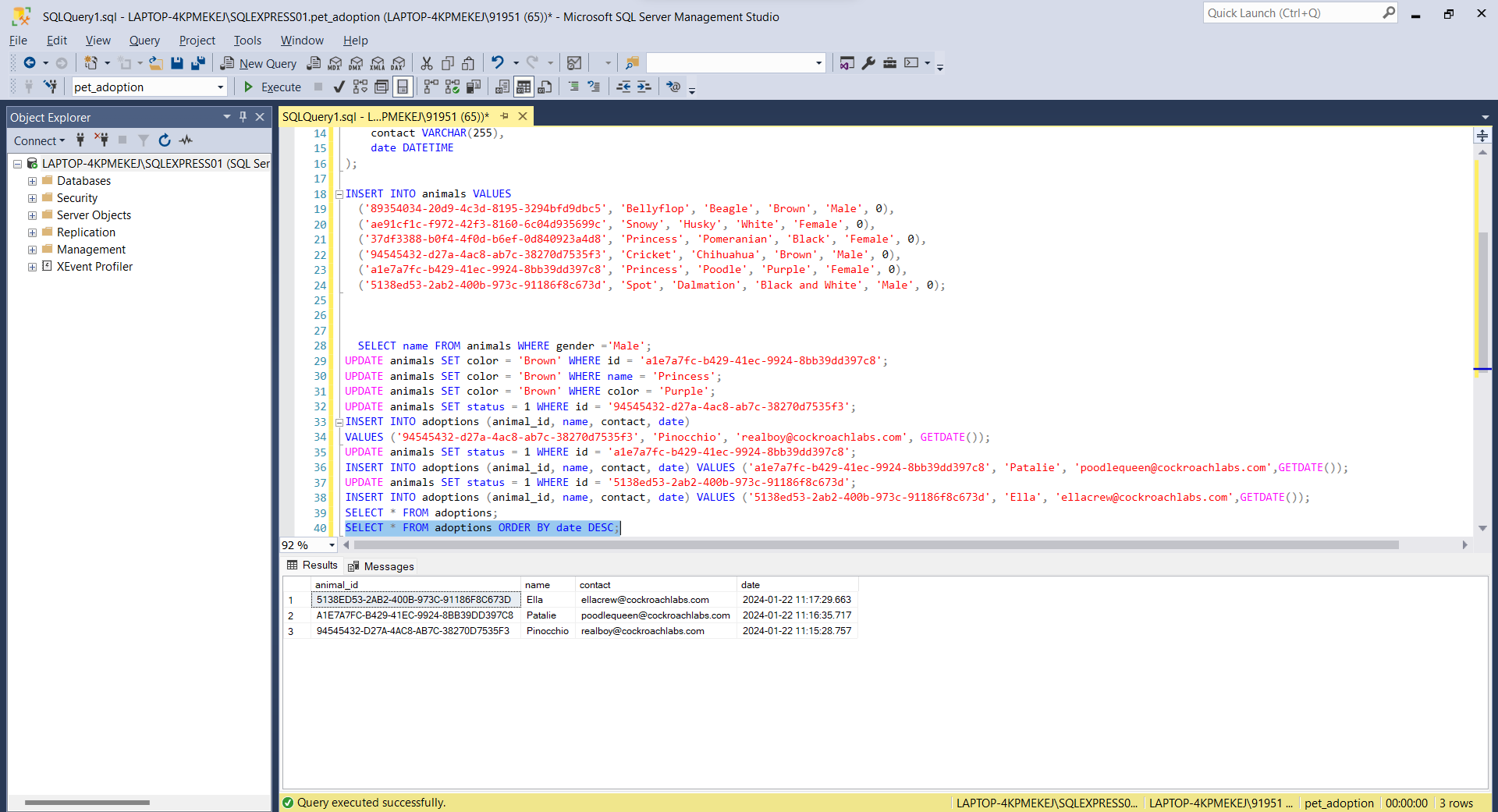
**WHERE:** The "WHERE" clause in SQL serves to selectively retrieve records by imposing specific conditions in a query. This allows you to narrow down the results to only include data that satisfies particular criteria from a table.



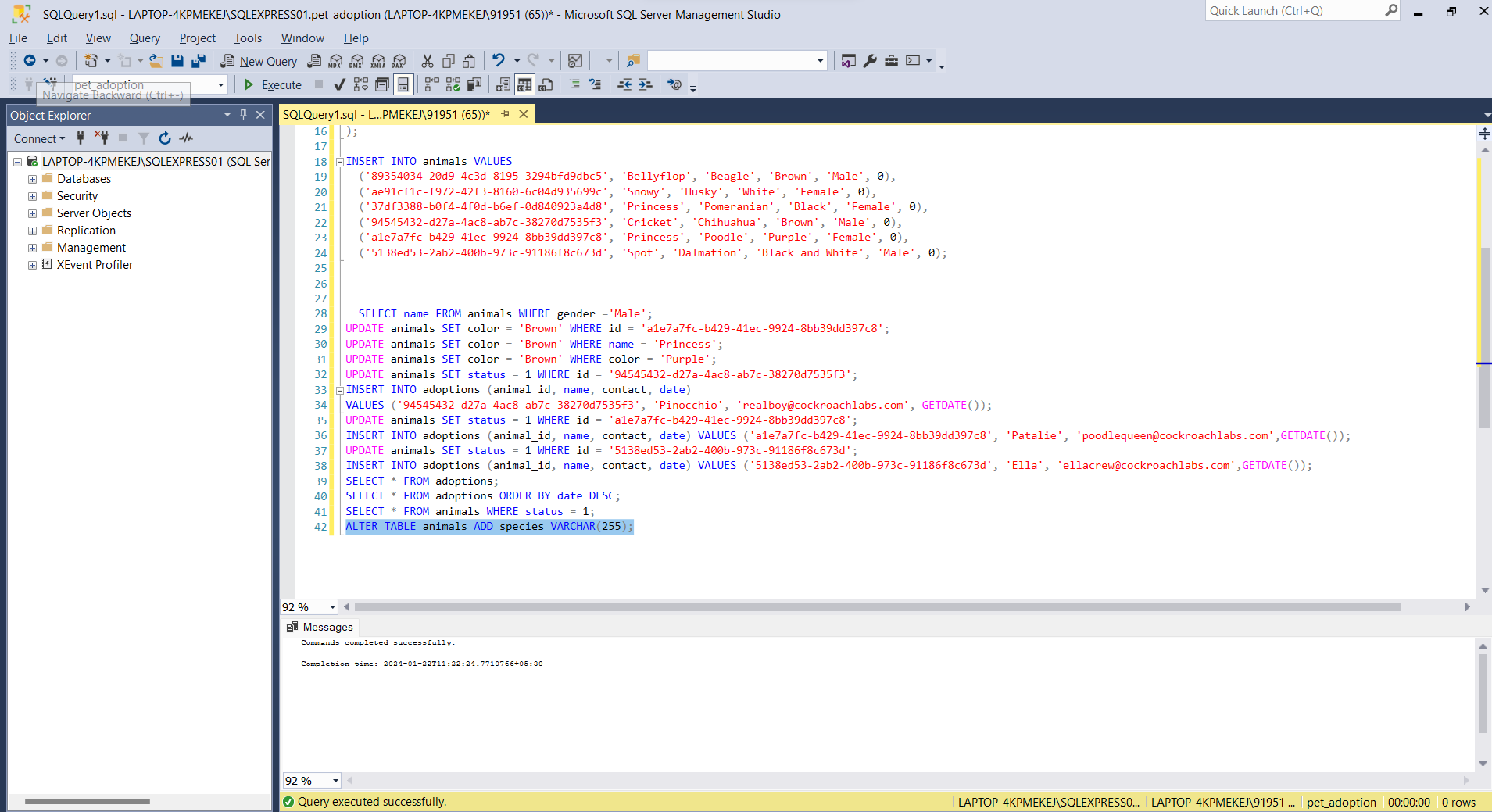
**UPDATE:** The SQL "UPDATE" command is employed to make alterations to existing records within a table. This involves adjusting the values of specified columns based on specified conditions. Essentially, it allows you to modify data in a table selectively.



