

**Computer Science (CSC)**

**CSC 511:** Database Management Systems Design

**LECTURER:** DR. Dara Jaiyeola-Ajayi (Dr. Dee)

**PROJECT TITLE:** Luxury-Oriented Scenic Tours​ (LOST)

**STATUS UPDATE DATE:** 6th December 2022

**GROUP MEMBERS:**

Niharika Cherukuri (W10154863)​

Jaya Deepthi Gunnam (W10154023)​

Babavali Kotcherla (W10155395)​

Rahul Reddy Ramasagaram (W10147971)​

Ritwik NA (W10129444)

**Description of problem/requirements**

Luxury-Oriented Scenic Tours (LOST) provides guided tours to groups of visitors to their place of liking keeping their feasibility in account. In recent years, LOST has grown quickly and is having difficulty keeping up with all of the various information needs of the company. LOST offers many different tours. Guides take a test to be qualified to lead specific tours. LOST maintains numerous relations that consist of tours, guides, locations, qualifications, outings, clients, booking information, and tour visits. To maintain all of this information, LOST requires a database to store. The below is the solution that we planned.

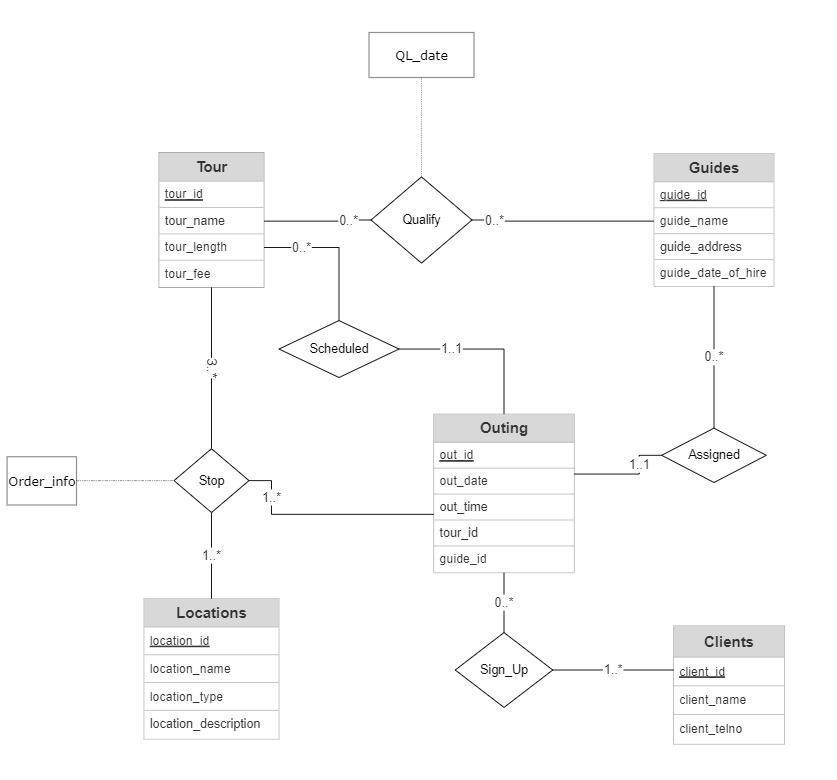
**Entities, Attributes, Primary Keys and Foreign keys for LOST Operations are mentioned below table.**

| **Entities** | **Attributes** | **Primary Keys** | **Foreign Keys** |
| --- | --- | --- | --- |
| Tour | tour\_id, tour\_name,tour\_length, tour\_fee | tour\_id | - |
| Locations | location\_id, location\_name,location\_type,location\_des | location\_id | - |
| Clients | client\_id, client\_name, client\_telno. | client\_id | - |
| Guides | guide\_id, guide\_name, guide\_address, guide\_date\_of\_hire | guide\_id |  |
| Stop | tour\_id, loc\_id, order\_info | tour\_id, location\_id | tour\_id, location\_id |
| Qualify | tour\_id, guide\_id, ql\_date | tour\_id, g\_id | tour\_id, guide\_id |
| Outing | out\_id, out\_date, out\_time, tour\_id, guide\_id | tour\_id | tour\_id, guide\_id |
| Signup | out\_id, client\_id | out\_id,client\_id | out\_id,client\_id |

