Group Assignment: Secure Data Storage and Encryption using Gnupg

Group Members:
Ishan Prabhune
Harshal Sawant
Pratik Darade
Aniket Singh
Raj Dedhia

Task1: Database Design and Setup

Enter MySQL

```
ishan@hpx360:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Create database Ecommerce

```
mysql> create database ecommerce;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| ecommerce          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| world              |
+--------------------+
6 rows in set (0.00 sec)

mysql> use ecommerce;
Database changed
```

Creating tables

```
mysql> CREATE TABLE Users (
    ->     UserID INT PRIMARY KEY,
    ->     FirstName VARCHAR(50),
    ->     LastName VARCHAR(50),
    ->     Email VARCHAR(100) UNIQUE,
    ->     Password VARCHAR(100),
    ->     RegistrationDate DATE
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> CREATE TABLE Orders (
    ->     OrderID INT PRIMARY KEY,
    ->     UserID INT,
    ->     OrderDate DATE,
    ->     TotalAmount DECIMAL(10, 2),
    ->     FOREIGN KEY (UserID) REFERENCES Users(UserID)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> CREATE TABLE Products (
    ->     ProductID INT PRIMARY KEY,
    ->     ProductName VARCHAR(100),
    ->     Price DECIMAL(10, 2),
    ->     StockQuantity INT
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> CREATE TABLE OrderDetails (
    ->     OrderDetailID INT PRIMARY KEY,
    ->     OrderID INT,
    ->     ProductID INT,
    ->     Quantity INT,
    ->     Subtotal DECIMAL(10, 2),
    ->     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
Query OK, 0 rows affected (0.01 sec)
```

Task2: Secure Data Storage

Populating table Users

```
mysql> INSERT INTO Users (UserID, FirstName, LastName, Email, Password, RegistrationDate)
    -> VALUES
    ->     (1, 'John', 'Doe', 'john.doe@example.com', 'password123', '2023-01-01'),
    ->     (2, 'Alice', 'Johnson', 'alice.j@example.com', 'securepass', '2023-01-02'),
    ->     (3, 'Michael', 'Brown', 'michael.b@example.com', 'mypass', '2023-02-28'),
    ->     (4, 'Emily', 'Smith', 'emily.s@example.com', 'pass1234', '2023-03-15'),
    ->     (5, 'William', 'Davis', 'william.d@example.com', 'qwerty', '2023-04-05'),
    ->     (6, 'Emma', 'Martinez', 'emma.m@example.com', 'abc123', '2023-04-18'),
    ->     (7, 'James', 'Anderson', 'james.a@example.com', 'p@ssw0rd', '2023-05-09'),
    ->     (8, 'Olivia', 'Garcia', 'olivia.g@example.com', 'letmein', '2023-06-22'),
    ->     (9, 'Liam', 'Hernandez', 'liam.h@example.com', 'welcome123', '2023-07-03'),
    ->     (10, 'Ava', 'Lopez', 'ava.l@example.com', '123456', '2023-08-14'),
```

```
    ->     (41, 'Ava', 'Hernandez', 'ava.h@example.com', 'faith', '2026-03-14'),
    ->     (42, 'Liam', 'Johnson', 'liam.j@example.com', 'courage', '2026-04-27'),
    ->     (43, 'Emma', 'Smith', 'emma.s@example.com', 'wisdom', '2026-05-08'),
    ->     (44, 'Olivia', 'Garca', 'olivia@example.com', 'patience', '2026-06-21'),
    ->     (45, 'Noah', 'Taylor', 'noah.t@example.com', 'kindness', '2026-07-04'),
    ->     (46, 'Sophia', 'Clark', 'sophia.c@example.com', 'compassion', '2026-08-17'),
    ->     (47, 'Liam', 'Anderson', 'liam.a@example.com', 'generosity', '2026-09-30'),
    ->     (48, 'Oliver', 'Martinez', 'oliver.m@example.com', 'humility', '2026-10-13'),
    ->     (49, 'Ava', 'Perez', 'ava.p@example.com', 'forgiveness', '2026-11-26'),
    ->     (50, 'Emma', 'Davis', 'emma.d@example.com', 'gratitude', '2026-12-07');
Query OK, 50 rows affected (0.04 sec)
Records: 50  Duplicates: 0  Warnings: 0
```

Populating table orders

