Case Study

on

**Python**

**Case study by,**

Mouli S,

Batch – 02,

moulisankar2002@outlook.com.

**Contents**

1. Understandings of Case Study ……………………………………………3
2. Database ……………………………………………………………………………...4-7
   1. Creation ……………………………………………………………..4
   2. Insertion …………………………………………………………….5-7
3. Case Study Questions ………………………………………………………….8-14
   1. Question 1 …………………………………………………………8
   2. Question 2 …………………………………………………………9
   3. Question 3 …………………………………………………………10

**1. Understandings of Case Study**

Btech-Chai-Wala is a start-up company by Vinit and his friends. Now-a-days in the advancement of the Internet and technologies and by considering the peoples locations they decided to start online streaming services instead of the Television. So, they have created the new streaming service which will allows only the beverage related contents.

They started subscriptions like weekly, monthly and Annual. Vinit has played roles in IT sectors like Data Analyst, Enterprise Data Architect, Principal Architect etc.

This case study questions and analysis will help the start-up to investment decisions and new features to improve Btech-Chai-Wala business by using informatica an ETL tool.

**2.1 & 2.2 Queries:**

-- Creating the database

CREATE DATABASE [btech-chai-wala];

-- Using the btech-chai-wala database

USE [btech-chai-wala];

-- Creating tables plans, subsricptions and customer

-- Creating plans table

CREATE TABLE [dbo].[plans]

(

plan\_id INT NOT NULL PRIMARY KEY IDENTITY(1,1),

plan\_name VARCHAR(100),

plan\_price DECIMAL

);

-- Creating subscriptions table

CREATE TABLE [dbo].[subscriptions]

(

subscrip\_id INT NOT NULL PRIMARY KEY IDENTITY(1,1),

cust\_id INT,

plan\_id INT,

start\_date DATE,

is\_active VARCHAR(1)

);

-- Creating customer table

CREATE TABLE [dbo].[customer]

(

cust\_id INT NOT NULL PRIMARY KEY IDENTITY(1,1),

cust\_name VARCHAR(100),

cust\_city VARCHAR(100),

cust\_dob DATE

);

-- Inserting the values into the plans, subscriptions and customer tables

--plans values

SET IDENTITY\_INSERT [dbo].[plans] ON;

INSERT INTO [dbo].[plans](plan\_id,plan\_name,plan\_price) VALUES(1,'Trail',0.00);

INSERT INTO [dbo].[plans](plan\_id,plan\_name,plan\_price) VALUES(2,'Basic\_Monthly',499.00);

INSERT INTO [dbo].[plans](plan\_id,plan\_name,plan\_price) VALUES(3,'Pro\_Monthly',999.00);

INSERT INTO [dbo].[plans](plan\_id,plan\_name,plan\_price) VALUES(4,'Pro\_Annual',11998.00);

INSERT INTO [dbo].[plans](plan\_id,plan\_name,plan\_price) VALUES(5,'Chrun',NULL);

SET IDENTITY\_INSERT [dbo].[plans] OFF;

-- subscriptions values

SET IDENTITY\_INSERT [dbo].[subscriptions] ON;

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(1,1,2,'2019-08-01',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(2,1,3,'2019-08-08',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(3,5,1,'2020-01-17',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(4,5,1,'2020-03-17',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(5,5,3,'2021-03-24',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(6,5,3,'2021-03-24',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(7,5,5,'2021-04-29',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(8,5,5,'2022-04-30',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(9,6,1,'2020-05-31',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(10,6,2,'2020-06-07',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(11,9,1,'2020-06-22',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(12,9,3,'2020-06-29',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(13,8,1,'2020-07-06',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(14,8,3,'2020-07-13',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(15,7,1,'2020-08-01',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(16,1,1,'2020-08-07',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(17,7,2,'2020-08-08',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(18,9,4,'2020-08-29',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(19,2,1,'2020-09-20',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(20,2,1,'2020-09-20',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(21,2,4,'2020-09-27',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(22,2,4,'2020-09-27',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(23,6,4,'2020-10-21',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(24,2,1,'2020-11-19',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(25,10,1,'2020-11-19',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(26,2,5,'2020-11-26',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(27,10,5,'2021-11-26',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(28,3,1,'2020-12-15',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(29,3,1,'2020-12-15',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(30,3,2,'2020-12-22',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(31,4,2,'2020-12-22',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(32,4,3,'2021-03-29',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(33,3,3,'2021-03-29',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(34,6,1,'2021-05-31',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(35,6,2,'2021-06-07',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(36,9,1,'2021-06-22',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(37,9,3,'2021-06-29',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(38,8,1,'2021-07-06',0);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(39,8,3,'2021-07-13',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(40,7,1,'2021-08-01',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(41,9,4,'2022-08-29',1);

