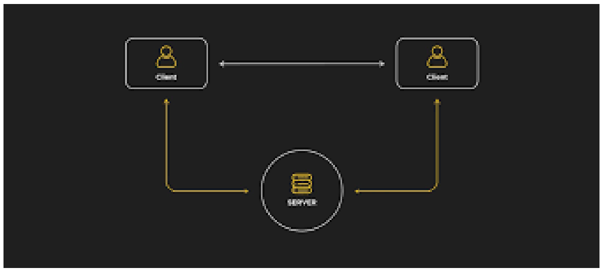
INTRODUCTION TO SQL



What is our GOAL for this CLASS?

In this class, we learnt about differences b/w SQL and NoSQL databases, learnt about a few basic queries in SQL and performed a SQL Injection.

What did we ACHIEVE in the class TODAY?

- Introduction to SQL
- Differences b/w SQL and NoSQL databases
- Select query with where clause and boolean operators
- SQL Injection

Which CONCEPTS/ CODING BLOCKS did we cover today?

- SQL
- SQL Injection

How did we DO the activities?

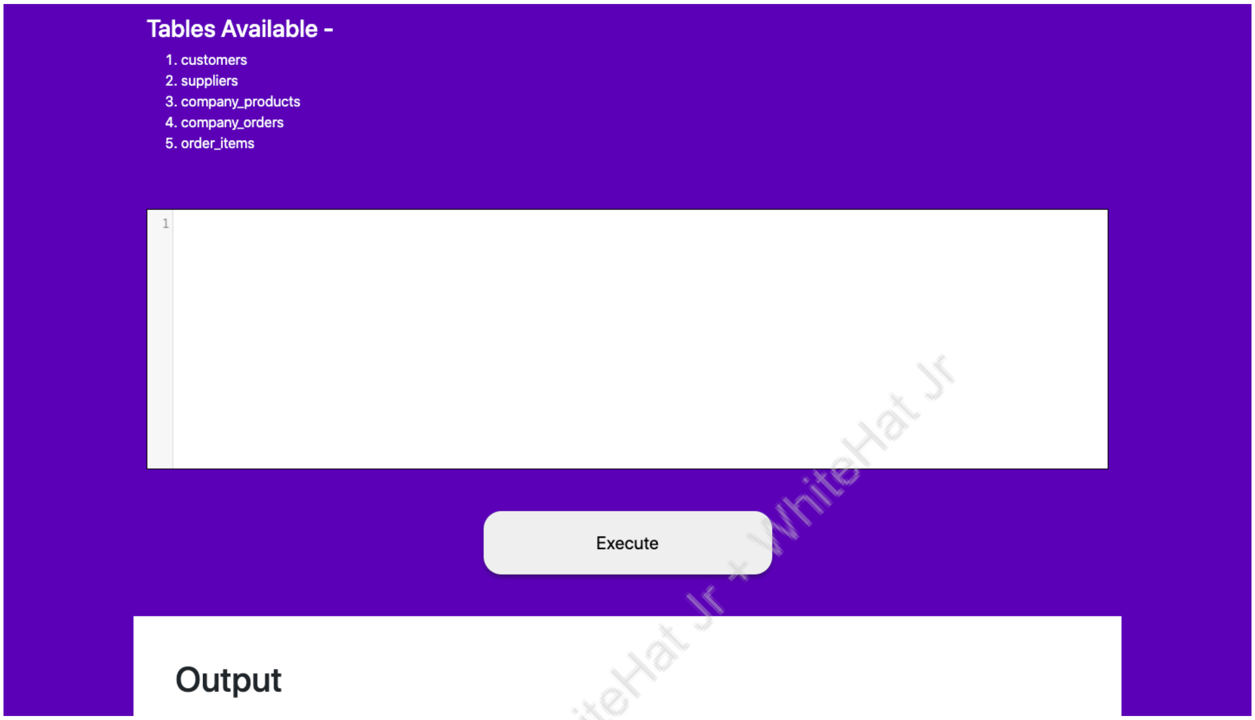
Activity:

1. We discussed about some of the key differences b/w SQL and NoSQL databases

SQL Database	NoSQL Database
SQL databases are table based. Different tables are created for different data.	NoSQL databases are JSON based. Usually it's key-value pairs.
SQL databases strictly rely on relations. This means that that data is present in separate tables with a relation between them.	NoSQL databases do not use relations.
SQL databases have structured predefined schemas. All the columns in a particular table define the data type of all the data.	NoSQL databases do not have schemas and are not structured. You can have as many key-value pairs with any data type you want to use.
SQL databases are better with multi row based structured data.	NoSQL databases are better for JSON like or unstructured data.
Example - PostgreSQL	Example - Firebase

2. We discussed **DBMS - Database Management Systems** - and why they are important. It can be used for -
    a. Creation of a database
    b. Retrieval of information from the database
    c. Updating the database
    d. Managing the database
3. SQL is a database language - **Structured Query Language.**
4. SQL databases are usually in the form of Tables, consisting of rows and columns. Columns are usually the kind of data with its type. You cannot enter an integer into a String column. This is known as the DB schema.

