**Task 1**

*Note: Tables are generated using SQL Server*

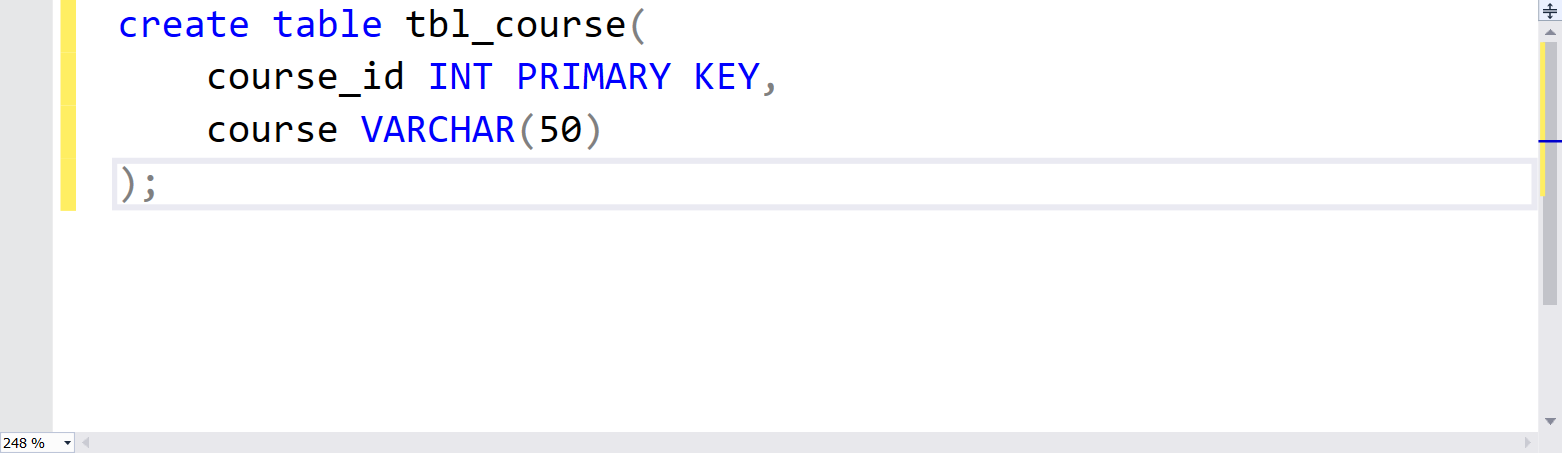
1- Consider the following tables, the course table and the instructor table. State the name of the join for the following output and **explain** it