INSERT INTO [dbo].[subscriptions](subscrip\_id, cust\_id, plan\_id, start\_date, is\_active) VALUES(42,6,4,'2021-10-21',1);

SET IDENTITY\_INSERT [dbo].[subscriptions] OFF;

-- customer values

SET IDENTITY\_INSERT [dbo].[customer] ON;

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(1,'Chaitanya','Bangalore', 2001-02-02');

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(2,'Harini','Mumbai', ‘2001-07-19’);

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(3,'Indu','Chennai', '2001-01-21');

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(4,'Jameela','Hyderabad', '2002-05-01');

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(5,'Neeha','Bangalore', '2000-03-15');

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(6,'Reshma','Mumbai', '2001-02-02');

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(7,'Sahithi','Chennai', '2000-11-02');

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(8,'Sharmila','Hyderabad', '2001-04-22');

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(9,'Sreelakshmi','Bangalore', '2000-06-17');

INSERT INTO [dbo].[customer](cust\_id, cust\_name, cust\_city, cust\_dob) VALUES(10,'Subasri','Mumbai', '2002-01-30');

SET IDENTITY\_INSERT [dbo].[customer] OFF;

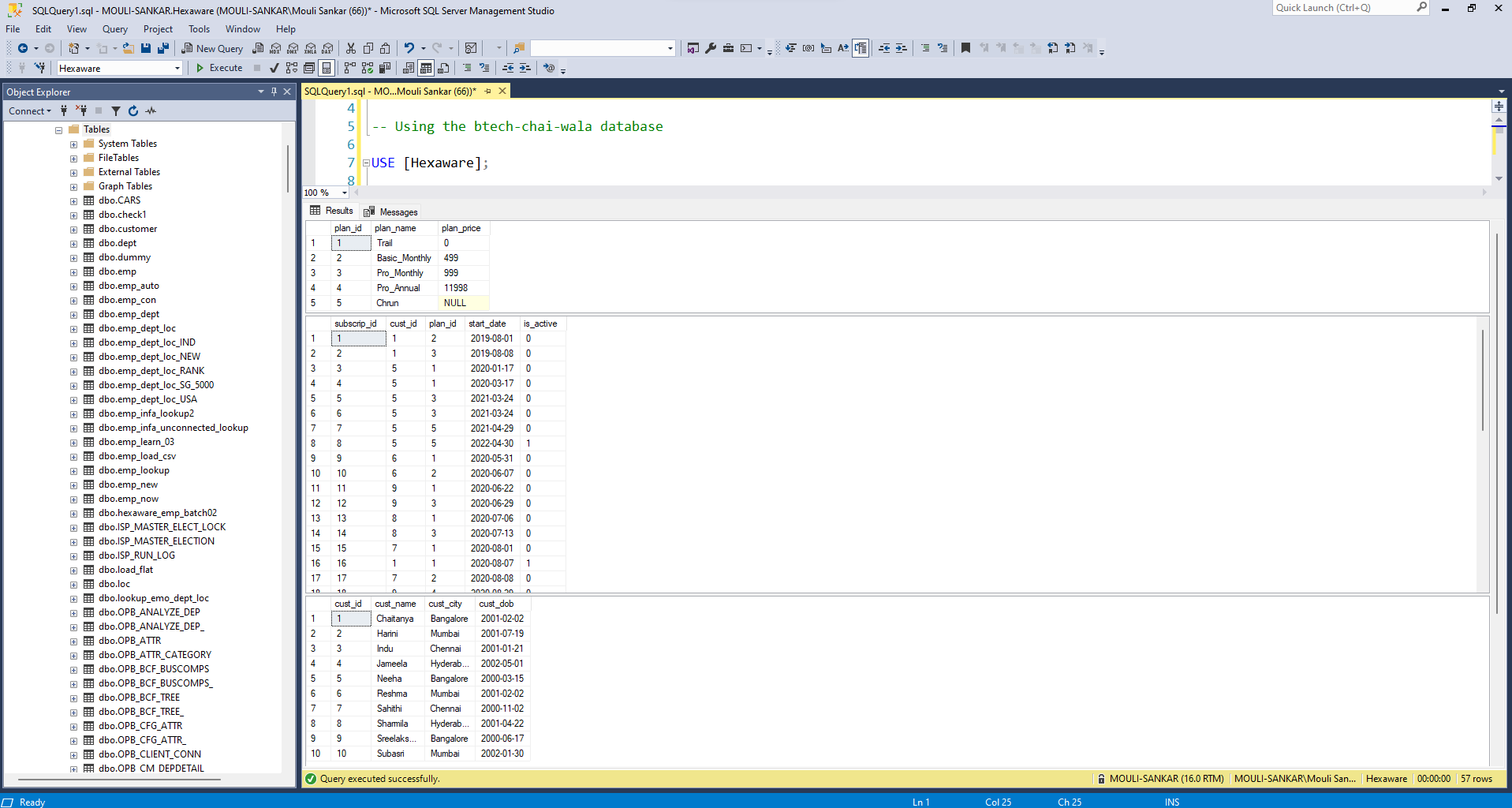
-- Verifying the values inserted properly