5. We open this link to learn SQL

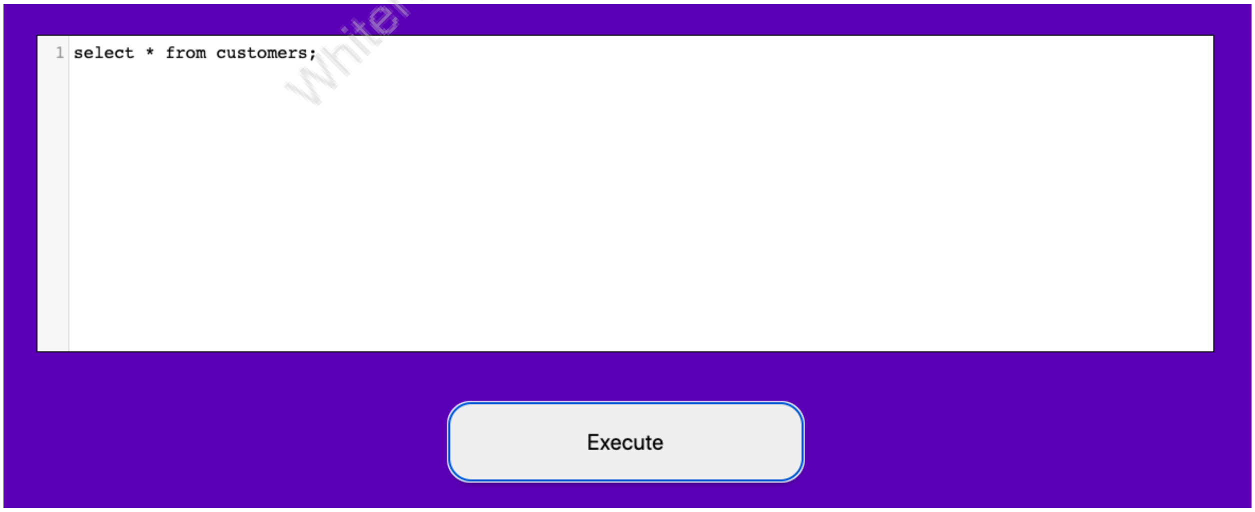


Tables Available -

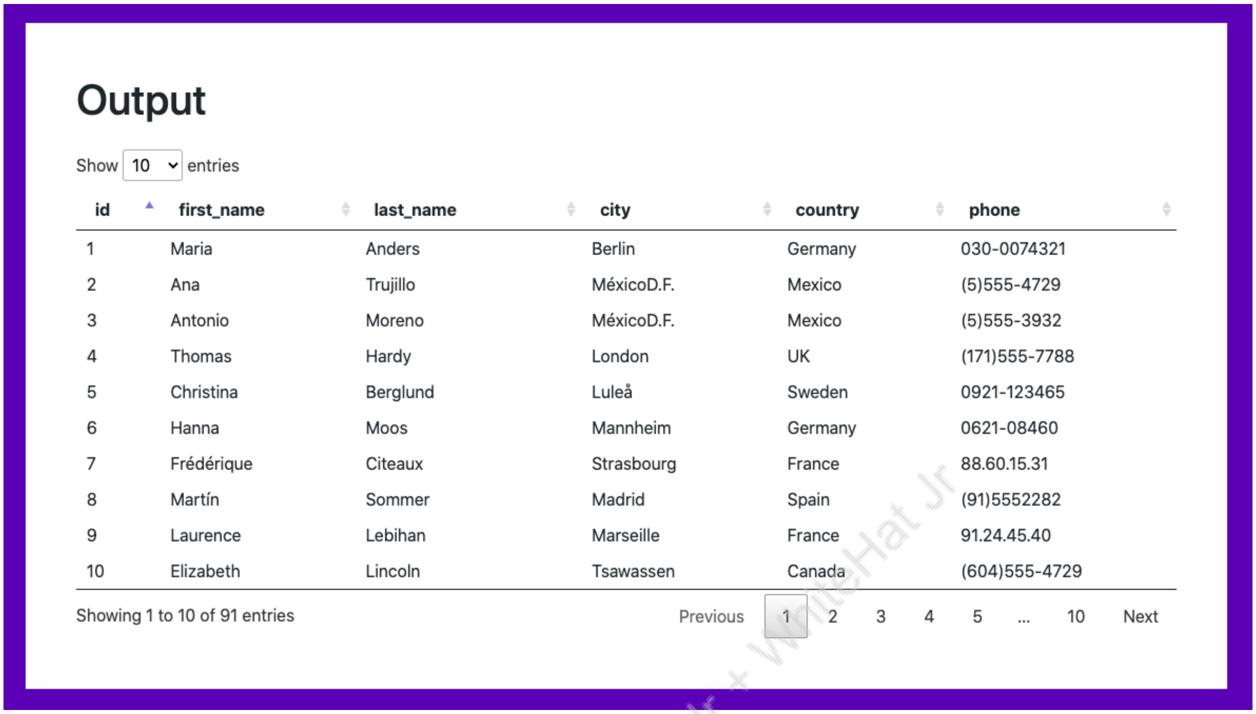
1. customers
2. suppliers
3. company_products
4. company_orders
5. order_items