**ORDER BY:** The SQL "ORDER BY" directive is employed to arrange the outcome of a query in either ascending or descending order, based on one or more specified columns.

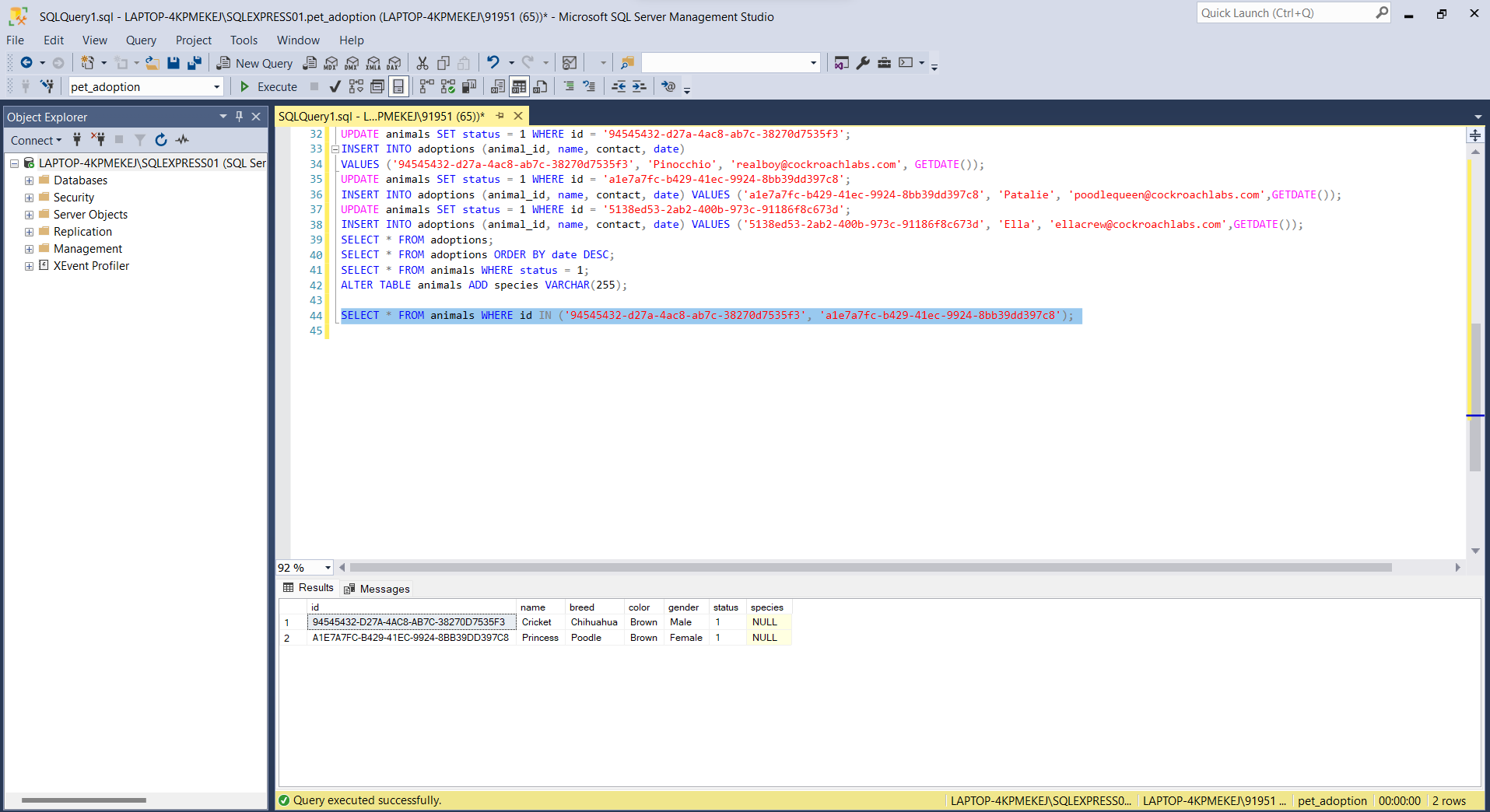


**ALTER**: The SQL "ALTER" statement serves to make changes to the configuration of established database elements like tables. This involves actions such as adding, modifying, or removing columns or constraints to enhance the structure of the database.

