```
# Install MySQL and MySQLWorkbench, open MySQLWorkbench and load create-
databases.sql schema from database_Schemas folder and run it to create
sample data, we'll use it.

# LIMIT records.
# to get only first 5 customers..
>> SELECT * FROM customers LIMIT 5;

# to skip 4 customers and get 6 customers after that..
>> SELECT * FROM customers LIMIT 4, 6;

# to get customers based on ordered by (points * customer_id/100)
(new_column) in descending order.
>> SELECT *, (points * customer_id)/100 AS new_column FROM customers
ORDER BY new_column DESC;
# So we can use alias column in ORDER BY clause.

# Retrieving data from multiple tables (joins).
# we have customers and orders table, check them by select * from ...
# as you can see, orders table has column of customer_id, we want to
retrieve orders but we want to show customer name instead of customer_id.
>> SELECT * from orders INNER JOIN customers ON orders.customer_id =
customers.customer_id;
OR
>> SELECT * from orders  JOIN customers ON orders.customer_id =
customers.customer_id;

# in the output, first few columns will be orders table columns and then
customers table's columns will be there, notice that the customer_id
column appears twice, once for customers table and once for orders table.
and their values are same because that is the column on which our join is
hosted.
# let's retrieve only some needful columns.

>> SELECT order_id, first_name, last_name from orders INNER JOIN
customers ON orders.customer_id = customers.customer_id;

# try below query, it will give an error.
>> SELECT order_id, customer_id, first_name, last_name from orders INNER
JOIN customers ON orders.customer_id = customers.customer_id;
# this error is because we've added customer_id column with join which
presents in both the tables, so mysql is saying that which column should
I show..
# we have to mention the table name before column name which is
ambiguous.

>> SELECT order_id, orders.customer_id, first_name, last_name from orders
INNER JOIN customers ON orders.customer_id = customers.customer_id;
# this one should work fine.

# every time writing orders.column name, we can create table alias and
use it like orders => ord,
# and customers => cust
>> SELECT order_id, ord.customer_id, first_name, last_name from orders
ord INNER JOIN customers cust ON ord.customer_id = cust.customer_id;

# We have products and order_items table, retrieve order_id, product_id,
name, quantity, unit_price by JOIN of two tables, you'll find the column
distributed in both tables.
```

```
>> select order_id, oi.product_id, p.name, quantity, oi.unit_price FROM
order_items oi JOIN products p ON oi.product_id = p.product_id;

# notice that unit_price column is different for both the tables. so we
should show them both, but name is same so we have to mention the table
name before them, check query below.
>> select order_id, oi.product_id, p.name, quantity, oi.unit_price AS
order_unit_price, p.unit_price AS product_unit_price FROM order_items oi
JOIN products p ON oi.product_id = p.product_id;

# Self join tables.
# we have database where we have employees and offices table, in
employees table, we have reports_to column where the manager id is there
and employee_id column where employee's id is there. now, if you check,
the reports_to column's data can be similar to employee_id column because
the manager of each employee can be an employee as well.
# Ex. for first row, reports_to value is 37270.
# check the employee_id, 37270 is employee id of Yovonnada, and this row
has reports_to is null which means he has no manager meaning that he is
CEO.
# we want to retrieve each employee, but instead of reports_to column, we
want to get the name of the manager by using reports_to to employee_id.
we have to join this table to it self.

>> USE sql_hr;

>> SELECT *
    FROM employees e
    JOIN employees m
    ON e.reports_to = m.employee_id;
# weird because we are retrieving all columns, let's get needful data
only.

>> SELECT
        e.employee_id,
        e.first_name AS "Employee First Name",
        m.first_name AS "Manager First Name"
    FROM employees e
    JOIN employees m
    ON e.reports_to = m.employee_id;
# it'll show employee first name and manager first name based on
reports_to column.

# Joining multiple tables.
# we want to join orders table with customer, order_statuses table.
>> USE sql_store;

>> SELECT *
    FROM orders o
    JOIN customers c
        ON o.customer_id = c.customer_id
    JOIN order_statuses os
        ON o.status = os.order_status_id;

# doesn't look good since we haven't specified the columns. let's do it.
>> SELECT
        o.order_id,
        o.order_date,
        c.first_name,
        c.last_name,
```

```
    os.name status
FROM orders o
JOIN customers c
    ON o.customer_id = c.customer_id
JOIN order_statuses os
    ON o.status = os.order_status_id;
```