```
mysql> INSERT INTO Orders (OrderID, UserID, OrderDate, TotalAmount)
    -> VALUES
    ->     (1, 1, '2023-01-10', 175.25),
    ->     (2, 2, '2023-01-15', 220.75),
    ->     (3, 3, '2023-01-20', 180.50),
    ->     (4, 4, '2023-01-25', 205.00),
    ->     (5, 5, '2023-01-30', 195.75),
    ->     (41, 41, '2023-07-30', 197.25),
    ->     (42, 42, '2023-08-04', 205.50),
    ->     (43, 43, '2023-08-09', 192.75),
    ->     (44, 44, '2023-08-14', 175.25),
    ->     (45, 45, '2023-08-19', 160.50),
    ->     (46, 46, '2023-08-24', 145.75),
    ->     (47, 47, '2023-08-29', 198.25),
    ->     (48, 48, '2023-09-03', 210.50),
    ->     (49, 49, '2023-09-08', 185.75),
    ->     (50, 50, '2023-09-13', 197.25);
Query OK, 50 rows affected (0.00 sec)
Records: 50  Duplicates: 0  Warnings: 0
```

Populate table Products

```
mysql> INSERT INTO Products (ProductID, ProductName, Price, StockQuantity)
    -> VALUES
    ->     (1, 'Laptop', 699.99, 30),
    ->     (2, 'Smartphone', 299.99, 50),
    ->     (3, 'Tablet', 199.99, 40),
    ->     (4, 'Gaming Console', 399.99, 25),
    ->     (5, 'Smart Watch', 149.99, 60),
    ->     (6, 'Desktop Computer', 899.99, 20),
    ->     (7, 'Wireless Mouse', 29.95, 100),
    ->     (8, 'External Hard Drive', 79.99, 45),
    ->     (9, 'Bluetooth Speaker', 59.99, 70),
    ->     (10, 'Printer', 129.95, 35),
```

```
    ->     (21, 'Security Camera', 129.99, 35),
    ->     (22, 'Graphic Tablet', 199.95, 20),
    ->     (23, 'Drones', 299.99, 10),
    ->     (24, 'Portable Projector', 159.95, 30),
    ->     (25, 'Gaming Keyboard', 69.99, 55),
    ->     (26, 'Wireless Router', 79.99, 40),
    ->     (27, 'Desk Lamp', 24.95, 90),
    ->     (28, 'Backpack', 39.99, 75),
    ->     (29, 'Cordless Phone', 59.95, 60),
    ->     (30, 'Headphones', 49.95, 80);
Query OK, 30 rows affected (0.01 sec)
Records: 30  Duplicates: 0  Warnings: 0
```

Populate table OrderDetails

```
mysql> INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity, Subtotal)
    -> VALUES
    ->     (1, 1, 1, 2, 1399.98),
    ->     (2, 2, 3, 4, 1199.96),
    ->     (3, 3, 5, 1, 149.99),
    ->     (4, 4, 8, 3, 239.97),
    ->     (5, 5, 10, 2, 259.90),
    ->     (6, 6, 12, 1, 79.99),
    ->     (7, 7, 14, 5, 249.95),
    ->     (8, 8, 17, 2, 69.90),
    ->     (9, 9, 19, 3, 449.97),
    ->     (10, 10, 21, 1, 199.99),
```

```
    ->     (41, 41, 27, 2, 149.90),
    ->     (42, 42, 29, 1, 79.99),
    ->     (43, 43, 2, 4, 1199.96),
    ->     (44, 44, 4, 1, 399.99),
    ->     (45, 45, 6, 2, 299.98),
    ->     (46, 46, 7, 3, 194.97),
    ->     (47, 47, 9, 1, 149.99),
    ->     (48, 48, 11, 5, 399.95),
    ->     (49, 49, 13, 2, 159.98),
    ->     (50, 50, 15, 1, 79.99);
Query OK, 50 rows affected (0.04 sec)
Records: 50  Duplicates: 0  Warnings: 0
```

Exporting table data