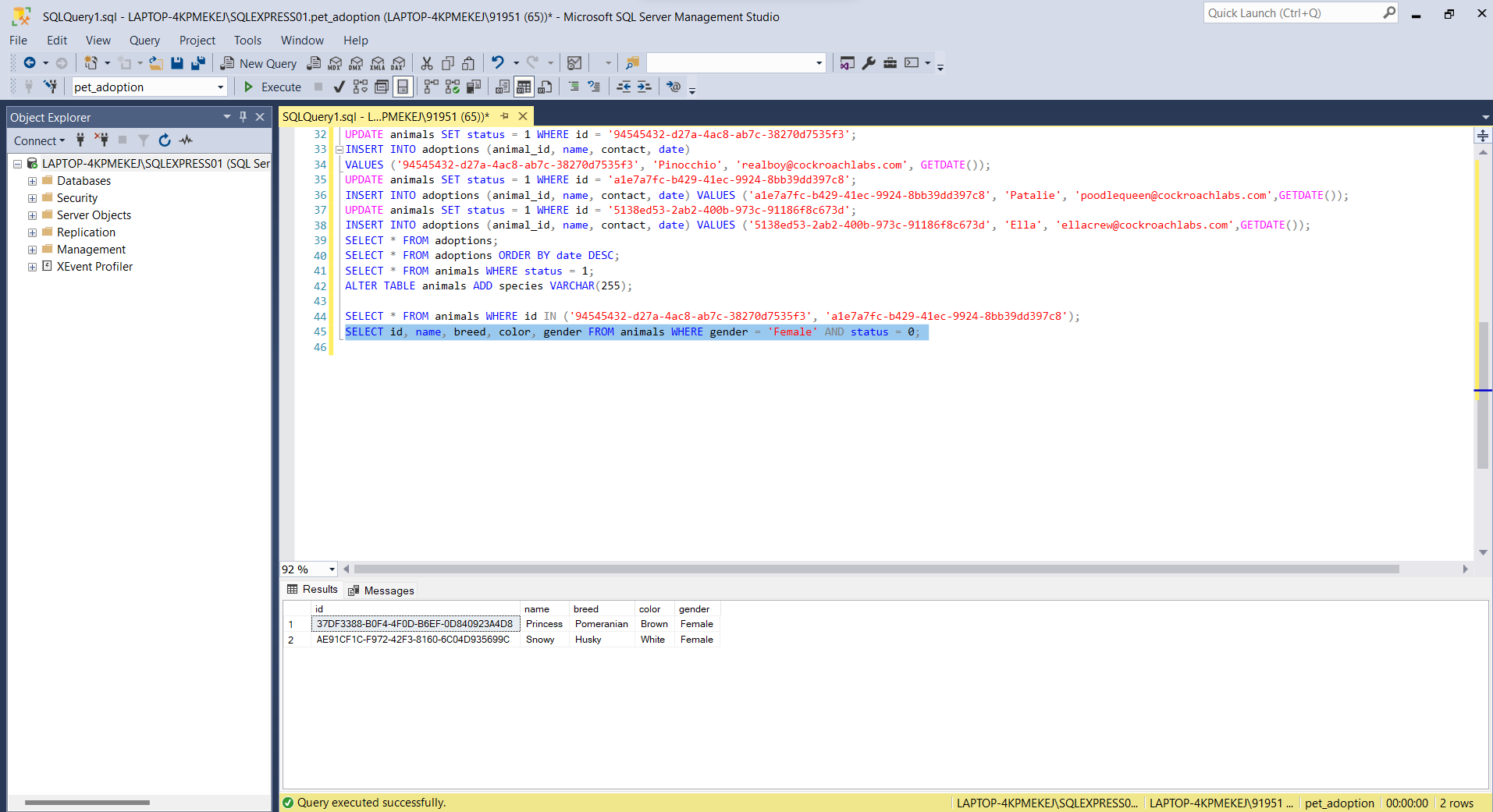


**OPERATORS:**

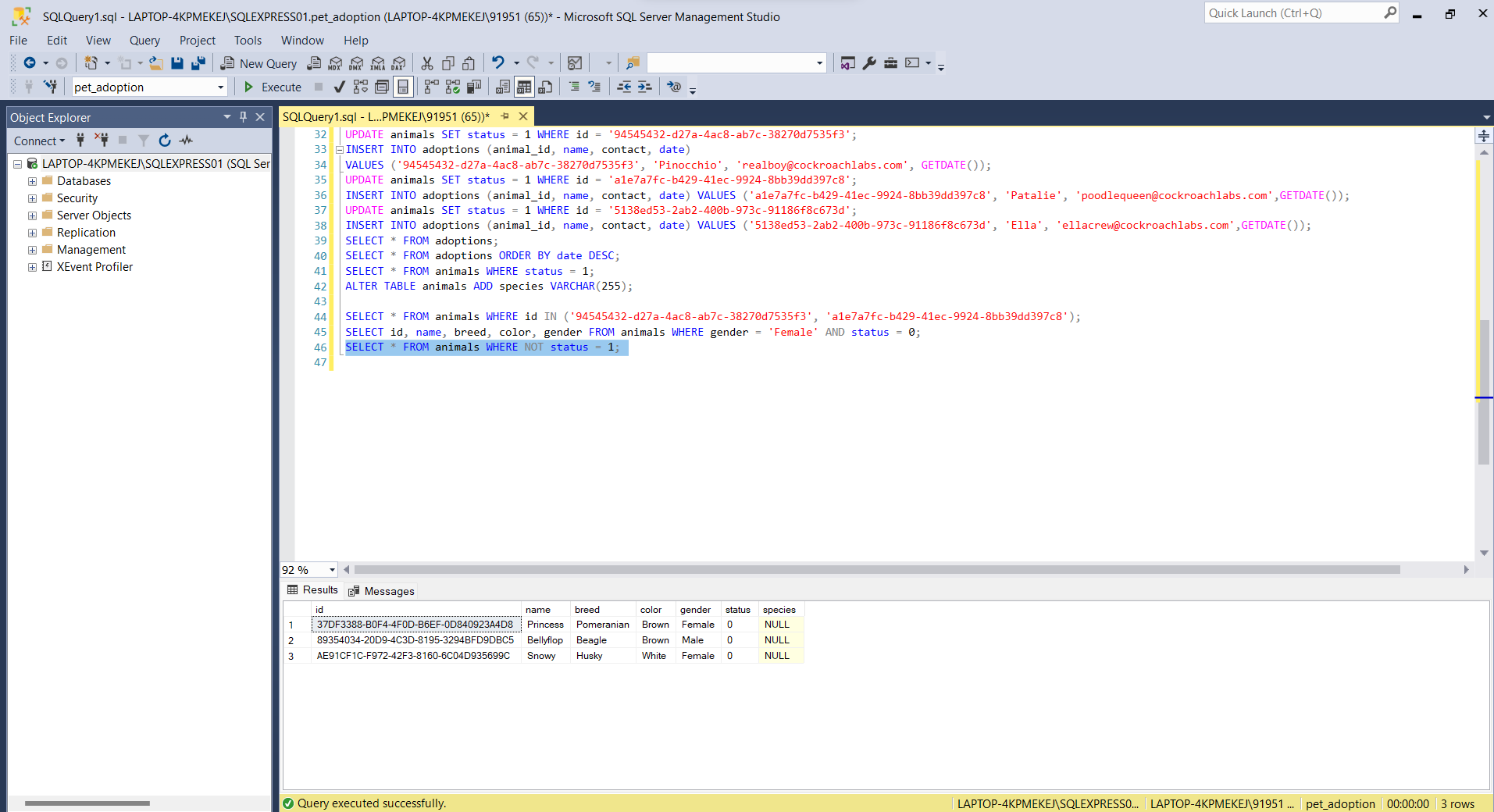
1. **IN OPERATOR:** The "IN" operator in SQL allows you to filter results by specifying a set of values. It retrieves rows where the value of a specified column matches any value within the provided list.



