# 4. Multi-Table JOINs

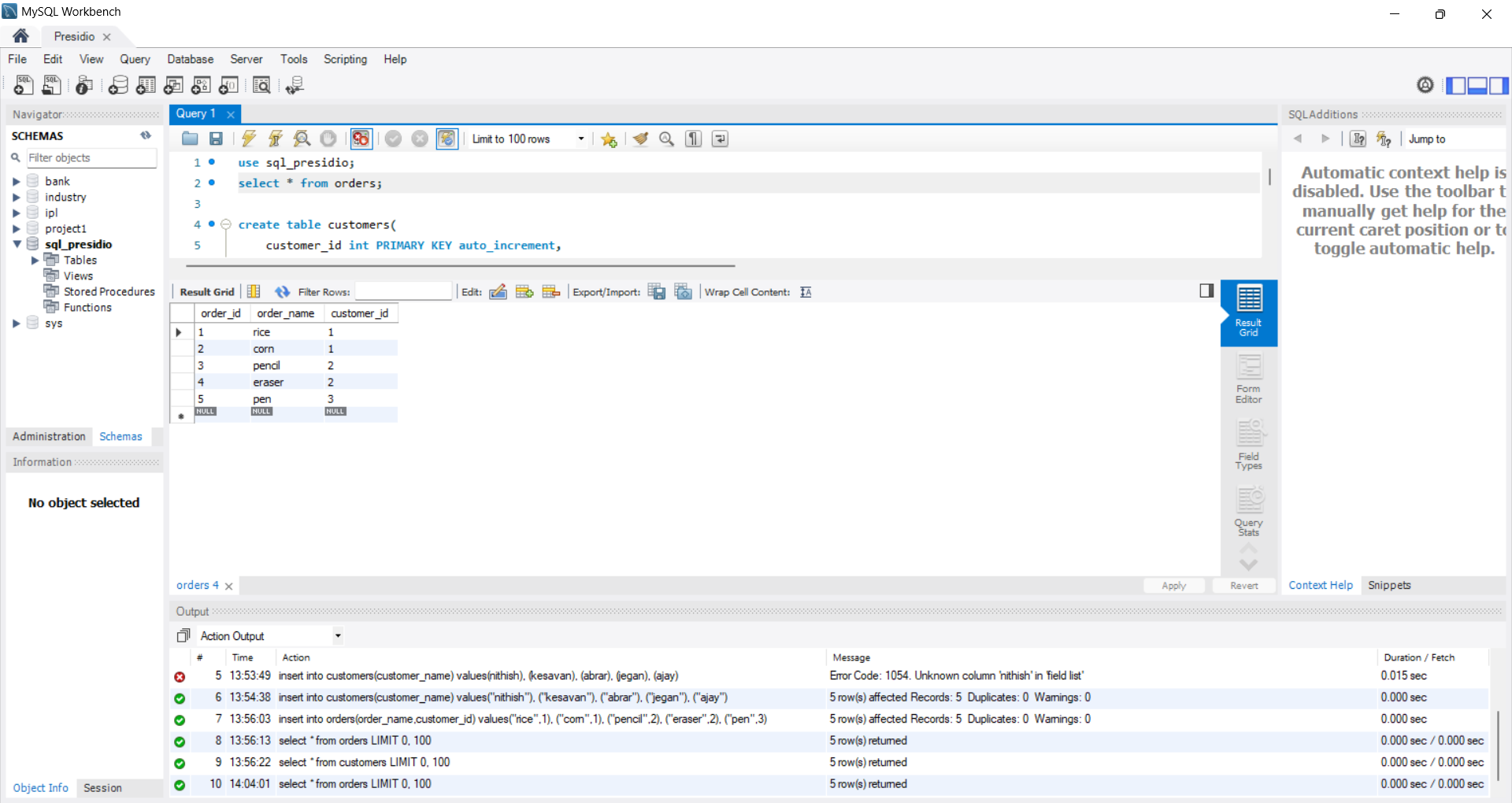