```
ishan@hpx360:~$ mysqldump -u root -p ecommerce > ecom_data.sql
Enter password:
ishan@hpx360:~$
```

Task 3: Encryption using GnuPG

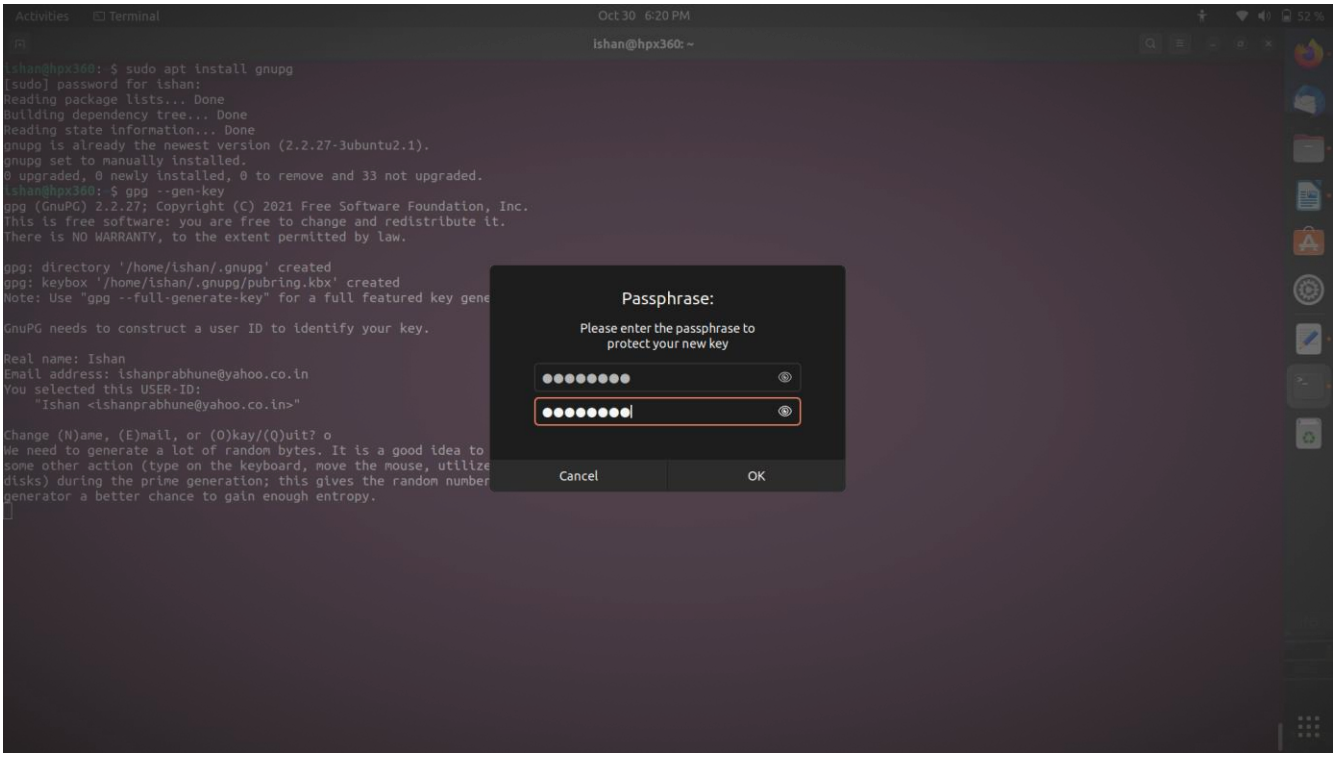
Installing GnuPG

```
ishan@hpx360:~$ sudo apt install gnupg
[sudo] password for ishan:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 33 not upgraded.
ishan@hpx360:~$
```

Displaying the keys of Gnupg

```
ishan@hpx360:~$ gpg --gen-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/ishan/.gnupg' created
gpg: keybox '/home/ishan/.gnupg/pubring.kbx' created
Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Ishan
Email address: ishanprabhune@yahoo.co.in
You selected this USER-ID:
    "Ishan <ishanprabhune@yahoo.co.in>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/ishan/.gnupg/trustdb.gpg: trustdb created
gpg: key E238785FAA151885 marked as ultimately trusted
gpg: directory '/home/ishan/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/ishan/.gnupg/openpgp-revocs.d/324ED9FFFE6EDE03451E12CFE238785FAA151885.rev'
public and secret key created and signed.