The following join is **Full Join**

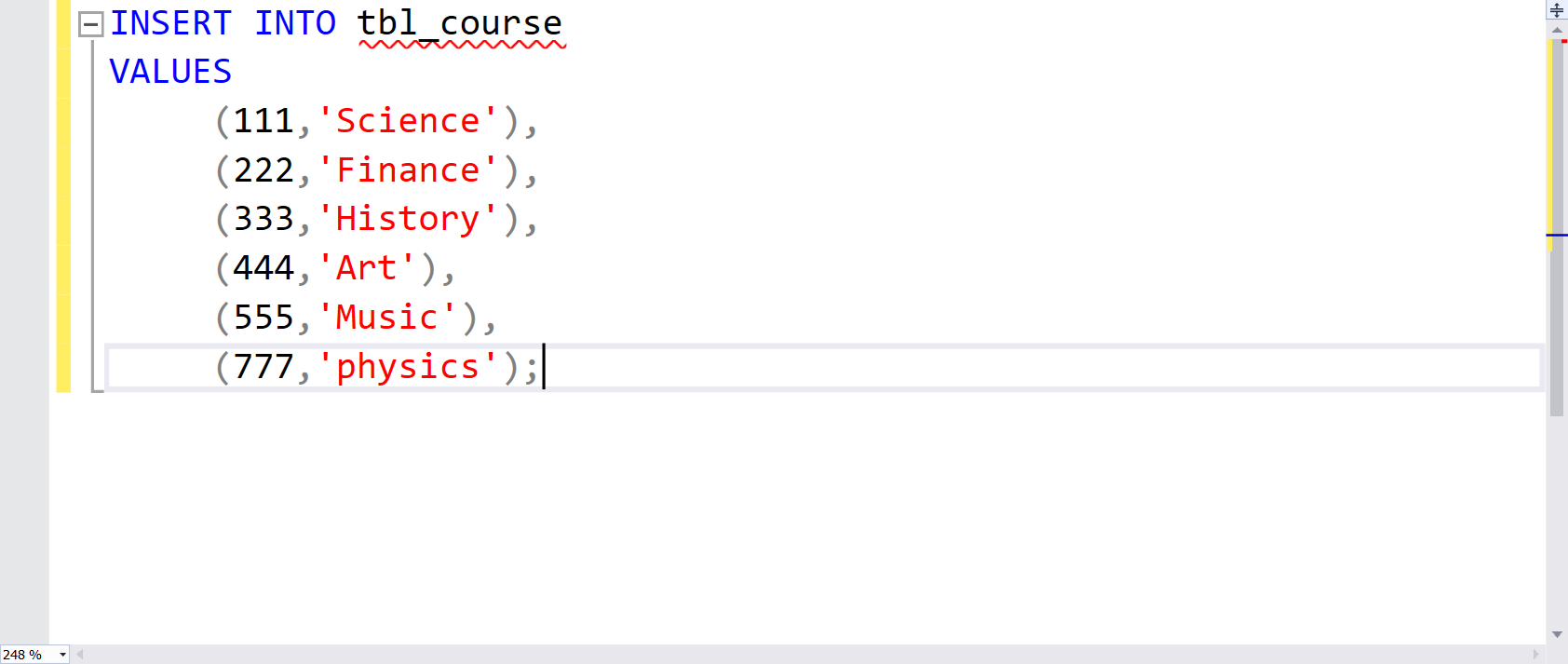
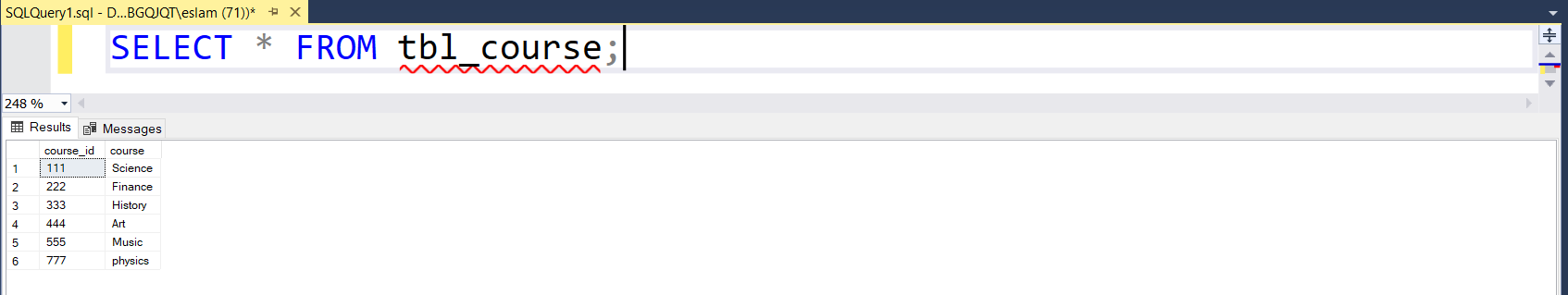
A full outer join combines all records from both tables being joined,