**ER - Diagram description**

* The relationships between individuals, things, locations, concepts, or events in an information technology (IT) system are graphically represented in an entity relationship diagram (ERD), often referred to as an entity relationship model.
* An ER diagram displays the relationships between tables and their attributes to depict the entire logical structure of a database, where an entity is a table or an attribute of a table in a database.​
* Different ER diagrams will express cardinality in various ways. Additionally, they will vary in how entities and their characteristics are represented, whether or not relationships or attributes are depicted as independent symbols, and how entities and their properties are shown.​
* Entities: A real-world object, such as a person, location, or even a notion, qualifies.
* Attributes: An object that has what is known as an attribute, which is a real-world property.
* Relationship: Relationship explains the connection between two attributes.
* Primary Key: It is a unique column (or group of columns) defined in relational database tables to uniquely identify each table entry.
* Foreign Key: It is a field (or group of fields) that points to the PRIMARY KEY in another table.

**ER-Diagram:**



Lost offers many different tours. For each tour, we need tour name, length, and fee. Let’s consider tour\_id which acts as the primary key for a tour entity. Tour needs Guides for it to take place. So taking guides as an entity which include Guide name, Guide address, Guide date of hire and guide\_id. guide\_id works as the identification number for the Guide which is also the primary key for the Guide entity. Tour needs qualified Guides so they are connected by the weak entity “Qualify” which has attributes tour\_id and guide\_id which acts as both primary key and foreign key for the entity “Qualify”. It also includes the QL\_date attribute which states the qualification date of the Guide. Guides need to be assigned for Outing. So, guides and outings are connected with the weak entity “Assigned”. Outing entity includes out\_id which works as a primary key which helps guides to find other attributes of outing which are out\_date, out\_time, and it also includes tour\_id and guide\_id which works as foreign key in the “Outing” entity. When clients call for the information about an outing they need to register to get an outing so they are connected by the weak entity “sign\_Up”. “Clients” entity include attributes client\_id, client\_name and client\_telno where client\_id is the primary key. Outings have stops for location. So, “Locations” entity and “Outings” entity are connected with the “stop” weak entity with an attribute “Order\_info”. “Locations” entity includes attribute location\_id, location\_type and location\_description in which location\_id works as the primary key.

**Schema Diagram:**

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organized using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.

To create a Schema Diagram that supports the LOST operation, you need to see the following tables. Here bold figures represent primary keys.

tour (**tour\_id**, tour\_name, tour\_length, tour\_fee)

In this table, tour\_id is a primary key, tour\_name is used to represent tour name, tour\_approx\_length is calculated in terms of hours and tour\_fee represents fee charge which is paid by clients.

Guide (**guide\_id, tour\_id**, guide\_name, guide\_address, guide\_doh, guide\_qualification)

In this table guide\_id is the primary key, which is assigned to each guide, guide\_name represents guide name, guide\_address represent guide home address, guide\_doh represent guide date-of-hire and guide\_qualification represent the qualification of guide whether guide is qualified for a particular tour or not.

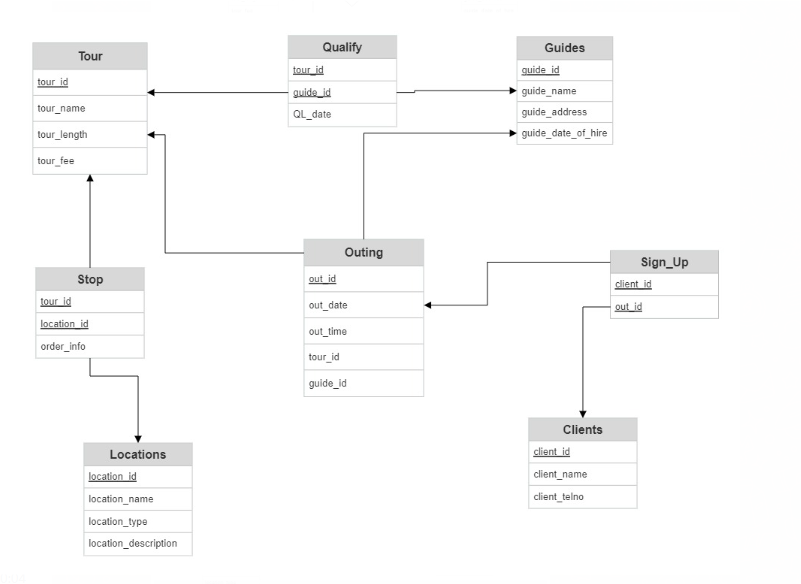
Location (**location\_name, tour\_id**, location\_type, location\_description)