SELECT \* FROM [dbo].[plans];

SELECT \* FROM [dbo].[subscriptions];

SELECT \* FROM [dbo].[customer];

**Screenshots:**

****

**3. Case Study Questions**

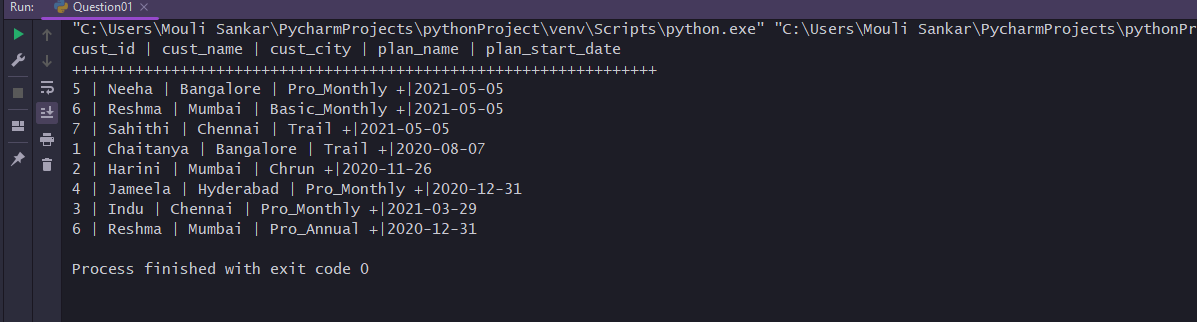
**3.1 Question 1**

**Write Python program to find how many customers of BTech Chai Wala have activate subscription as on 5th May 20217 Your output will look like:**

**Code:**

import pyodbc  
  
# Establish a connection to the SQL Server  
conn = pyodbc.connect('Driver={SQL Server};'  
 'Server=Mouli-Sankar;'  
 'Database=Hexaware;'  
 'Trusted\_Connection=yes;')  
  
# Create a cursor to execute SQL queries  
cursor = conn.cursor()  
  
# Define the SQL query  
sql\_query = '''  
SELECT [customer].[cust\_id], [customer].[cust\_name], [customer].[cust\_city],   
 [plans].[plan\_name], [subscriptions].[start\_date] as plan\_start\_date  
FROM [dbo].[subscriptions] [subscriptions]  
INNER JOIN [dbo].[customer] [customer] ON [subscriptions].[cust\_id] = [customer].[cust\_id]  
INNER JOIN [dbo].[plans] [plans] ON [subscriptions].[plan\_id] = [plans].[plan\_id]  
WHERE [subscriptions].[is\_active] = 1 AND [subscriptions].[start\_date] <= '2021-05-05'  
'''  
  