6. We execute the following SQL statement -

SELECT * from customers;



```
select * from customers;
```



Output

Show 10 entries

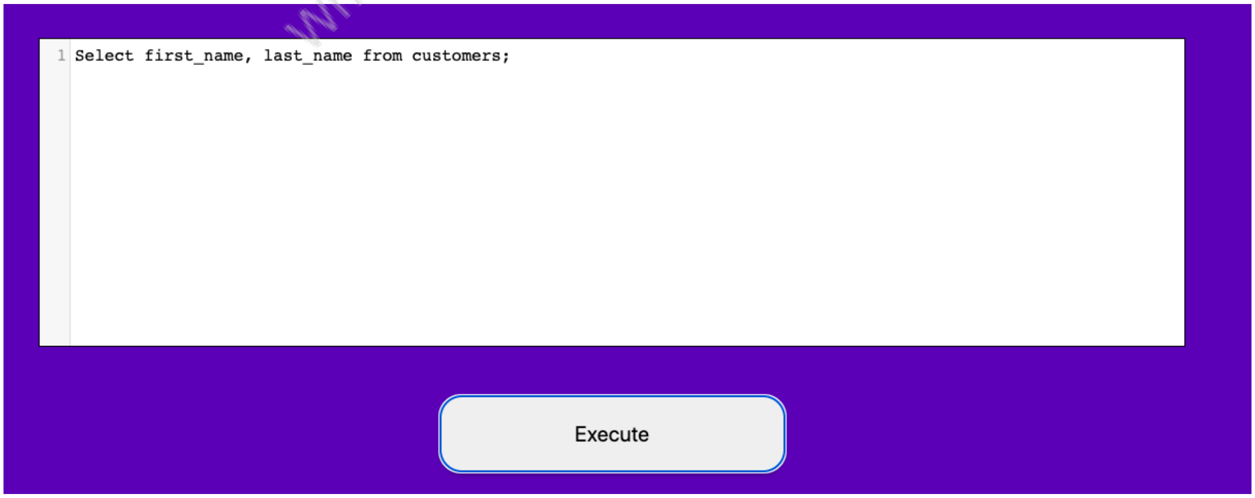
id	first_name	last_name	city	country	phone
1	Maria	Anders	Berlin	Germany	030-0074321
2	Ana	Trujillo	MéxicoD.F.	Mexico	(5)555-4729
3	Antonio	Moreno	MéxicoD.F.	Mexico	(5)555-3932
4	Thomas	Hardy	London	UK	(171)555-7788
5	Christina	Berglund	Luleå	Sweden	0921-123465
6	Hanna	Moos	Mannheim	Germany	0621-08460
7	Frédérique	Citeaux	Strasbourg	France	88.60.15.31
8	Martín	Sommer	Madrid	Spain	(91)5552282
9	Laurence	Lebihan	Marseille	France	91.24.45.40
10	Elizabeth	Lincoln	Tsawassen	Canada	(604)555-4729

Showing 1 to 10 of 91 entries

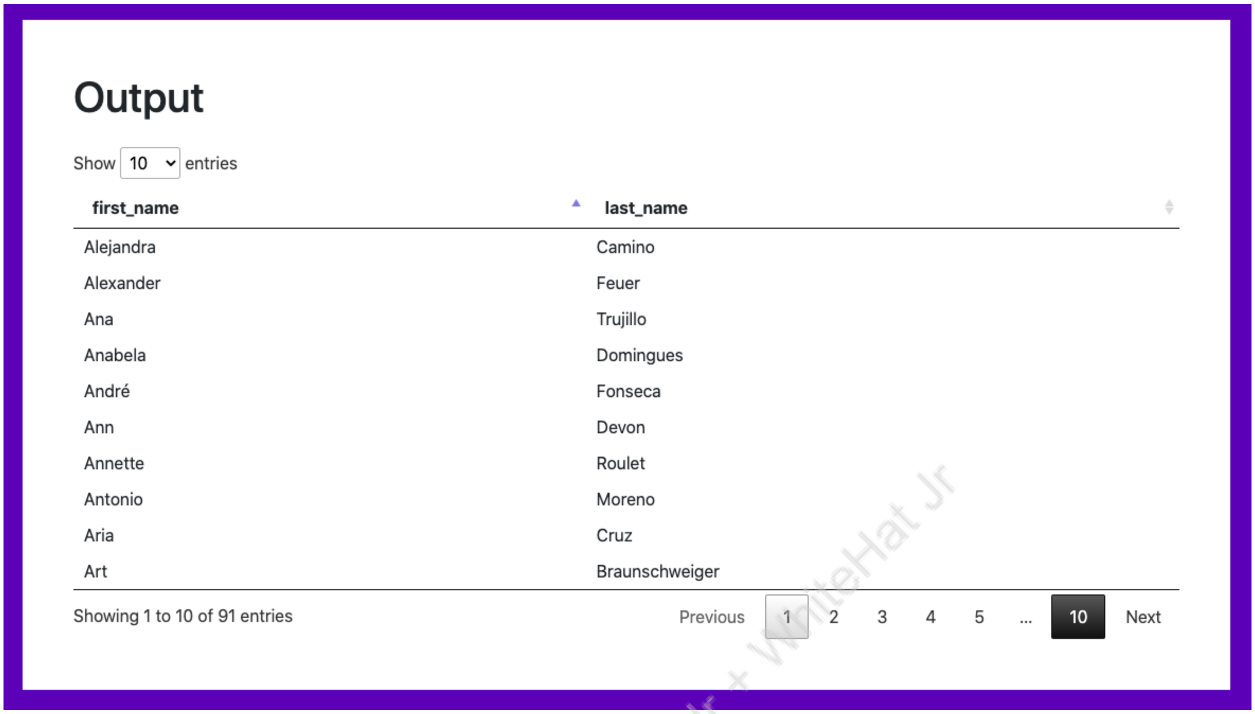
a. **Select** statement is used to select data from a table.
b. (*) asterisk is used to select all the columns
c. **From** keyword is used to name the table from which we want to select data. In the case above, it's **customers**
d. Semicolon (;) is mandatory in SQL after all the statements.