pub   rsa3072 2023-10-30 [SC] [expires: 2025-10-29]
      324ED9FFFE6EDE03451E12CFE238785FAA151885
uid                      Ishan <ishanprabhune@yahoo.co.in>
sub   rsa3072 2023-10-30 [E] [expires: 2025-10-29]

ishan@hpx360:~$
```

setting a password for gnupg



Listing keys and exporting our key

```
ishan@hpx360:~$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2025-10-29
/home/ishan/.gnupg/pubring.kbx
------------------------------
pub   rsa3072 2023-10-30 [SC] [expires: 2025-10-29]
      324ED9FFFE6EDE03451E12CFE238785FAA151885
uid           [ultimate] Ishan <ishanprabhune@yahoo.co.in>
sub   rsa3072 2023-10-30 [E] [expires: 2025-10-29]

ishan@hpx360:~$ gpg --output public_key.gpg --armor --export ishanprabhune@yahoo.co.in
ishan@hpx360:~$
```

installing GnuPG for python pip

```
ishan@hpx360:~$ pip3 install gnupg
Defaulting to user installation because normal site-packages is not writeable
Collecting gnupg
  Downloading gnupg-2.3.1.tar.gz (100 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100.4/100.4 kB 2.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting psutil>=1.2.1 (from gnupg)
  Downloading psutil-5.9.6-cp36-abi3-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (21 kB)
Downloading psutil-5.9.6-cp36-abi3-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (283 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 283.6/283.6 kB 14.8 MB/s eta 0:00:00
Building wheels for collected packages: gnupg
  Building wheel for gnupg (setup.py) ... done
  Created wheel for gnupg: filename=gnupg-2.3.1-py3-none-any.whl size=94622 sha256=cfd2a18b9efe3c5ca77f070696bded17f9eec323bad19de4ba9a1fa522ef7d3b
  Stored in directory: /home/ishan/.cache/pip/wheels/96/12/15/82eadc149bbdb08063f4ce8b1b3971cc5d0e442707368f995f
Successfully built gnupg
Installing collected packages: psutil, gnupg
Successfully installed gnupg-2.3.1 psutil-5.9.6
```

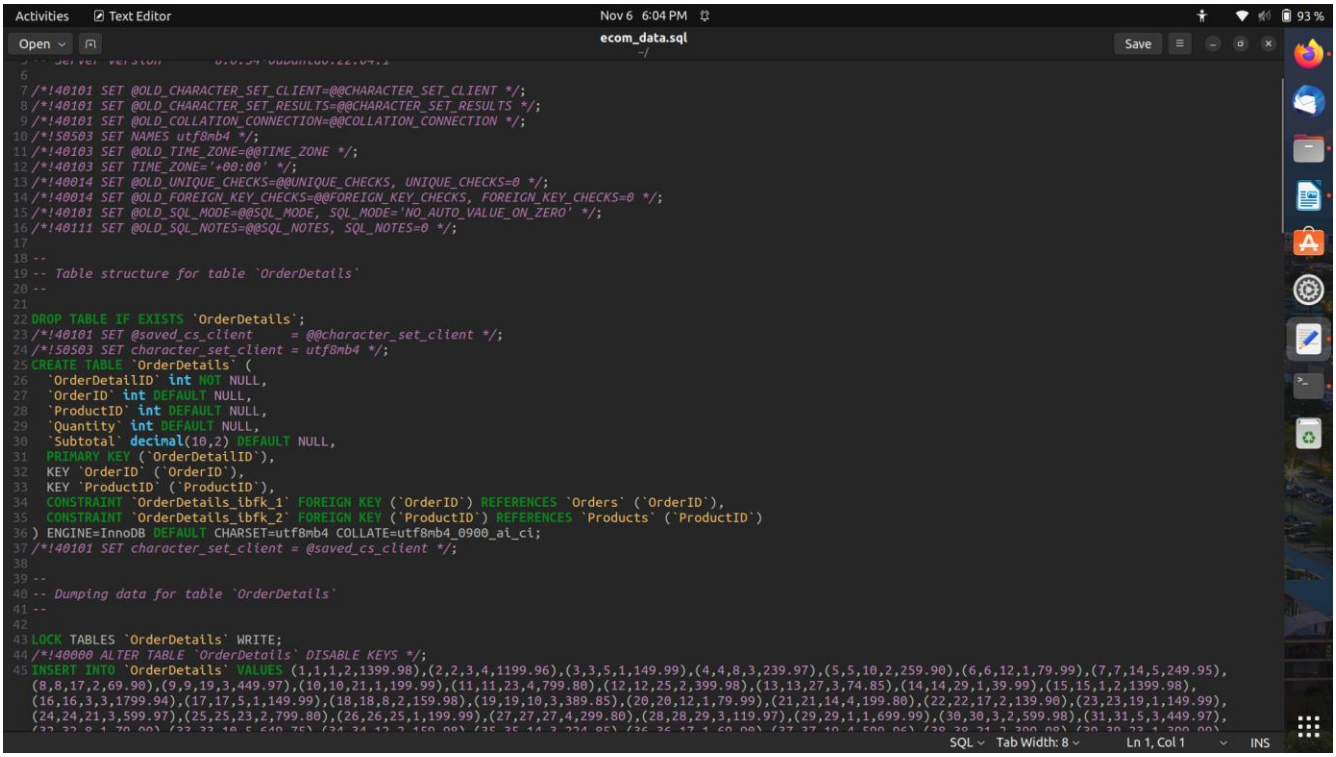
installing mysql connector