# Execute the query and fetch the results  
cursor.execute(sql\_query)  
results = cursor.fetchall()  
  
# Print the table headers  
print("cust\_id | cust\_name | cust\_city | plan\_name | plan\_start\_date")  
print("+" \* 65)  
  
# Print the query results  
for row in results:  
 print(f"{row[0]} | {row[1]} | {row[2]} | {row[3]} +|{row[4]}")  
  
# Close the cursor and connection  
cursor.close()  
conn.close()

**Output:**

****

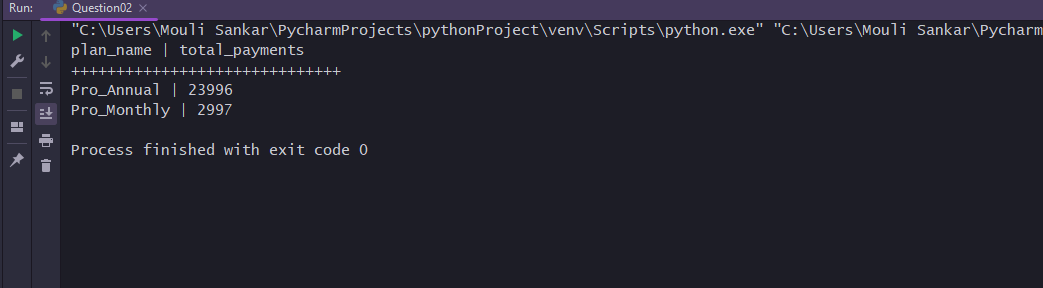
**3.2 Question 2**

**Write Python program to derive sum of customer's payment for all plan name values at 2020-1231 can be summarized.**

**Code:**

import pyodbc  
  
# Set up connection to SQL Server  
conn = pyodbc.connect('Driver={SQL Server};'  
 'Server=Mouli-Sankar;'  
 'Database=Hexaware;'  
 'Trusted\_Connection=yes;')  
# Execute SQL query to retrieve sum of payments for each plan as of Dec 31, 2020  
sql\_query = """  
SELECT p.plan\_name, SUM(p.plan\_price) as total\_payments  
FROM subscriptions s  
JOIN customer c ON s.cust\_id = c.cust\_id  
JOIN plans p ON s.plan\_id = p.plan\_id  
WHERE s.start\_date <= '2020-12-31' AND s.start\_date >= '2020-12-31'  
GROUP BY p.plan\_name  
"""  
  
# Execute SQL query and fetch results  
cursor = conn.cursor()  
cursor.execute(sql\_query)  
results = cursor.fetchall()  
  
# Print results  
print("plan\_name | total\_payments")  
print("+" \* 30)  
for row in results:  
 print(f"{row[0]} | {row[1]} ")  
  