7. To select just specific columns, we can specify the column names in the query -

SELECT first_name, last_name from customers;



```
Select first_name, last_name from customers;
```



Output

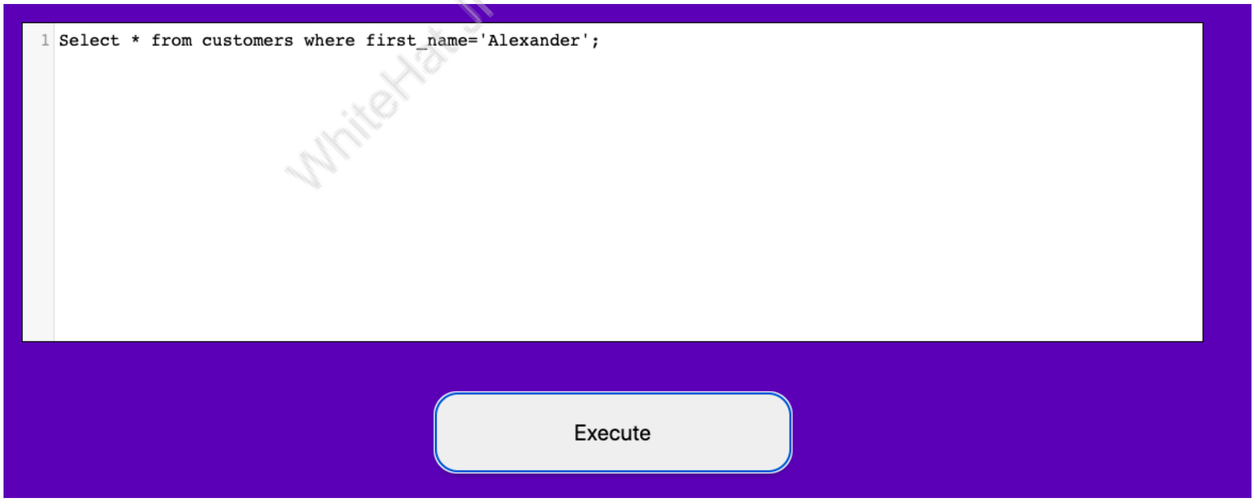
Show 10 entries

first_name	last_name
Alejandra	Camino
Alexander	Feuer
Ana	Trujillo
Anabela	Domingues
André	Fonseca
Ann	Devon
Annette	Roulet
Antonio	Moreno
Aria	Cruz
Art	Braunschweiger

