1. Write a Python program that uses the HiveQL language to create a table named "Employees" with columns for "id," "name," and "salary."

from pyhive import hive

conn = hive.Connection(host='localhost', port=10000, username='username', password='password', database='database')

cursor = conn.cursor()

query = '''

create table Employees (

id int,

name string,

salary int

)

row format delimited

fields terminated by ‘,’

'''

cursor.execute(query)

conn.commit()

cursor.close()

conn.close()

2. Create a Python program that retrieves records from a Hive table named "Customers" where the age is greater than 30.

from pyhive import hive

conn = hive.Connection(host='localhost', port=10000, username='username', password='password', database='database')

cursor = conn.cursor()

query = "select \* from customers where age > 30"

cursor.execute(query)

rows = cursor.fetchall()

cursor.close()

conn.close()

3. Write a Python script that sorts records in descending order based on the "timestamp" column in a Hive table named "Logs."

from pyhive import hive

conn = hive.Connection(host='localhost', port=10000, username='username', password='password', database='database')

cursor = conn.cursor()

query = "select \* from logs order by timestamp desc"

cursor.execute(query)

rows = cursor.fetchall()

cursor.close()

conn.close()

4. Write a Python program that connects to a Hive server using PyHive library and retrieves all records from a table named "Products".

from pyhive import hive

conn = hive.Connection(host='localhost', port=10000, username='username', password='password', database='database')

cursor = conn.cursor()

query = "select \* from Products"

cursor.execute(query)

rows = cursor.fetchall()

cursor.close()

conn.close()

5. Write a Python script that calculates the average salary of employees from a Hive table named "Employees".

from pyhive import hive

conn = hive.Connection(host='localhost', port=10000, username='username', password='password', database='database')

cursor = conn.cursor()

query = "select avg(salary) as avg\_sal from employees"

cursor.execute(query)

rows = cursor.fetchall()

cursor.close()

conn.close()

6. Implement a Python program that uses Hive partitioning to create a partitioned table named "Sales\_Data" based on the "year" and "month" columns.

from pyhive import hive

conn = hive.Connection(host='your\_host', port=your\_port, username='username', password='password', database='database')

cursor = conn.cursor()

query = '''

create table sales\_data (

id int,

sales\_amount float

)

partitioned by (year int, month int)

'''

cursor.execute(query)

cursor.close()

conn.close()

7. Develop a Python script that adds a new column named "email" of type string to an existing Hive table named "Employees."

from pyhive import hive

conn = hive.Connection(host='localhost', port=10000, username='username', password='password', database='database')

cursor = conn.cursor()

query = '''

alter table employee add column email string

'''

cursor.execute(query)

conn.commit()

cursor.close()

conn.close()

8. Create a Python program that performs an inner join between two Hive tables, "Orders" and "Customers," based on a common column.

from pyhive import hive

conn = hive.Connection(host='localhost', port=10000, username='username', password='password', database='database')

cursor = conn.cursor()

query = '''

select \* from orders

inner join customers on orders.customer\_id = customers.customer\_id

'''

cursor.execute(query)

result = cursor.fetchall()

for row in result:

print(row)

cursor.close()

conn.close()

9. Implement a Python program that uses the Hive SerDe library to process JSON data stored in a Hive table named "User\_Activity\_Logs."

from pyhive import hive

conn = hive.Connection(host='localhost', port=10000, username='username', password='password', database='database')

cursor = conn.cursor()

query = '''

create User\_Activity\_Logs (myjson struct<id:int, log:string>)

row format serde 'org.apache.hive.hcatalog.data.jsonserde'

location 'hive/warehouse/ sample.db/json\_data;

select \* from User\_Activity\_Logs;

'''

cursor.execute(query)

result = cursor.fetchall()

for row in result:

print(row)

cursor.close()

conn.close()