In this table location\_name represents location name and is used as a primary key, location\_type represent type of location and location\_description represent official description about the location.

Outing (**out\_id, tour\_id, guide\_id**, out\_date, out\_time)

In this table, out\_date, out\_time represent scheduled date and time.

client (**client\_id, out\_id,** client\_name, client\_telno)



**Summary of Implementation.**

We started with the ER and Schema diagrams. Took some time to finalize the diagrams and later on started with the implementation. We have chosen Azure SQL as our Database. Before creating the table, we made sure to ‘Drop’ the tables if there are any previous tables with the same name. Based on the Schema diagram we started writing and executing our queries in the Azure Database. When we faced the errors while running the queries improvised them.

**Azure SQL:**

* Azure SQL Database is a NoSQL database that delivers scalable, dependable, and quick data access. It gives you the option of running your app on-premises or in the cloud, as well as the scalability to handle massive amounts of data.
* Azure SQL Database is constantly running on the most recent stable version of the SQL Server database engine and a patched operating system, with 99.99% uptime. Azure SQL Database allows you to concentrate on the domain-specific database administration and optimization tasks that are important to your organization.

**Screenshots of the SQL implementation**

1. **We have created the SQL database:**

**Graphical user interface, application

Description automatically generated**

1. **Created the Table named ‘Tour’ to store all the tour related data with the entities tour id, tour name, tour length and tour fee.**

**Here the tour id is the primary key.**

**The table is created successfully.**



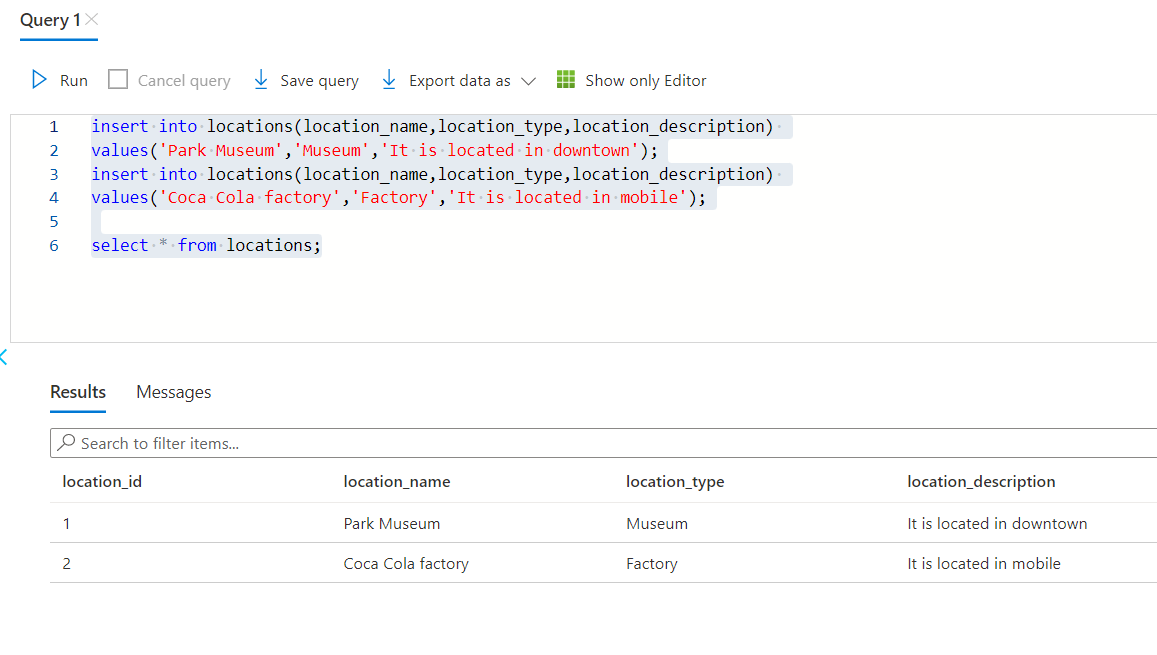
1. **Here we are inserting the values into the ‘Tour’ table:**