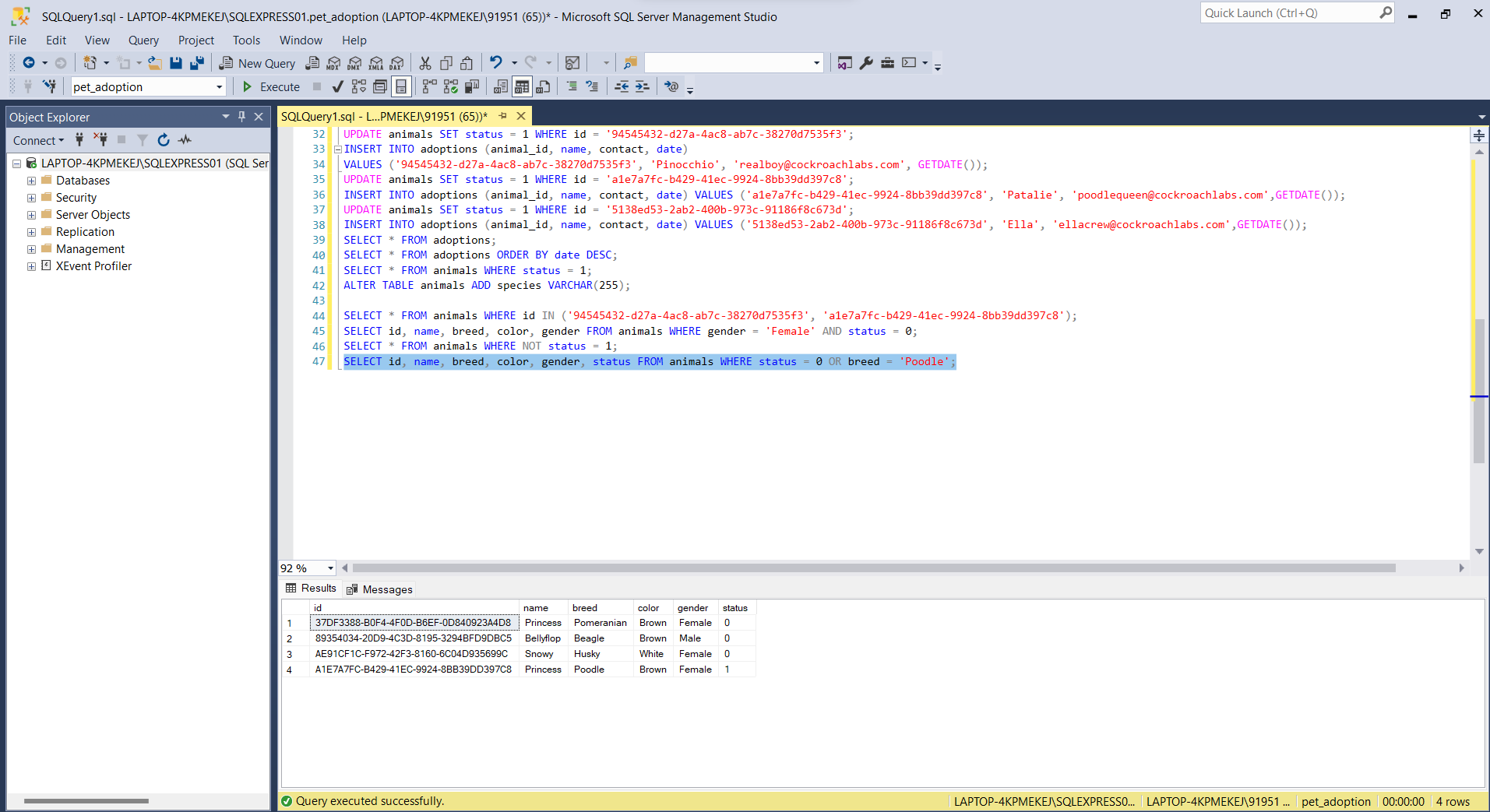
1. **AND OPERATOR:** The “AND” operator is used to combines two or more conditions but if it is true when all the conditions are satisfied.



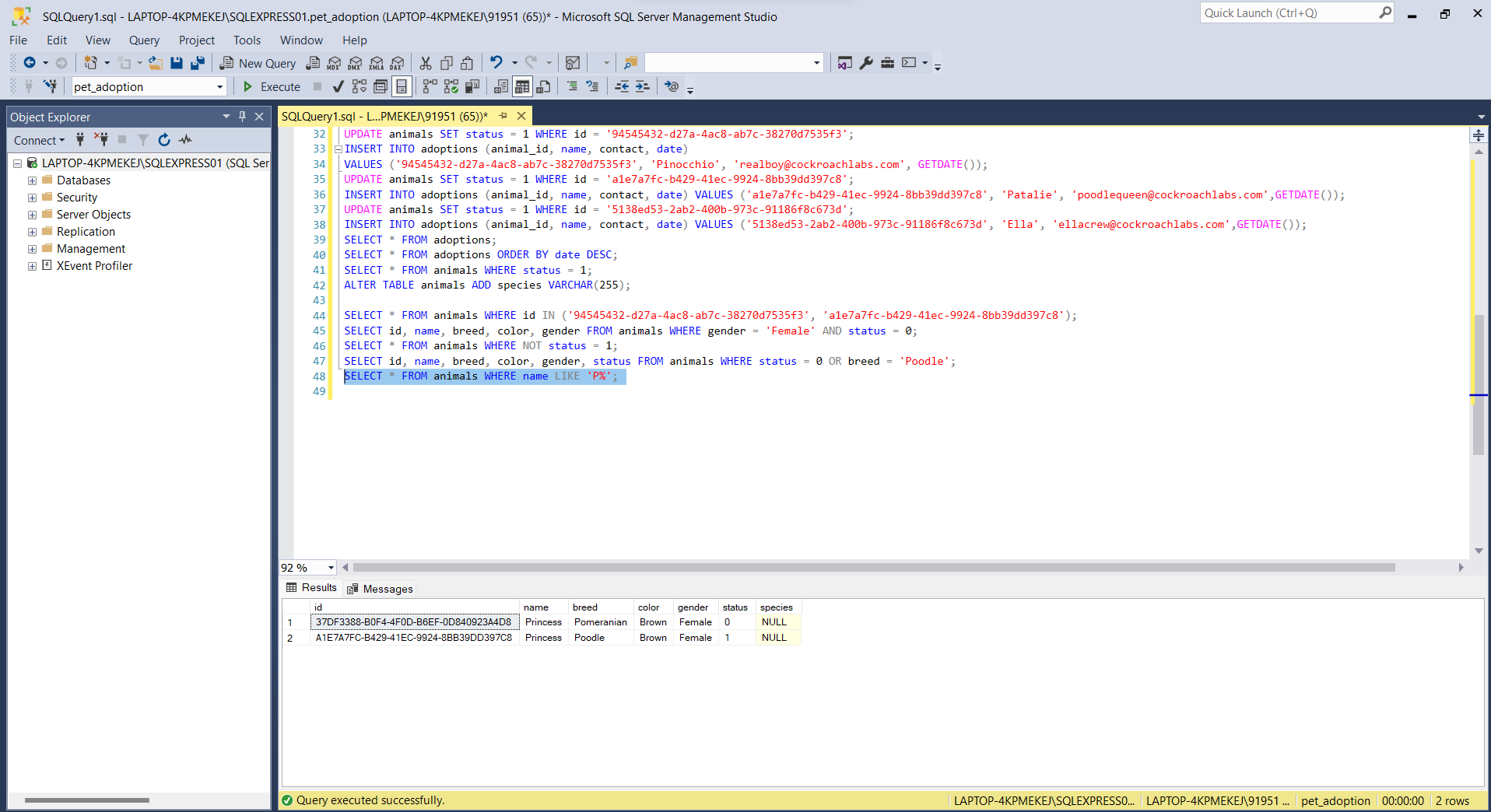
1. **NOT OPERATOR**: The "NOT" operator in SQL is used to negate a condition, returning rows that do not satisfy the specified condition.