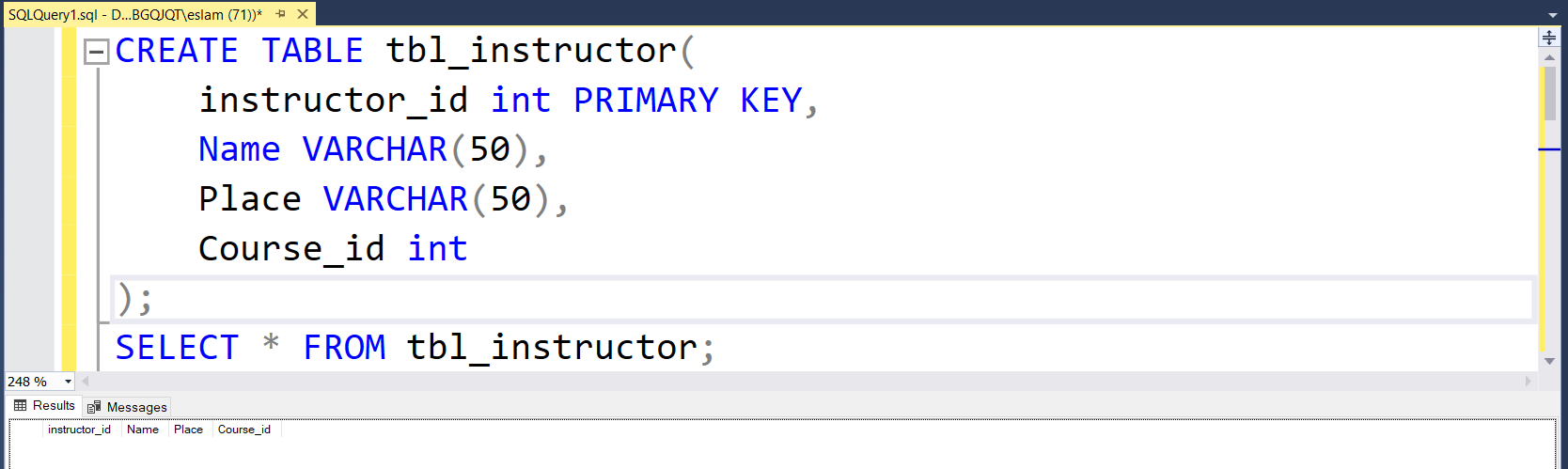
2- **Create and manage** advanced query to produce the above output.