```
ishan@hpx360:~$ pip3 install mysql-connector-python
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.2.0-cp310-cp310-manylinux_2_17_x86_64.whl.metadata (2.1 kB)
Collecting protobuf<=4.21.12,>=4.21.1 (from mysql-connector-python)
  Downloading protobuf-4.21.12-cp37-abi3-manylinux2014_x86_64.whl (409 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 409.8/409.8 kB 7.9 MB/s eta 0:00:00
Downloading mysql_connector_python-8.2.0-cp310-cp310-manylinux_2_17_x86_64.whl (31.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 31.6/31.6 MB 36.5 MB/s eta 0:00:00
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.2.0 protobuf-4.21.12
ishan@hpx360:~$
```

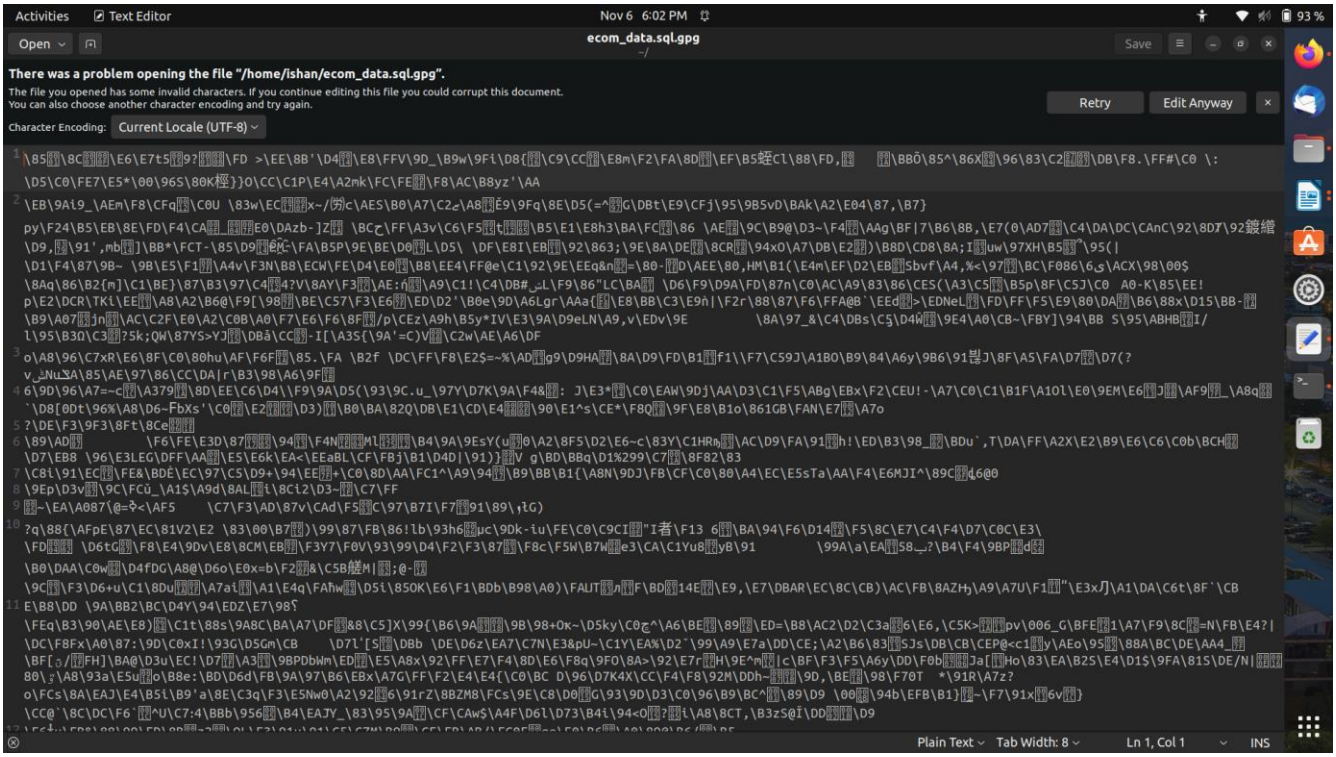
encrypting entire file using GnuPG