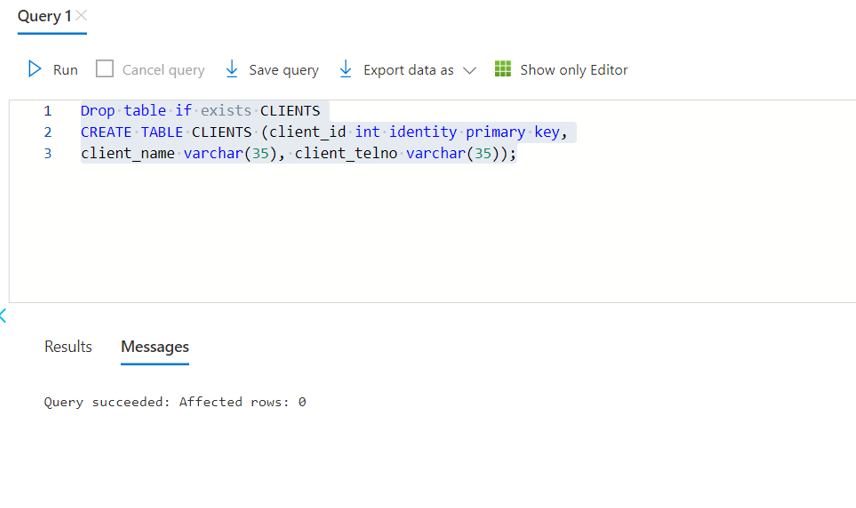
1. **Created the table named ‘Locations’ to store all the location details with attributes location id, location name, location type, location description. Here the location id is the primary key:**



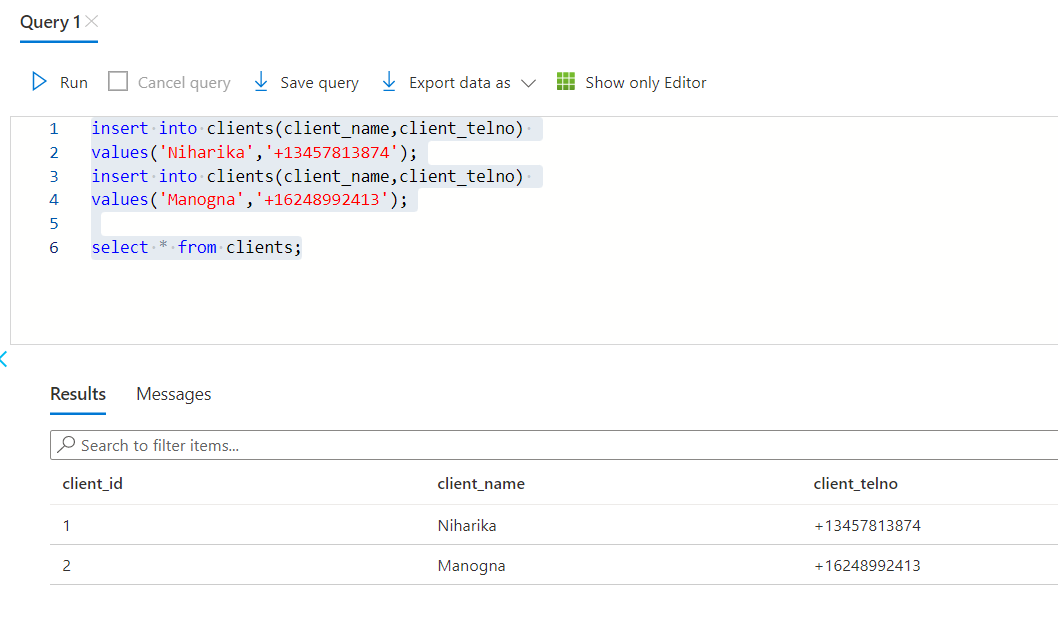
1. **Later, inserted the values into the locations table**:



1. **Created the ‘clients’ table to store all the client related information with the attributes client id, client name, client telno, where the client id is the primary key:**



1. **Inserted the values into the ‘clients’ table**:



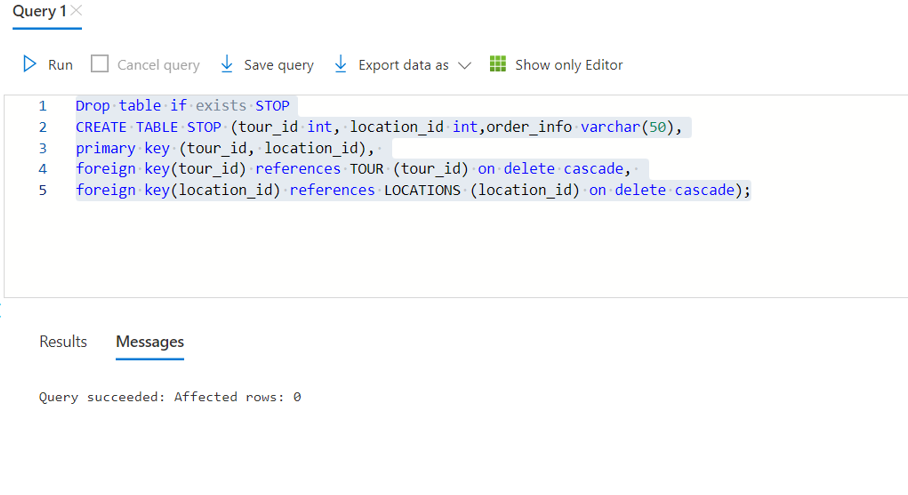
1. **Created the ‘guide’ table to store the guide related data with the attributes guide name, guide address, guide’s date of hire where the guide id is the primary key:**



1. **Inserted the values into the guide table:**



1. **Created the ‘stop’ table to store all the stops that are going to be in the tour. The attributes are tour id, location id, order info. Here the tour id serves as both Primary and foreign key. And the location id serves as the foreign key:**



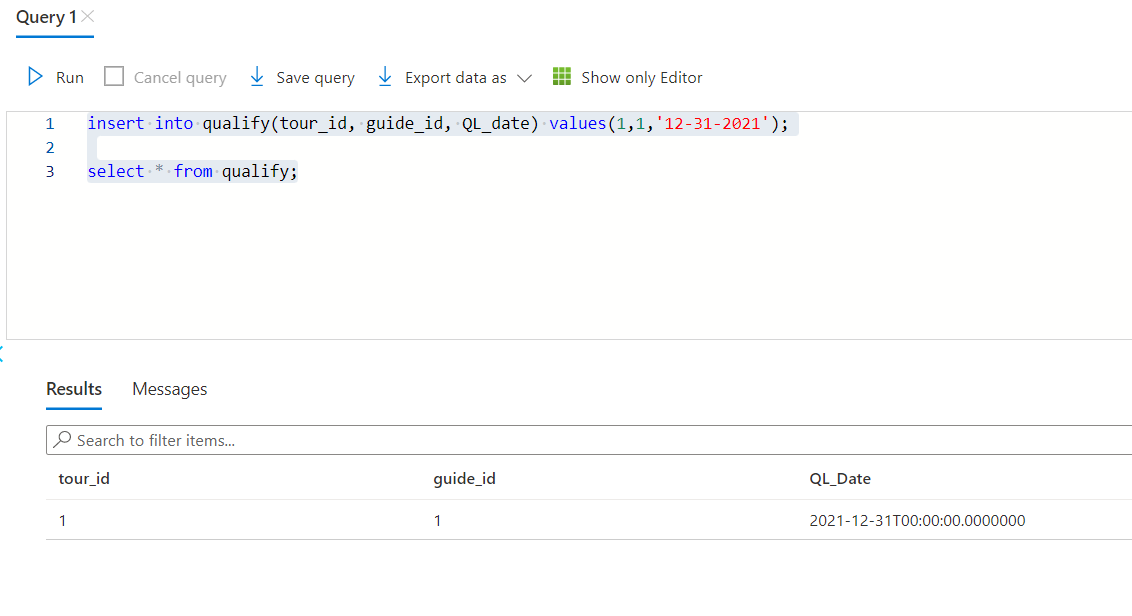
1. **Here we are inserting the values into the ‘stop’ table:**