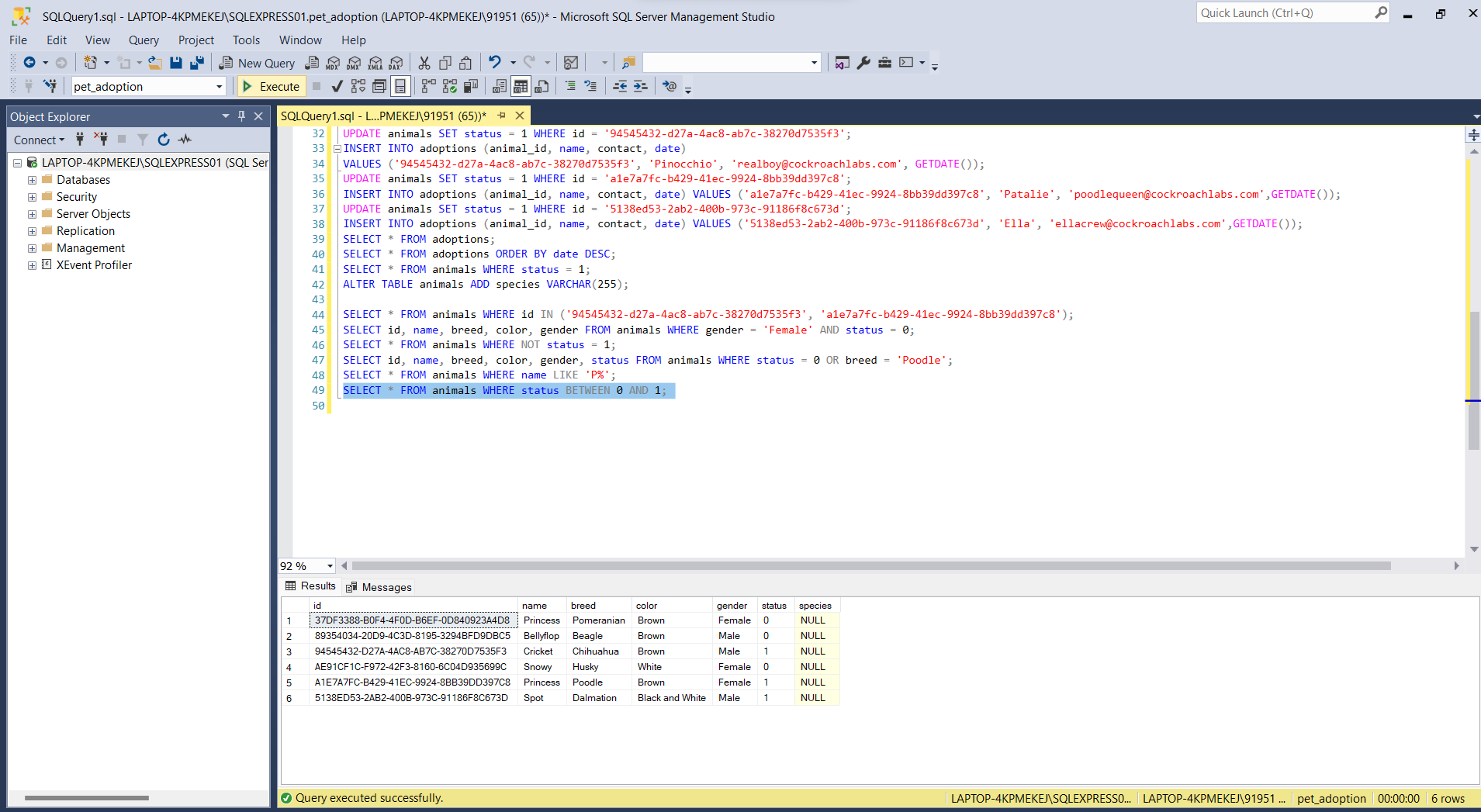
1. **OR OPERATOR**: The “OR” operator is used to combines two or more conditions but if it is true when one of the conditions are satisfied.