```
ishan@hpx360:~$ gpg -e -r ishanprabhune@yahoo.co.in ecom_data.sql
```

File before encrypting



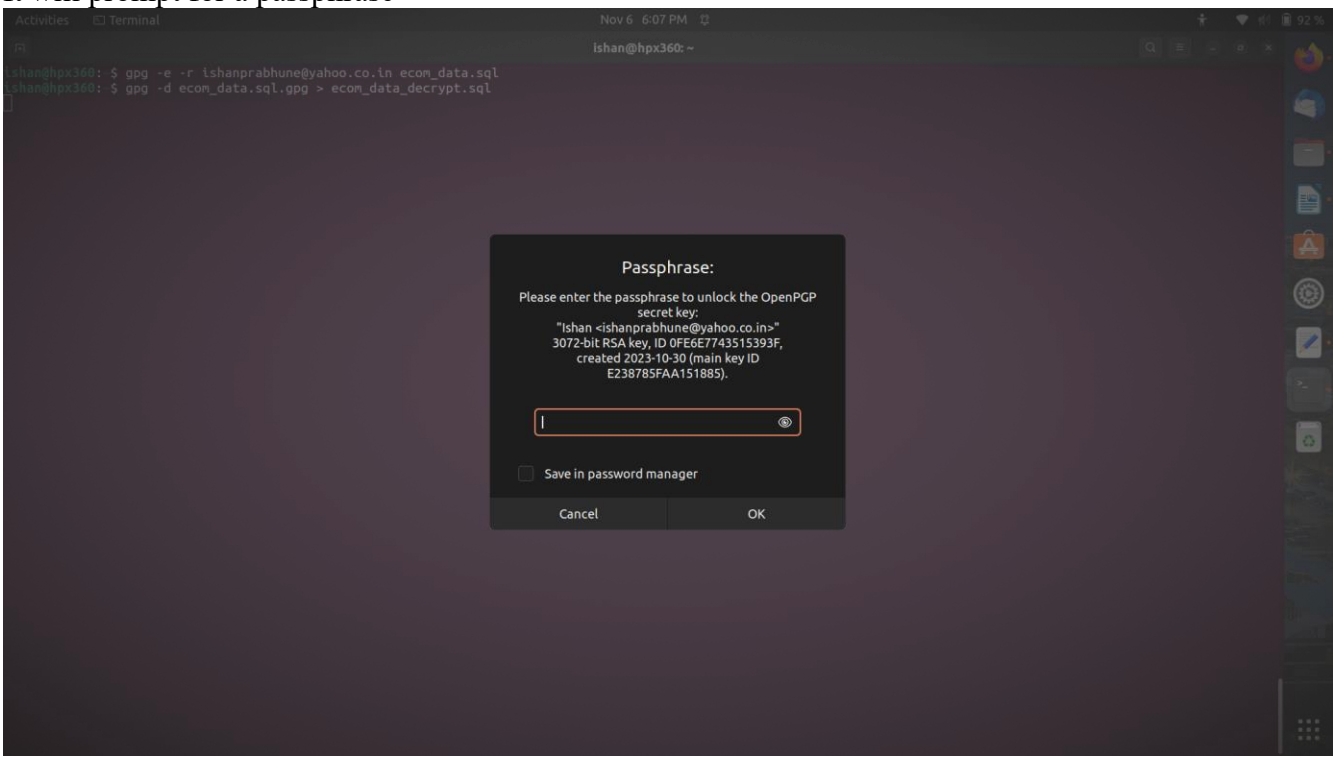