1. **Created a table named ‘qualify’ with the attributes tour id, guide id and Qualified date. Here the tour id and guide id acts as both primary and foreign key:**



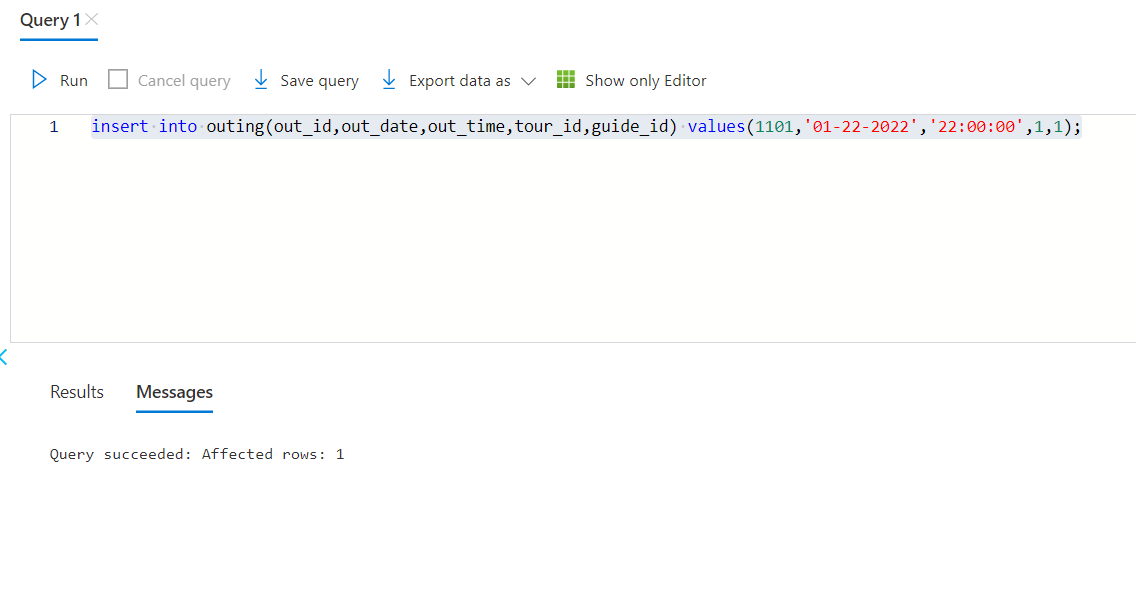
1. **Here we are inserting the values into the ‘qualify’ table:**



1. **Created a table named ‘outing’ with the outing id and outing date as the attributes. Here the outing id is the primary key and tour id and guide id are the foreign keys:**