**Create Course Table**

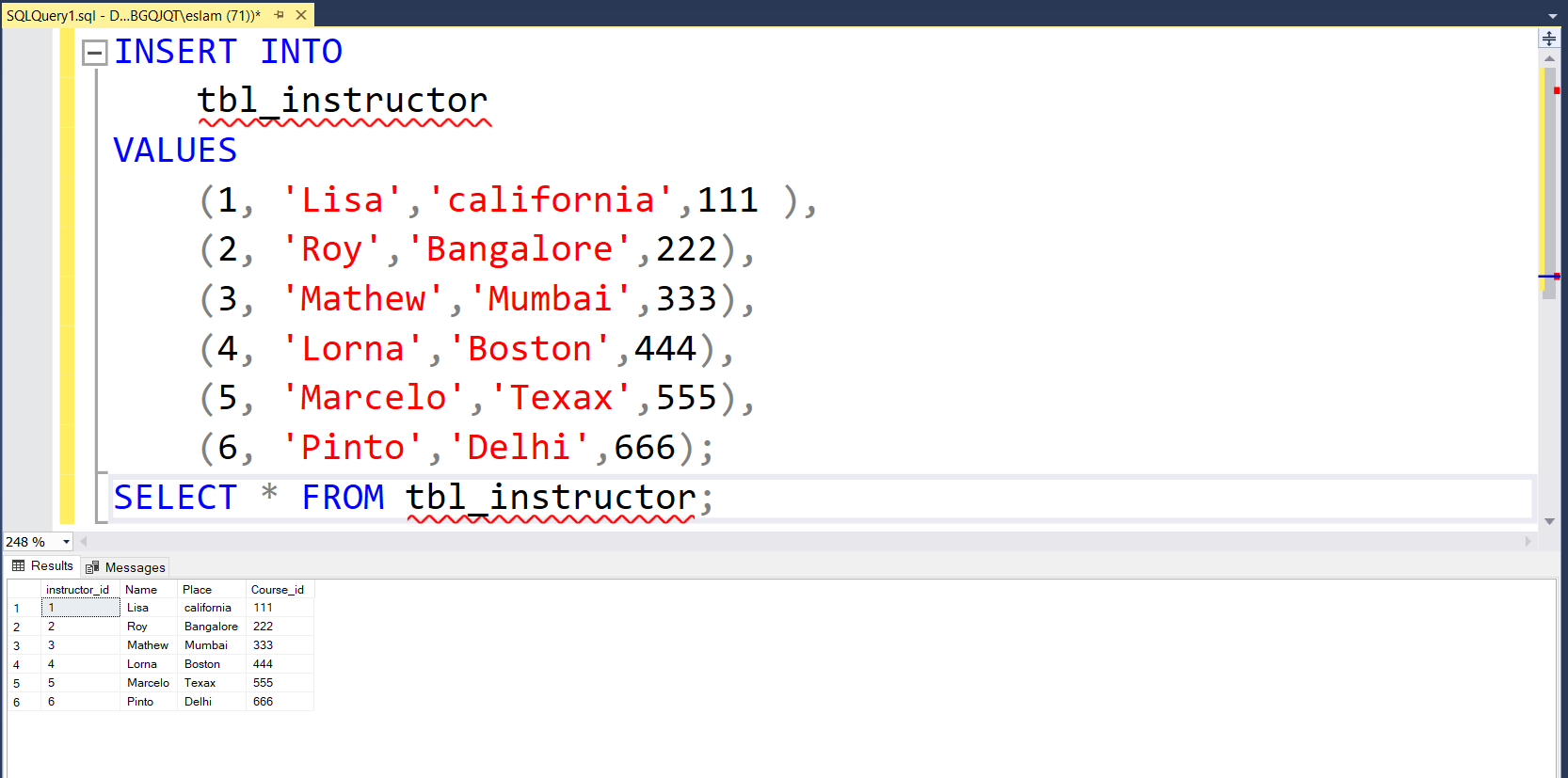


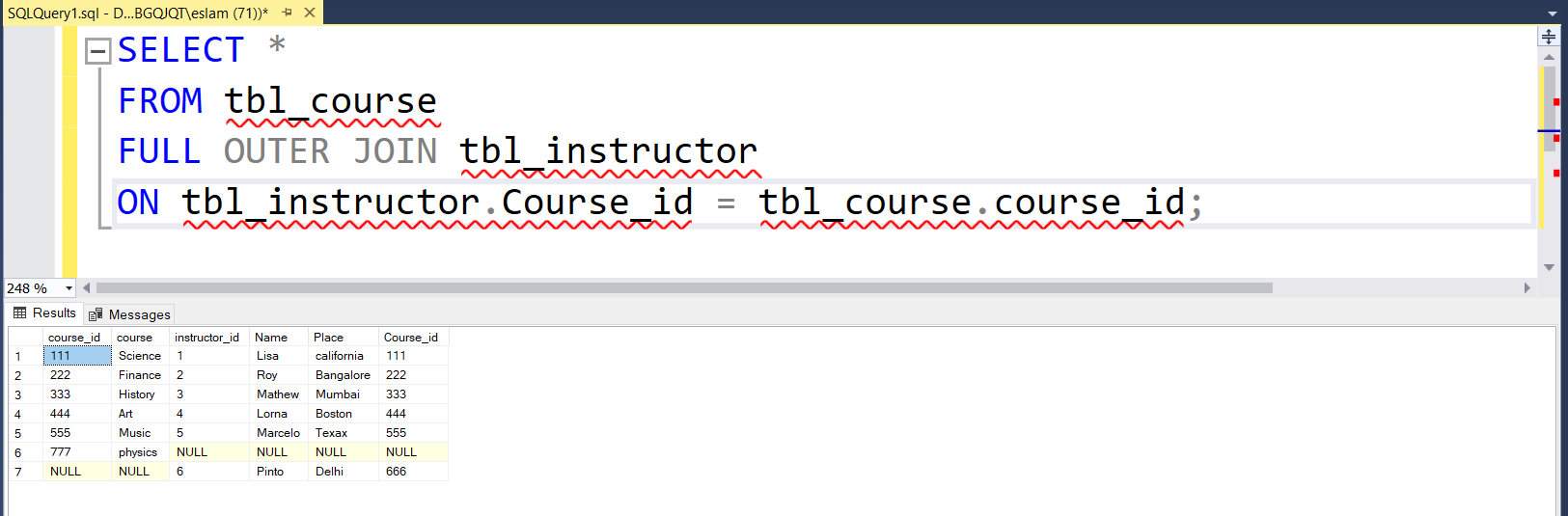
**Insert the Values**



Create **instructor** table

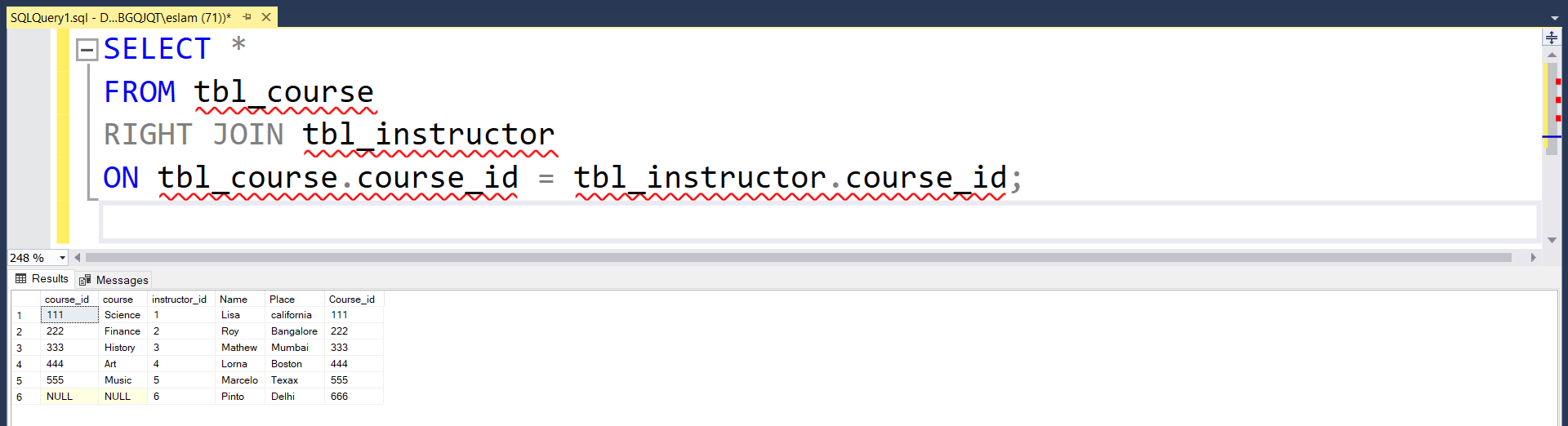
**Inserting the values**

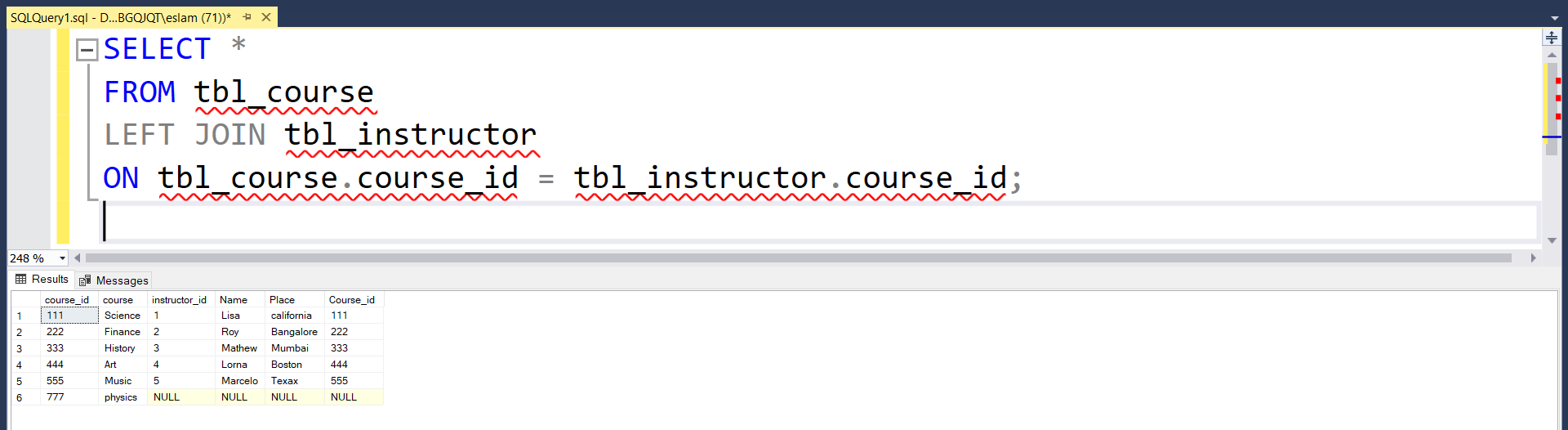
****

**Full Outer Join**

3- **Write** MySQL query to perform Left join, Right join on the above tables

**Right Join**

****

**Left Join**

**4- The following query is producing an error. Identify the error and also create the correct core SQL query:**

a) GRANT TO peter@localhost ALL PRIVILEGES on \*.\*;

Error: <> privileges should have come before the user.