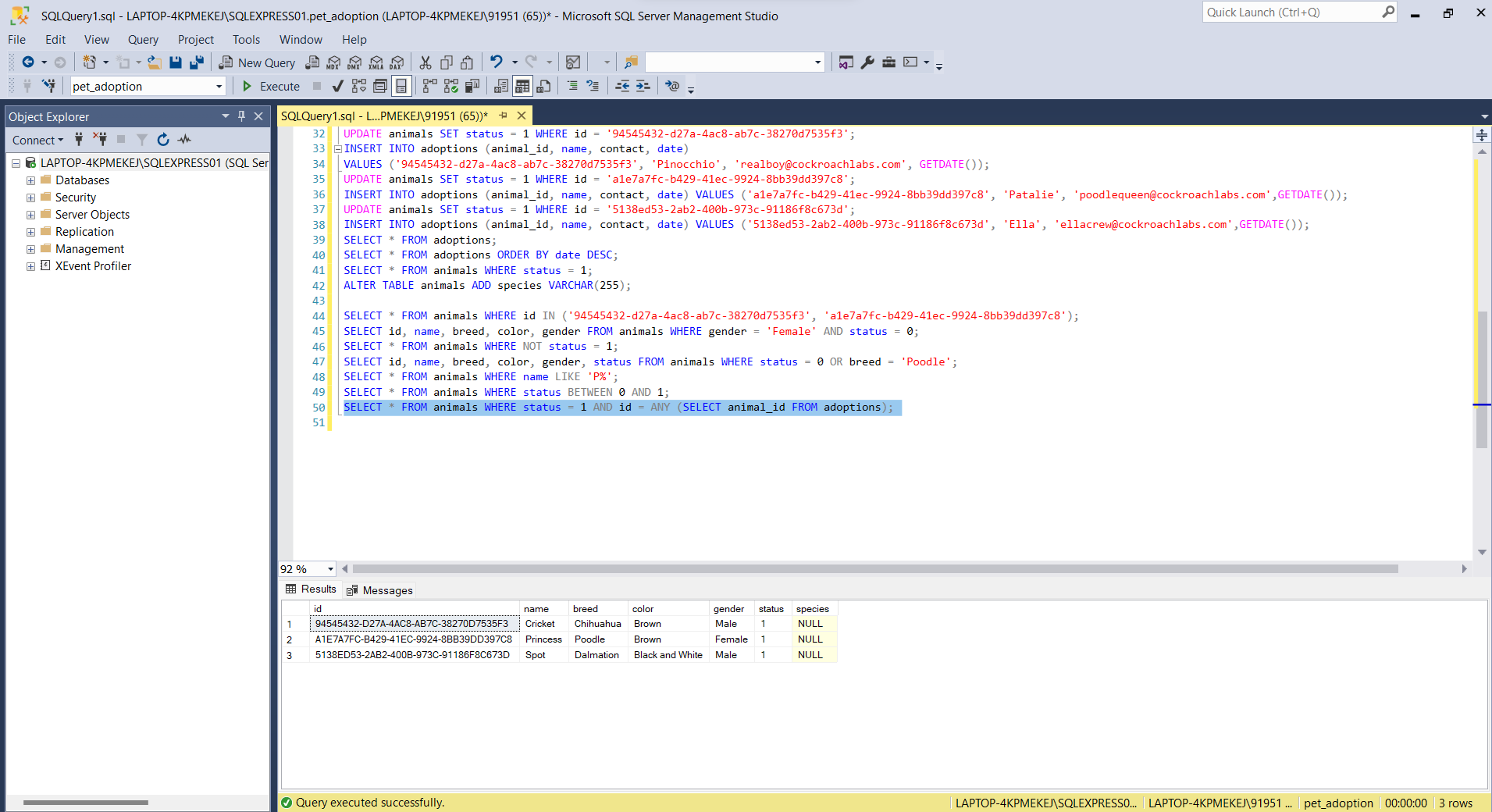
1. **LIKE OPERATOR**: The "LIKE" operator in SQL is used to search for a specified pattern in a column, allowing the selection of rows that match the pattern.



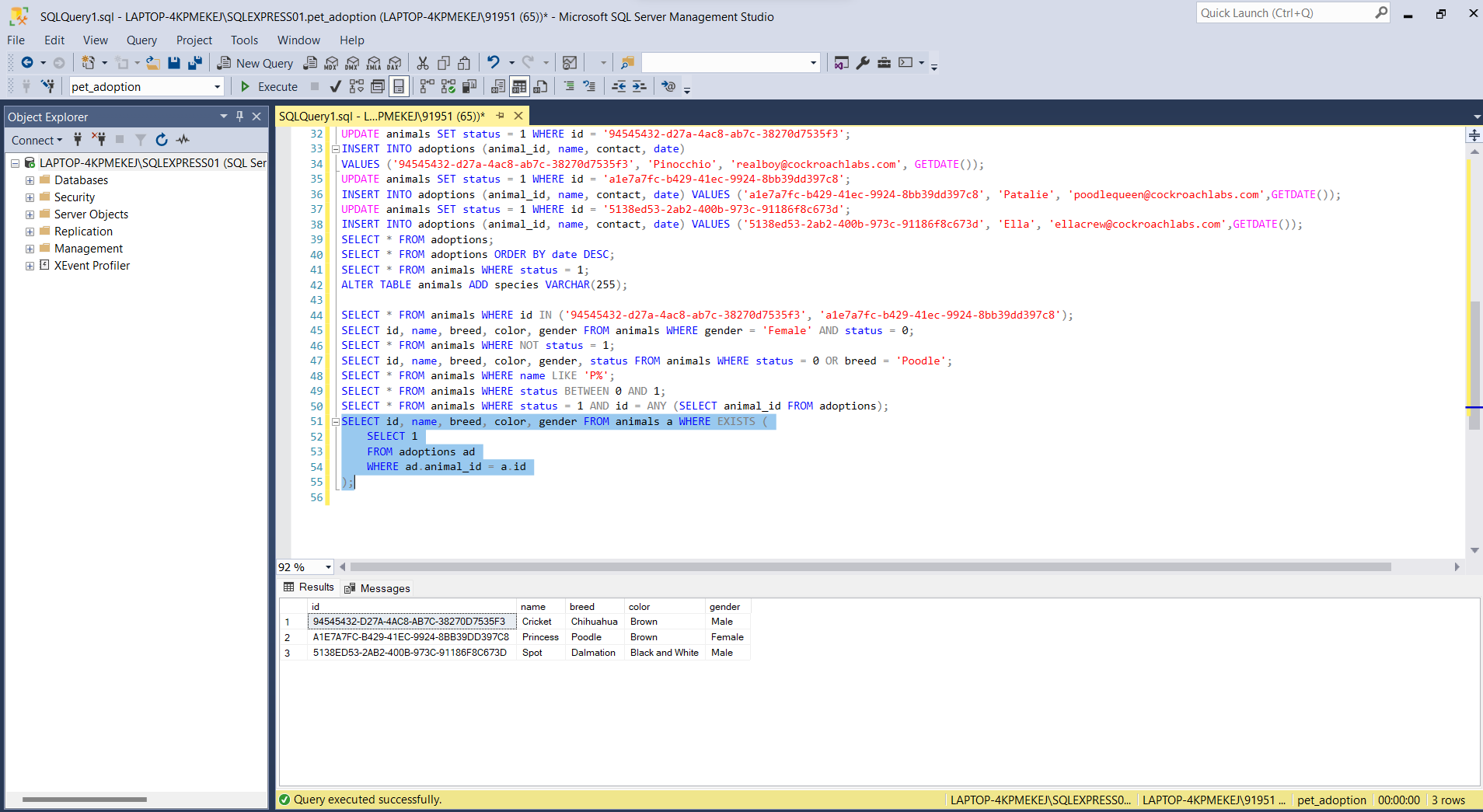
1. **BETWEEN OPERATOR**: The "BETWEEN" operator in SQL is used to filter query results based on a specified range of values for a given column.