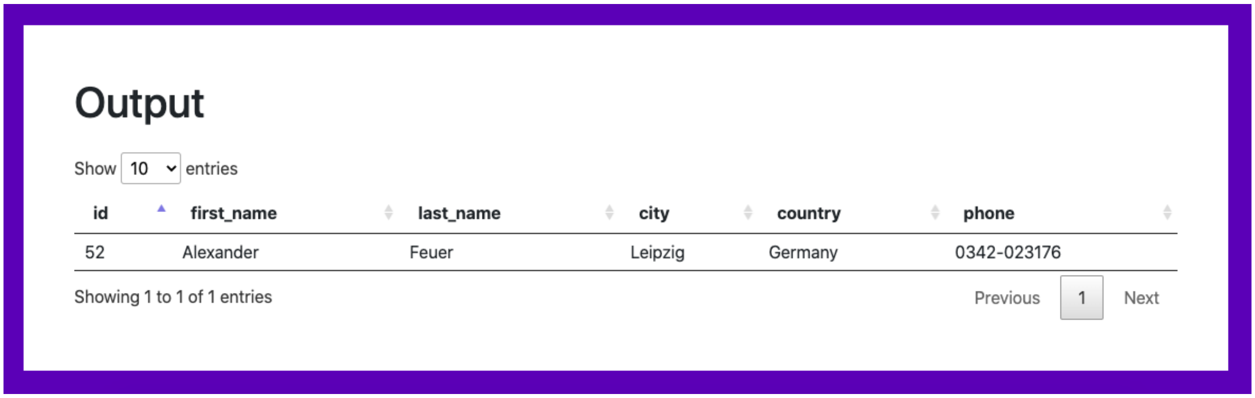
Showing 1 to 10 of 91 entries

8. To select specific rows, we use **where** clause -

SELECT * from customers where first_name='Alexander';



```
Select * from customers where first_name='Alexander';
```



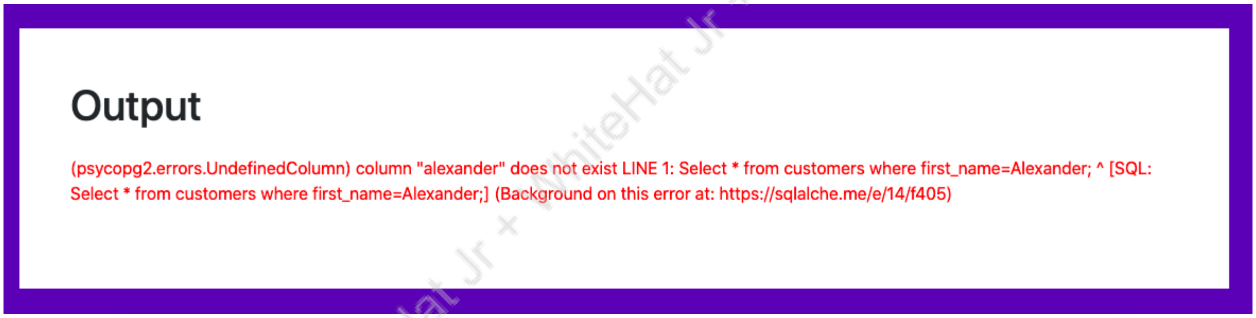
Output

Show 10 entries

id	first_name	last_name	city	country	phone
52	Alexander	Feuer	Leipzig	Germany	0342-023176

Showing 1 to 1 of 1 entries

a. String only works with single quotes ‘’ in SQL. Double quotes “” would throw an error.



Output

```
(psycopg2.errors.UndefinedColumn) column "alexander" does not exist LINE 1: Select * from customers where first_name=Alexander; ^ [SQL: Select * from customers where first_name=Alexander;] (Background on this error at: https://sqlalche.me/e/14/f405)
```

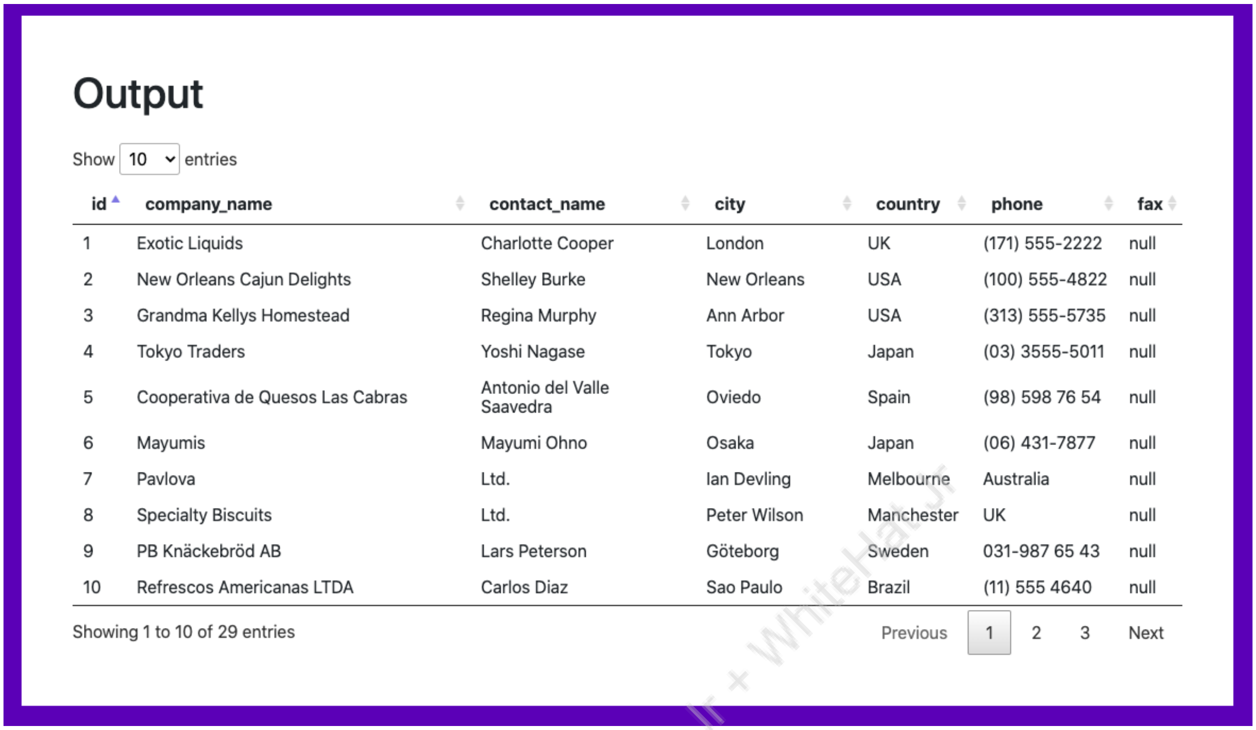
9. We can use **AND** & **OR** boolean operators in our SQL statements to set the condition with **where** clause -

```
Select * from customers where first_name='Alexander' and last_name='Feuer';

Select * from customers where first_name='Alexander' or last_name='Feuer';
```

