**RESPONSE**

Comments are used in SQL to explain or stop the execution of a section of code. The three types used in SQL are single line, multi-line or in-line comments.

1. **Single-line comments** begin with double hyphens and span a single line.

**Example:**

--This comment is being used to explain that this is a SQL SELECT Query

**SELECT \* from book**

Keep in mind when using single line comments:

* The two hyphens must be on the same line and must not be separated by a space.
* Single line comments cannot be continued to the next line.
* Single line comments can be started wherever a space is valid (except within a delimiter token or between 'EXEC' and 'SQL').

1. **Multi-line comments** can be anywhere in your SQL statement. This comment starts in one line with /\* ending in a different line with \*/ and anything between these will be ignored by the program.

**Example:**

/\* This is a multi-line comment being

used to explain that this code will pull

all the records from the author table \*/

**SELECT \* FROM author;**

1. **In line comment** is the same as a multiline comment enclosed between /\* and /\*; however inline comments are present on the same line where the SQL query is provided.

**Example:** In-line comment used to prevent the program from running specific code.

SELECT \* FROM tblemp;   
SELECT \* FROM **/\* tblcompany;   
SELECT \* FROM tblorders;   
SELECT \* FROM \*/** tbldept;

Keep in mind when using multi-line and in-line comments:

* The /\* must be on the same line and must not be separated by a space.
* The \*/ must be on the same line and must not be separated by a space.
* Bracketed comments can be continued to subsequent lines.
* Multi-line comments can be started wherever a space is valid (except within a delimiter token or between 'EXEC' and 'SQL').

SOURCE: Programiz. “SQL Comments”. https://www.programiz.com/sql/comments

**DISCUSSION RESPONSES**

**#1**

**Emanuel Pagan**

Hi Emanuel,

You did a great job covering primary keys. The use of a credit union account was easy to follow, since it included enough information for you to really expand on the reasons primary keys are used and how they ensure data integrity. Pointing out that using the SSN or member number as the primary key was perfect. Plus, mentioning how other fields may not work as well because they could change at any time helped distinguish why the primary key needs to be unique. Your example really supported your explanation about primary keys. I like that you used an example that relates to daily life, because its easy to relate to and understand.

**#2**

**Joshua Gruidl - Comments in SQL**

Hi Joshua,

You did a nice job breaking down the different comments and describing how to use them. Adding the heads up about some versions of SQL Server’s inability to support the single line comment was a great addition to your post. I wouldn’t have thought about this if you hadn’t mentioned it. It makes me wonder how developers keep track of the different ways to add comments for the various languages with so many nuances out there.

It’s amazing how often you use comments when coding. They come in handy when you’re looking at unfamiliar code. I’ve found comments useful when researching code examples for different projects. They’re like a lifeline, especially if you’re new to coding.

**#3**

**Darius Dinkins - Primary Key**

Hi Darius,

This was a great post explaining the use of primary keys. It was fantastic that you used an example to illustrate the use of a primary key and one explaining why you wouldn’t need one. We haven’t focused much on the fact that you don’t always need a primary key, so the addition of this explanation was helpful. The examples you used were great, since they are relatable to the average person making it easier to understand.