1. **ANY OPERATOR**: The "ANY" operator in SQL is used to compare a value to any value in a set of results returned by a subquery.



1. **EXISTS OPERATOR**: The "EXISTS" operator in SQL is used to check whether a subquery returns any rows and returns a Boolean value (True or False) based on the existence of those rows.



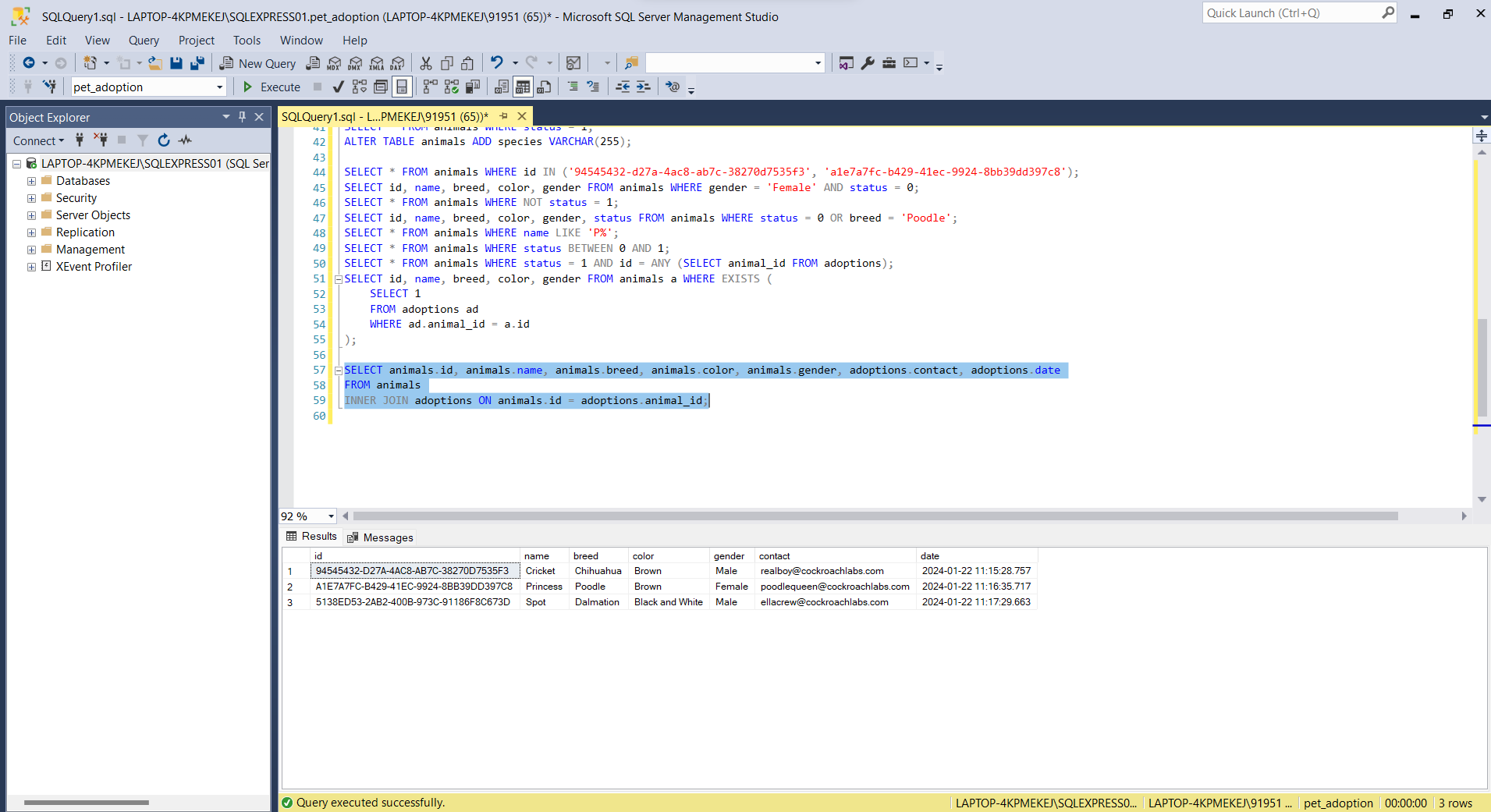
**JOINS**

The "JOIN" command in SQL is used to combine rows from two or more tables based on a related column between them.

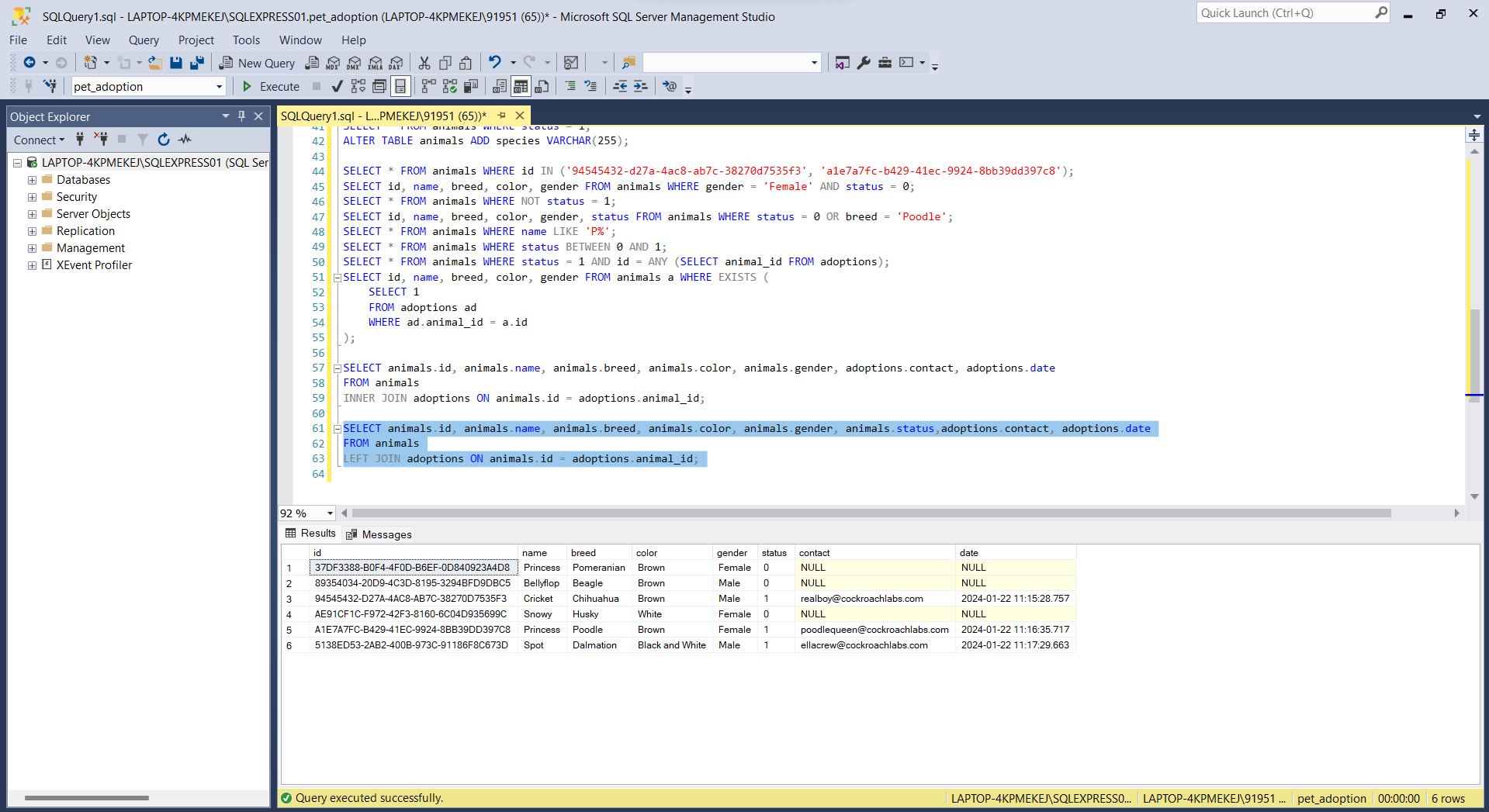
Different types of Joins are as follows:

* **INNER JOIN**
* **LEFT JOIN**
* **RIGHT JOIN**
* **FULL JOIN**

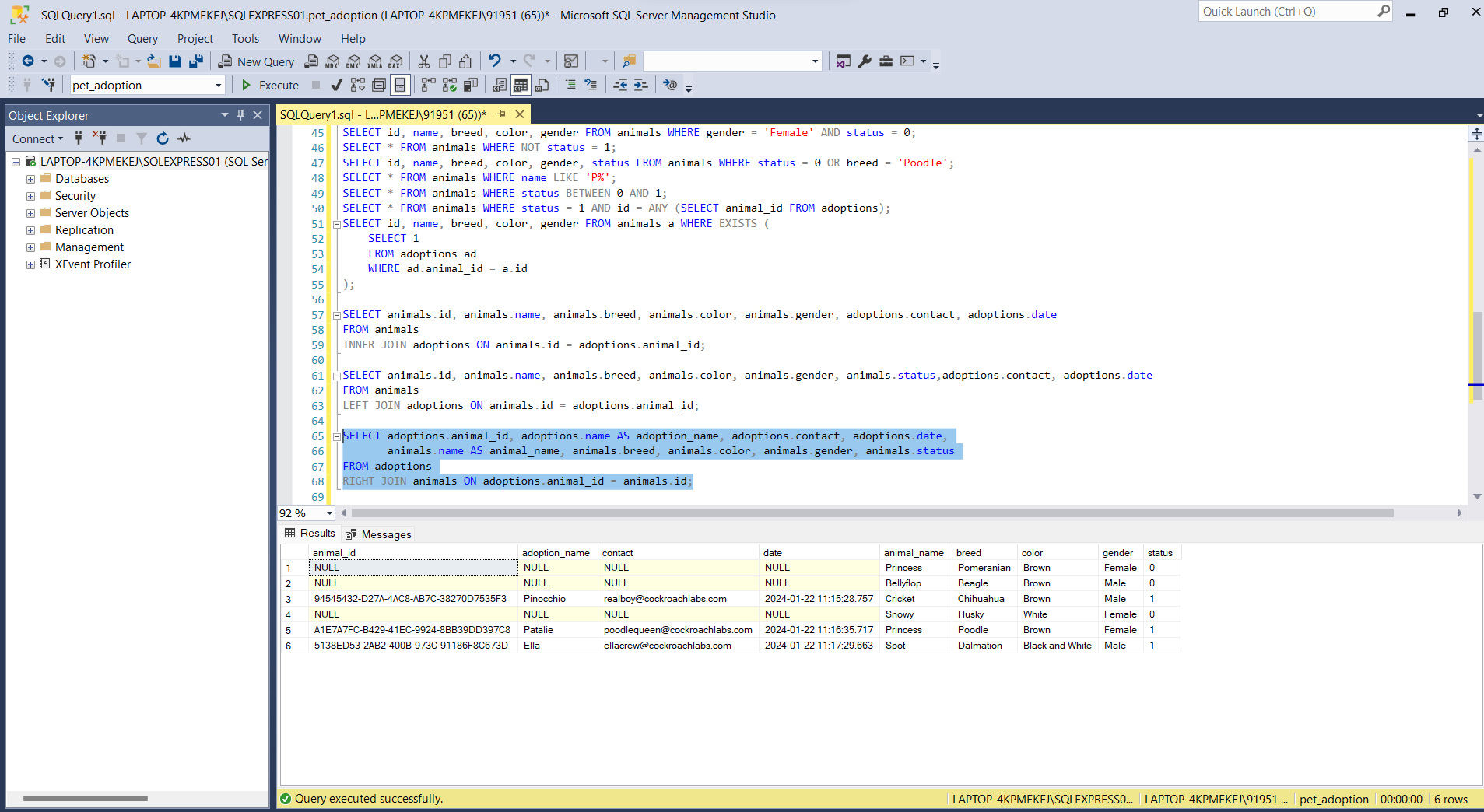
**INNER JOIN:** An inner join in SQL is like combining information from two tables based on a shared condition. It only includes rows where there is a match between the specified columns in both tables. It's like putting together puzzle pieces that fit.



**LEFT JOIN**: A left join in SQL is like showing all the rows from the left table and matching rows from the right table. If there is no match in the right table, you still get the rows from the left table, but with NULL values in the columns from the right table.



**RIGHT JOIN**: In a right join in SQL, all the rows from the right table and the matching rows from the left table are included in the result. If there is no match, the result will still show all the rows from the right table, and the columns from the left table will have NULL values.



**FULL JOIN:** A full join in SQL combines rows from two tables based on a common condition and includes all rows from both tables, even if there is no match. It includes matched rows as well as unmatched rows from both tables in the result set.