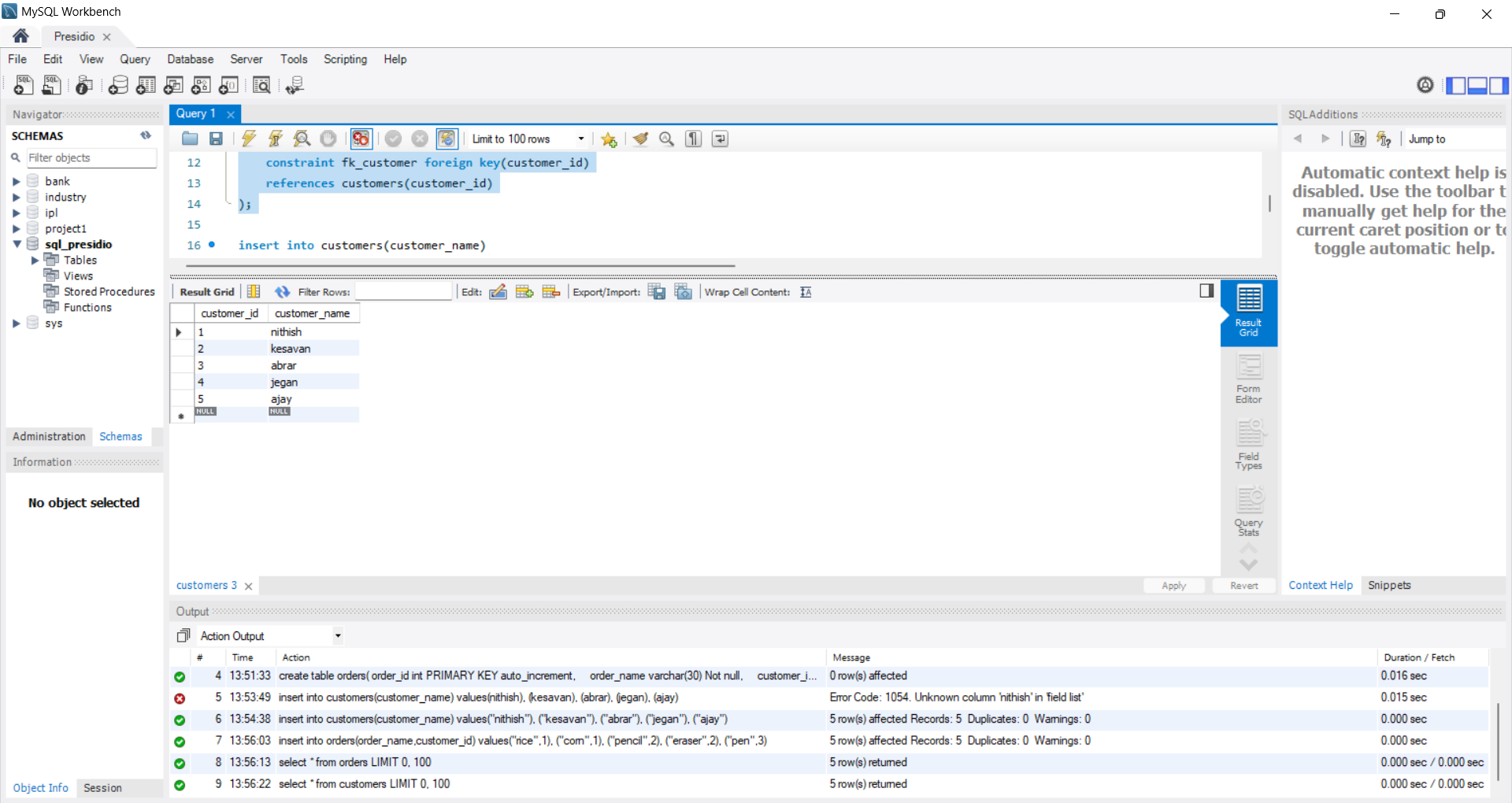
## Objective