10. We perform a student activity to query all the suppliers.

```
SELECT * from suppliers;
```



Output

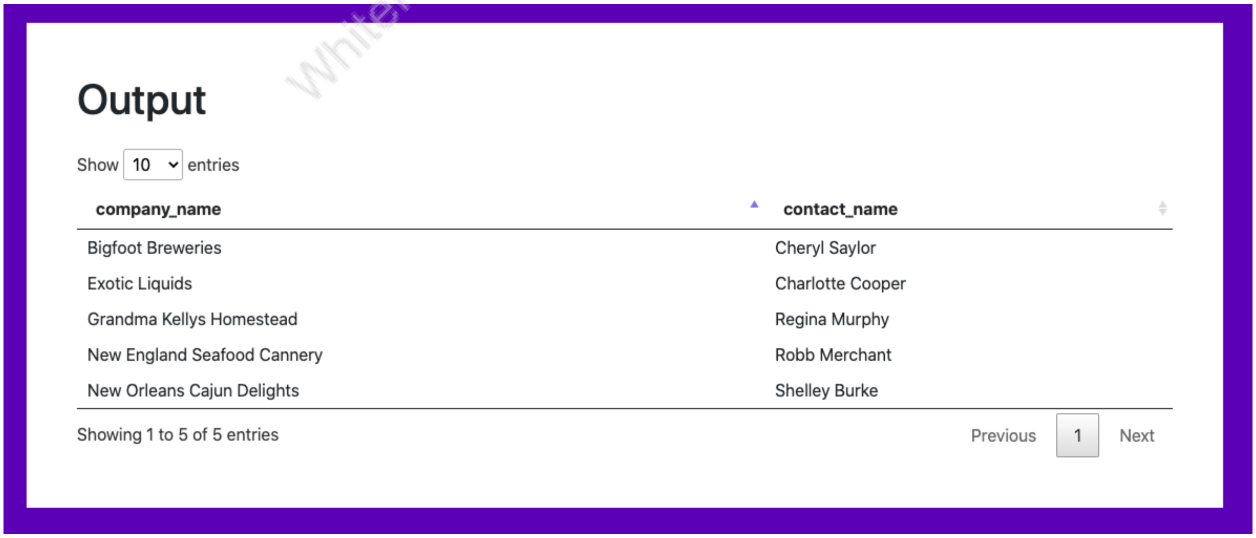
Show 10 entries

id	company_name	contact_name	city	country	phone	fax
1	Exotic Liquids	Charlotte Cooper	London	UK	(171) 555-2222	null
2	New Orleans Cajun Delights	Shelley Burke	New Orleans	USA	(100) 555-4822	null
3	Grandma Kellys Homestead	Regina Murphy	Ann Arbor	USA	(313) 555-5735	null
4	Tokyo Traders	Yoshi Nagase	Tokyo	Japan	(03) 3555-5011	null
5	Cooperativa de Quesos Las Cabras	Antonio del Valle Saavedra	Oviedo	Spain	(98) 598 76 54	null
6	Mayumis	Mayumi Ohno	Osaka	Japan	(06) 431-7877	null
7	Pavlova	Ltd.	Ian Devling	Melbourne	Australia	null
8	Specialty Biscuits	Ltd.	Peter Wilson	Manchester	UK	null
9	PB Knäckebröd AB	Lars Peterson	Göteborg	Sweden	031-987 65 43	null
10	Refrescos Americanas LTDA	Carlos Diaz	Sao Paulo	Brazil	(11) 555 4640	null

Showing 1 to 10 of 29 entries

11. We perform a student activity to query all the suppliers' company_name and contact_name based out of the USA or UK.

SELECT company_name, contact_name from suppliers where country='USA' or country='UK';