# Close connection to SQL Server  
conn.close()

**Output:**

****

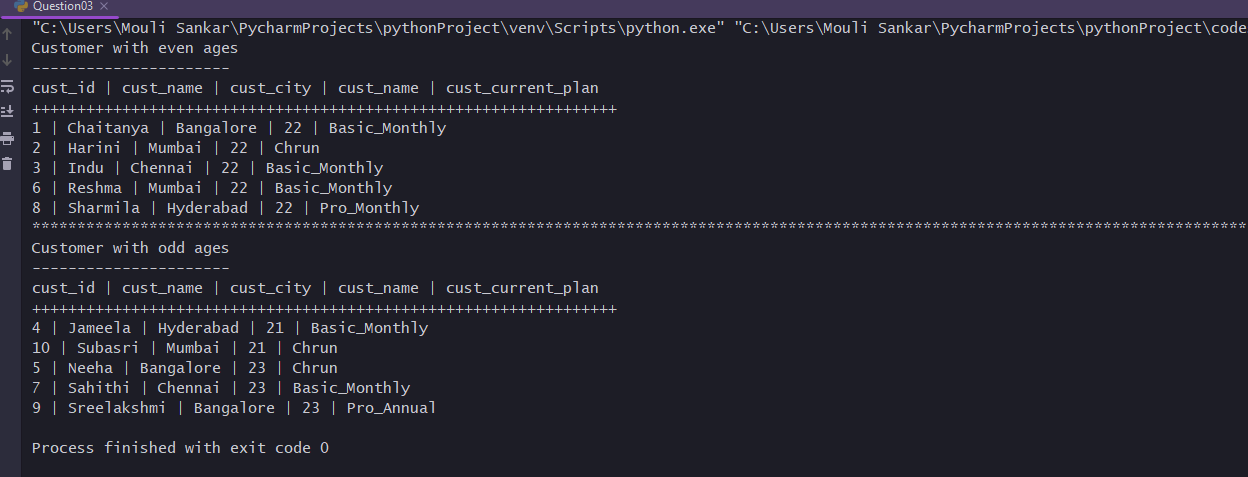
**3.3 Question 3**

**Write Python program to route Customers with even age (like 18, 20, 22, 24 etc.) goes in even table and odd ages (like 17, 19, 21, 23 etc.) goes in odd table. Age will compute on the day of job run.**

**Code:**

import pyodbc  
  
# Set up the connection to your SQL Server database  
cnxn = pyodbc.connect('Driver={SQL Server};'  
 'Server=Mouli-Sankar;'  
 'Database=Hexaware;'  
 'Trusted\_Connection=yes;')  
# Create a cursor object to execute queries  
cursor = cnxn.cursor()  
  
# Execute the SQL query to calculate the age and join the tables  
# even  
query1 = '''  
SELECT c.cust\_id, c.cust\_name, c.cust\_city, DATEDIFF(YEAR, c.cust\_dob, GETDATE()) AS cust\_age, MIN(p.plan\_name) AS plan\_name  
FROM customer c  
INNER JOIN subscriptions s ON c.cust\_id = s.cust\_id  
INNER JOIN plans p ON s.plan\_id = p.plan\_id  
WHERE DATEDIFF(YEAR, c.cust\_dob, GETDATE()) % 2 <> 1  
GROUP BY c.cust\_id, c.cust\_name, c.cust\_city, DATEDIFF(YEAR, c.cust\_dob, GETDATE())  
'''  
# odd  
query2 = '''  
SELECT c.cust\_id, c.cust\_name, c.cust\_city, DATEDIFF(YEAR, c.cust\_dob, GETDATE()) AS cust\_age, MIN(p.plan\_name) AS plan\_name  
FROM customer c  
INNER JOIN subscriptions s ON c.cust\_id = s.cust\_id  
INNER JOIN plans p ON s.plan\_id = p.plan\_id  
WHERE DATEDIFF(YEAR, c.cust\_dob, GETDATE()) % 2 = 1  
GROUP BY c.cust\_id, c.cust\_name, c.cust\_city, DATEDIFF(YEAR, c.cust\_dob, GETDATE())  
'''  
  
# Execute the query1 and fetch the results  
cursor.execute(query1)  
results1 = cursor.fetchall()  
  
# Print the results1 with the specified columns  
print("Customer with even ages")  
print('-' \* 22)  
print("cust\_id | cust\_name | cust\_city | cust\_name | cust\_current\_plan")  
print("+" \* 65)  
  
for row in results1:  
 print(f'{row.cust\_id} | {row.cust\_name} | {row.cust\_city} | {row.cust\_age} | {row.plan\_name}')  
print('\*\*\*'\*45)  
  
# Execute the query2 and fetch the results  
cursor.execute(query2)  
results2 = cursor.fetchall()  
  
# Print the results2 with the specified columns  
print("Customer with odd ages")  
print('-' \* 22)  
print("cust\_id | cust\_name | cust\_city | cust\_name | cust\_current\_plan")  
print("+" \* 65)  
  
for row in results2:  
 print(f'{row.cust\_id} | {row.cust\_name} | {row.cust\_city} | {row.cust\_age} | {row.plan\_name}')  
  
# Close the cursor and connection  
cursor.close()  
cnxn.close()

**Output:**

****