To combine data from two related tables using JOIN operations.

## Queries

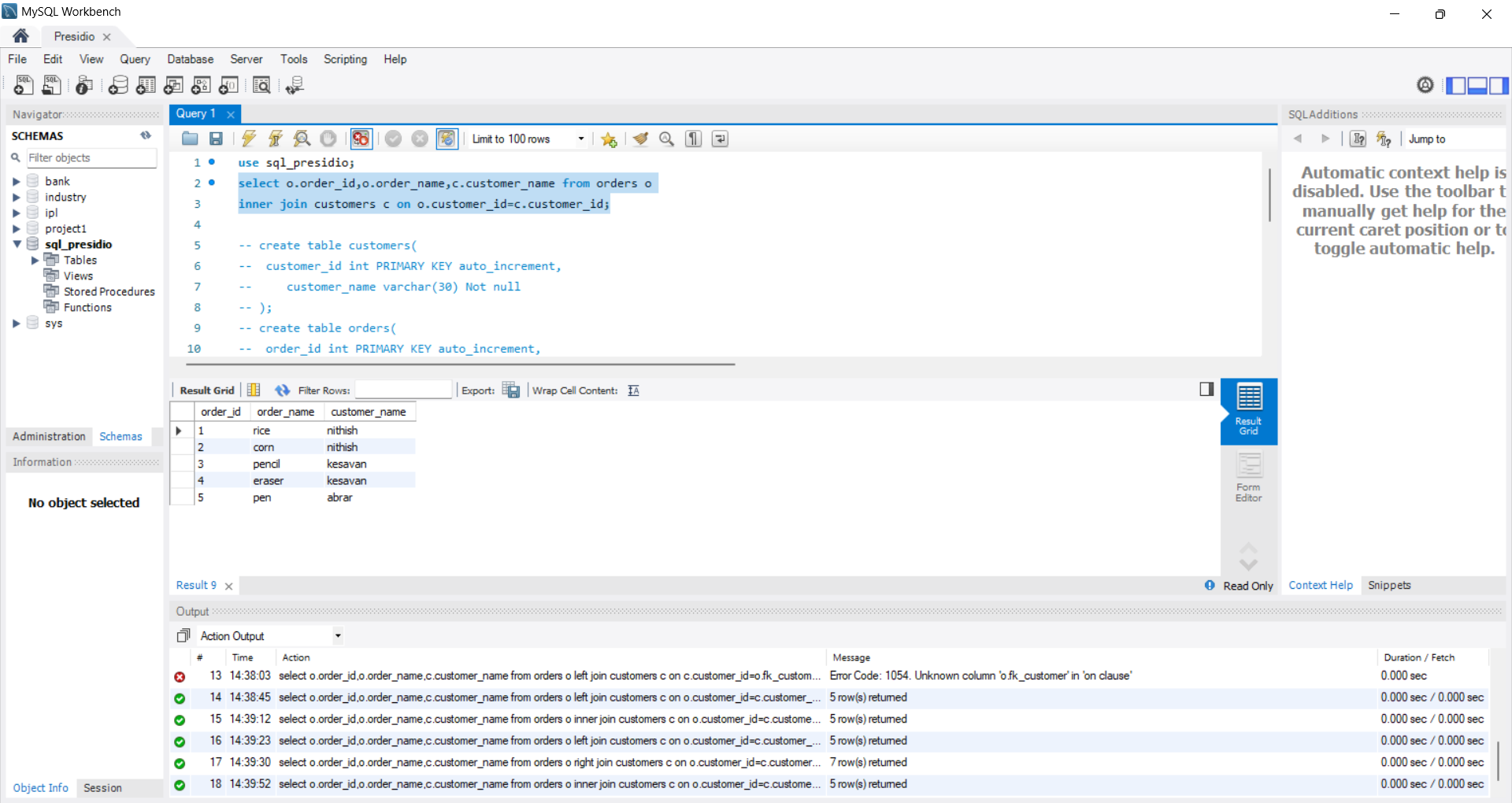
1. ***create table customers(***
2. ***customer\_id int PRIMARY KEY auto\_increment,***
3. ***customer\_name varchar(30) Not null***
4. ***);***
5. ***create table orders(***
6. ***order\_id int PRIMARY KEY auto\_increment,***
7. ***order\_name varchar(30) Not null,***
8. ***customer\_id int,***
9. ***constraint fk\_customer foreign key(customer\_id)***
10. ***references customers(customer\_id)***
11. ***);***

* Created two tables named **customers** and **orders** with related columns.
* In **orders** table, **customer\_id** is used as a **foreign key** that refers the **primary key** of the **customers** table.
* Customers – parent table, Orders – Child table.