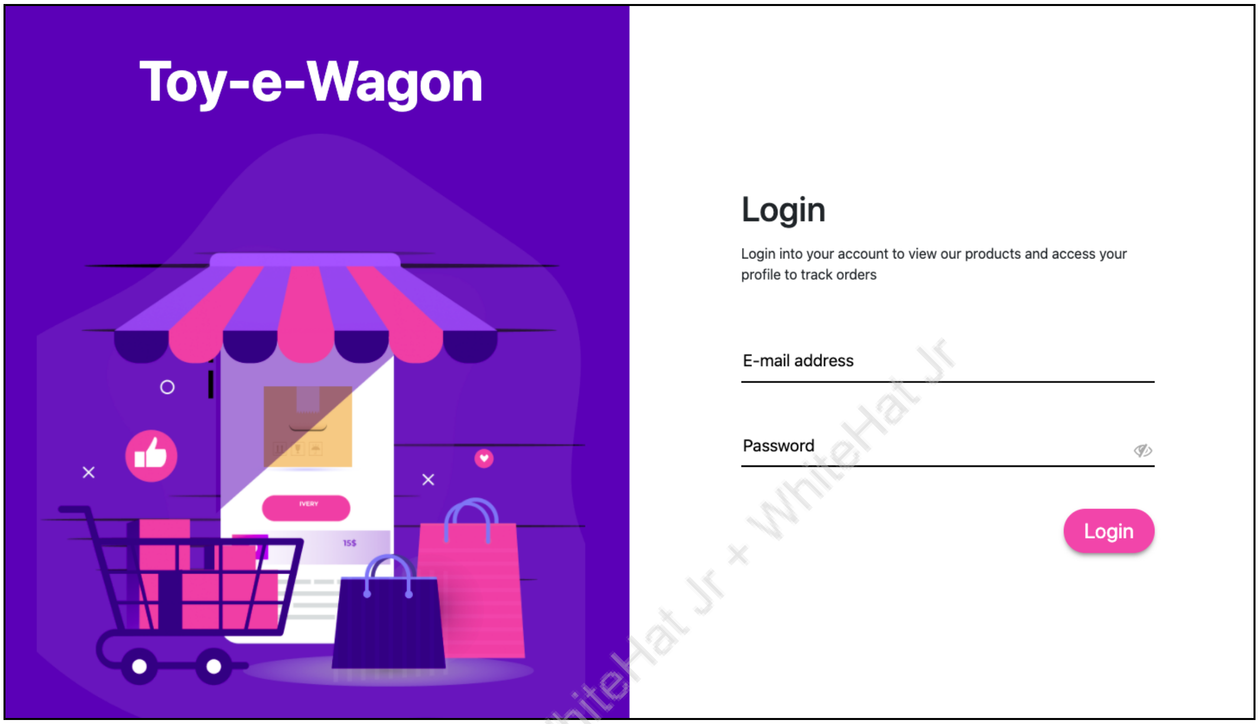
Output

Show 10 entries

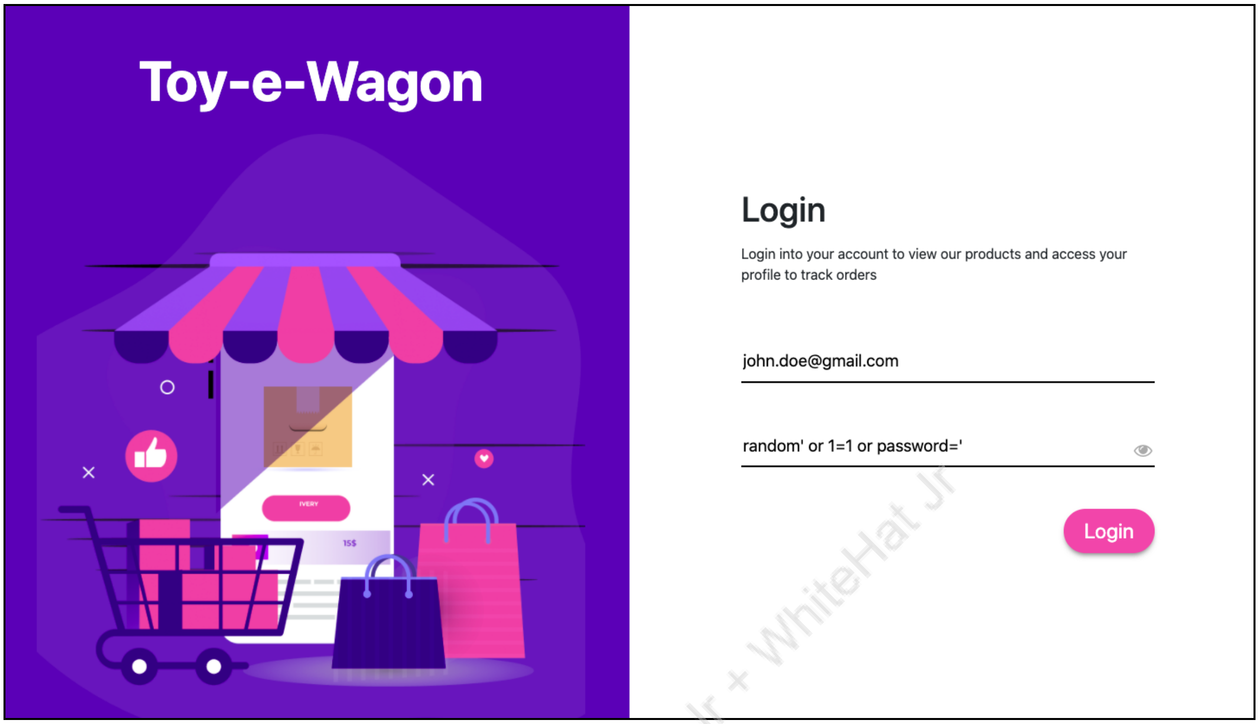
company_name	contact_name
Bigfoot Breweries	Cheryl Saylor
Exotic Liquids	Charlotte Cooper
Grandma Kellys Homestead	Regina Murphy
New England Seafood Cannery	Robb Merchant
New Orleans Cajun Delights	Shelley Burke