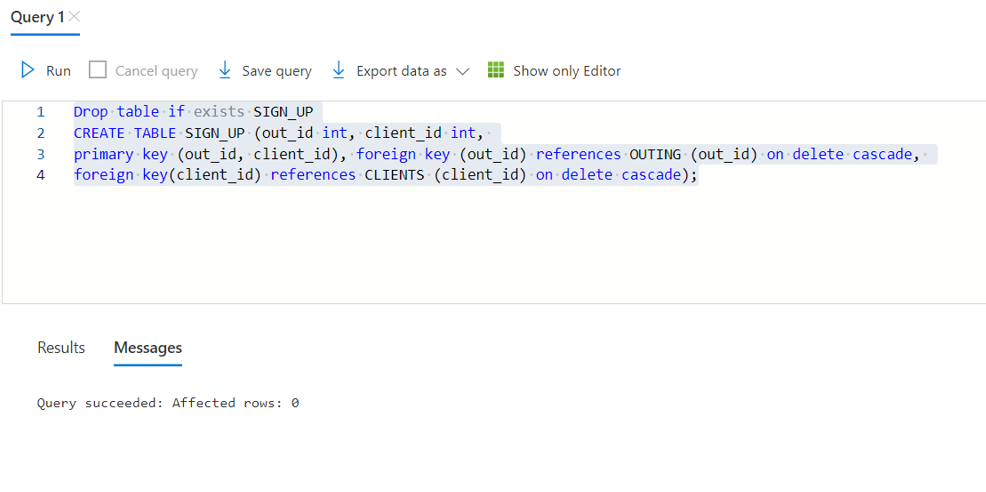
****

1. **Inserted the values into outing table:**

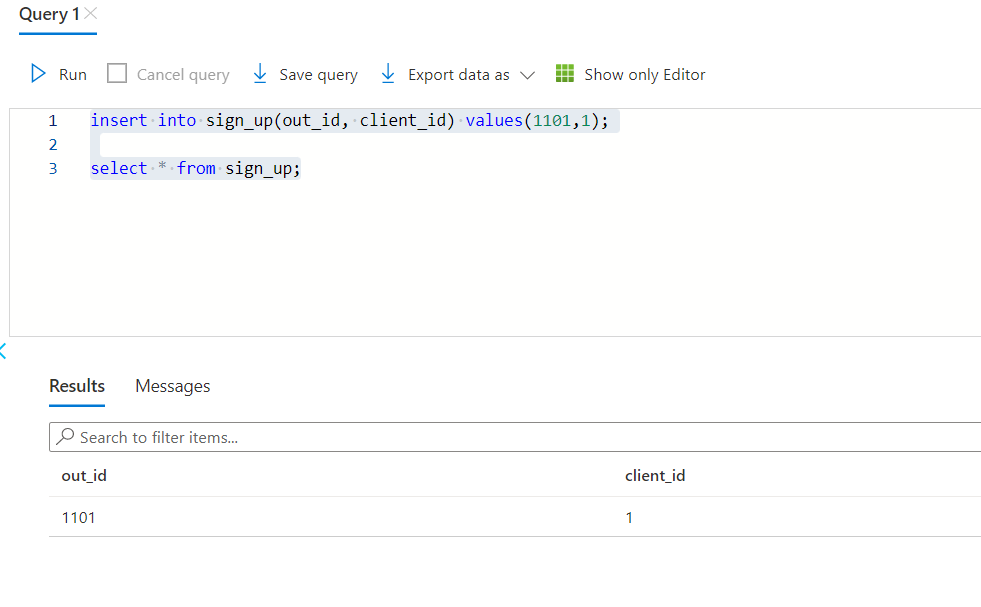


1. **Created the ‘sign\_up’ table with the outing id and the client id. Here both the outing**

**id and the client id acts as both the primary and foreign key:**



1. **Inserted the values into the sign\_up table:**



**Discussion Section:**

We had a discussion on Schema diagrams and ER- Diagrams. While implementing them for the first time we faced issues with the foreign key and our queries were not running properly. And later we identified the issue, resolved and implemented our queries in the Azure SQL database.

**Conclusion:**

The ‘LOST’ offers many different tours. We have given our solution for the LOST based on the requirements. We had multiple opinions and solutions but as a team we came up with the best solution. All the queries were running successfully.

**Reference:**

* **Microsoft Azure:** <https://azure.microsoft.com/en-us/>
* **ER diagram:** <https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model>
* **Schema diagram:** <https://www.lucidchart.com/pages/database-diagram/database-schema>
* **SmartDraw:** https://www.smartdraw.com/?id=104640&gclid=Cj0KCQiA7bucBhCeARIsAIOwr--tIgfdkqj2TPagzQMw9bDiDmrDe-cYl1BJPM308FphglpG9Uw7ICgaAsQAEALw\_wcB