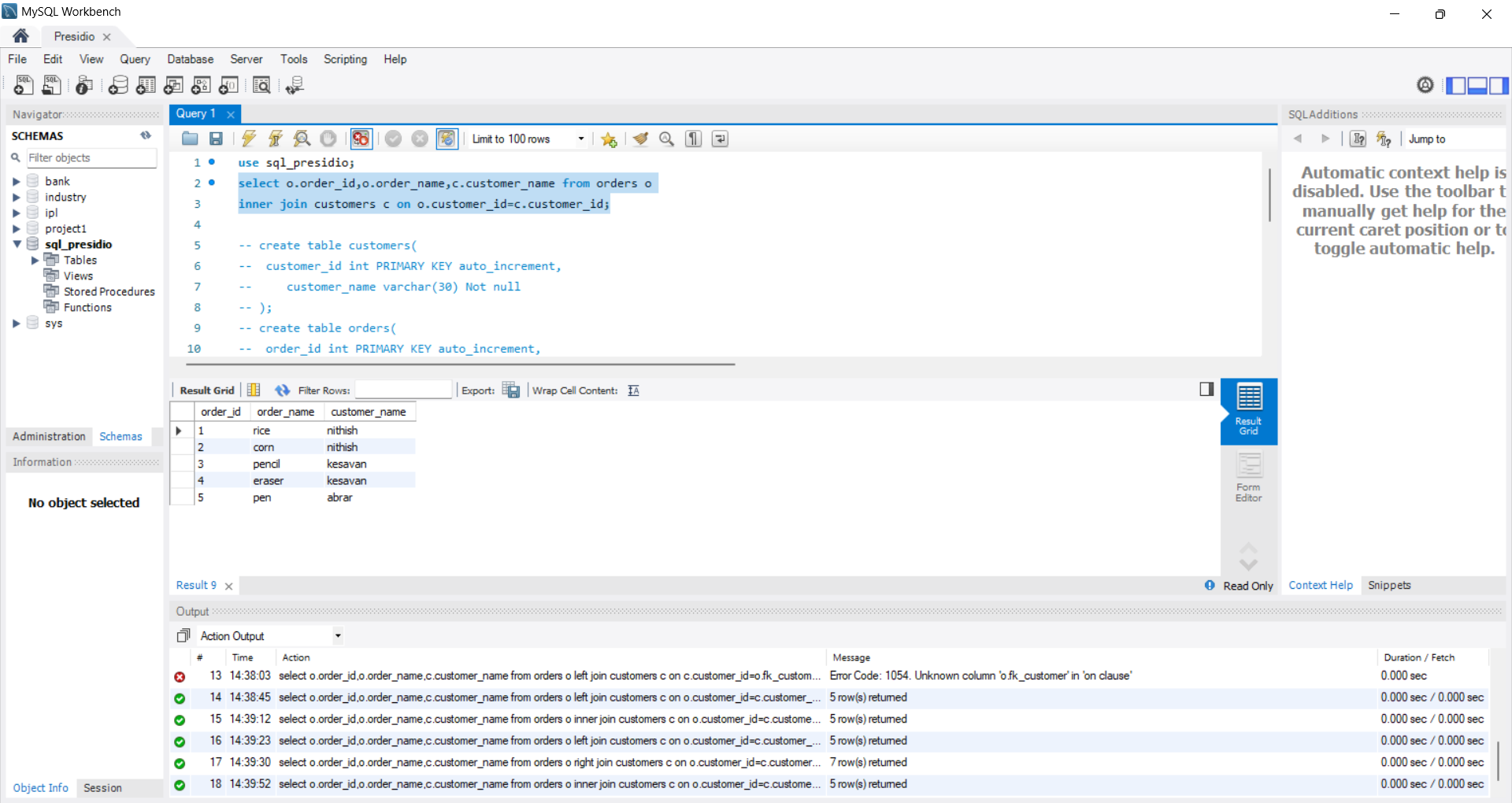
1. ***select o.order\_id,o.order\_name,c.customer\_name from orders o***
2. ***inner join customers c on o.customer\_id=c.customer\_id;***

* Join concept is used to combine two or more tables based on specific column(Foreign key).
* Here, I used inner join to return the rows which values are common for both tables.



1. ***select o.order\_id,o.order\_name,c.customer\_name from orders o***
2. ***left join customers c on o.customer\_id=c.customer\_id;***

* Left join will return the same values but the concept is different from inner join.
* Here, Left join will return common values of both table and also the table1 (Left-**orders**) values too.



1. ***select o.order\_id,o.order\_name,c.customer\_name from orders o***
2. ***right join customers c on o.customer\_id=c.customer\_id;***

* By executing this query, I can understand the join concept.
* Right join will return the common values and table2 (right-**customers**) values.