Showing 1 to 5 of 5 entries

12. We open this link and perform a SQL injection with the following credentials -


Toy-e-Wagon
Login
Login into your account to view our products and access your profile to track orders
E-mail address
Password
Login

a. Email - john.doe@gmail.com
b. Password - random' or 1=1 or password='
c. We understand this by thinking of the statement running in the background. Something like -
    i. Select * from users where email='{}' and password='{}'
    ii. Here, it is simply going to replace the values of email and password with what is entered by the user, and see if a user exists or not. If it does, login is successful.
    iii. With our values, this statement becomes -
        1. Select * from users where email='john.doe@gmail.com' and password='random' or 1=1 or password='';
        2. In this statement, the second part - password='random' or 1=1 or password='' becomes true because 1=1.
        3. Since the statement uses an AND statement, and we made the second part True without actually entering the right password by inserting some SQL, the statement changes and it lets you login.



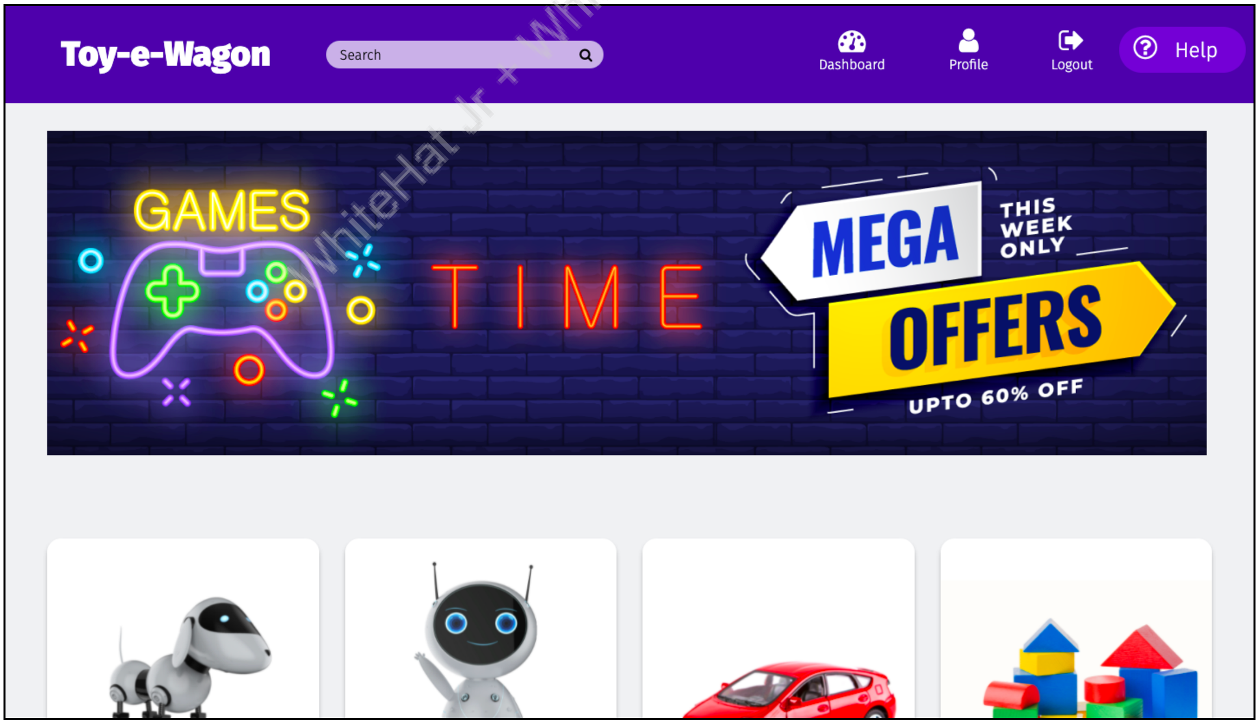
Login

Login into your account to view our products and access your profile to track orders

john.doe@gmail.com

random' or 1=1 or password='

And on login, we see the dashboard without actually entering the password -



What's NEXT?

In the next class, we will learn about join statements in SQL

Expand Your Knowledge:

Explore more about SQL v/s NoSQL databases here.