GRANT ALL PRIVILEGES on \*.\* peter;

b) CREATE VIEW AS SELECT name, age FROM CUSTOMERS;

**Error:** <syntax error>the view name wasn’t exist

CREATE VIEW VIEW\_NAME AS SELECT name, age FROM CUSTOMERS;

c) CREATE USER IDENTIFIED BY 'password' AHMED;

**Error:** <syntax error> the user name should be after USER

In MySQL: CREATE USER AHMED IDENTIFIED BY 'password';

5- **Describe** the SQL commands which are used to:

1. Create a virtual table based on the result-set of an SQL statement.

**Command:** CREATE VIEW VIEW\_NAME AS SELECT name, age FROM CUSTOMERS;

1. Enables you to improve the faster retrieval of records on a database table. It creates an entry for each value of the indexed columns

**Command:** CREATE INDEX <index\_name> ON <table\_name> (<column\_name(s)>)

1. A special type of stored procedure that is invoked automatically in response to an event.

**Command:** CREATE TRIGGER <trigger\_name> <trigger\_event> ON <table\_name> FOR EACH ROW <trigger\_action>

1. A collection of pre-compiled SQL statements stored inside the database. It always contains a name, parameter lists, and SQL statements

**Command:** CREATE PROCEDURE <procedure\_name> (<parameter\_list>)

1. It is used to combine rows from two or more tables, based on a related column between them

**Command:** JOIN (various types like INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN)

1. It is used to specify the rule that allows or restricts what values/data will be stored in the table.

**Command:** CREATE TABLE table\_name (column\_name data\_type [, column\_name data\_type]..., [CONSTRAINT constraint\_name CHECK (condition)]) (Constraints can define rules like NOT NULL, PRIMARY KEY, FOREIGN KEY etc.)

**Task 2**

**6- Explain** NoSQL database and its characteristics

NoSQL databases are a type of database management system (DBMS) that diverge from traditional relational databases (SQL) in how they store and retrieve data.

NoSQL Characteristics

* Non-relational
* Flexible schema
* Other or additional query languages than SQL
* Distributed – horizontal scaling (Scaling Out)
* Less structured data
* Supports big data

7- Describe CAP theorem

GIVEN:

* Many nodes
* Nodes contain replicas of partitions of the data

• Consistency

* All replicas contain the same version of data
* Client always has the same view of the data (no matter what node)

• Availability

* System remains operational on failing nodes
* All clients can always read and write

• Partition tolerance

* multiple entry points
* System remains operational on system split (communication malfunction)
* System works well across physical network partitions

8- Compare between SQL database and NoSQL database.

|  |  |
| --- | --- |
| SQL | NoSQL |
| Relational Databases (RDBMS) | Non-relational or distributed database |
| Vertically scalable | Horizontally scalable |
| Table based databases | Document based, key-value pairs, graph databases or wide- column stores. |
| Supports predefined schema | Supports dynamic schema |
| SQL (structured query language) for defining and manipulating the data | Uses unstructured Query Language |
| Standard interface for executing complex query | Not good for executing complex query |
| Best suited for huge load and complex transactional applications | Not suited for huge load and complex transactional type applications |
| SQL databases maintains on ACID properties Atomicity, Consistency, Isolation and Durability) | NoSQL database follows the Brewers CAP theorem/BASE properties |
| Synchronous Inserts & Updates | Asynchronous Inserts & Updates |

9- **Critically evaluate** NoSQL database types

1. **Document Stores:** These databases store data in flexible, JSON-like documents, allowing for dynamic schemas and nested data structures. Examples include MongoDB and CouchDB.
2. **Key-Value Stores:** Key-value databases store data as a collection of key-value pairs, offering high-speed access and simple data models. Examples include Redis, Amazon DynamoDB, and Riak.
3. **Column-Family Stores:** Also known as wide-column stores, these databases organize data into columns rather than rows, allowing for efficient storage and retrieval of large volumes of data. Examples include Apache Cassandra and HBase.
4. **Graph Databases:** Graph databases are designed for storing and querying highly connected data, making them ideal for applications with complex relationships. Examples include Neo4j, Amazon Neptune, and TigerGraph.