File after encrypting



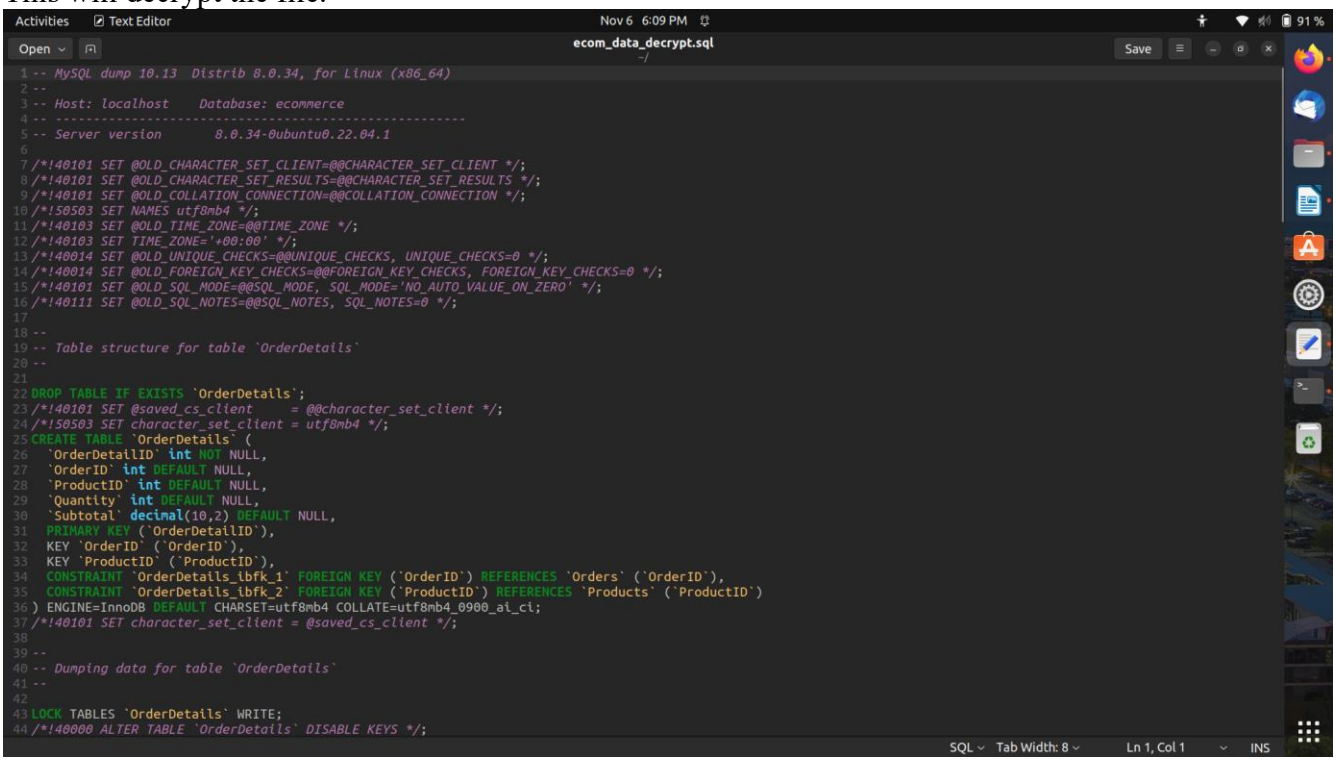
Procedure for decryption

```
ishan@hpx360:~$ gpg -d ecom_data.sql.gpg > ecom_data_decrypt.sql
gpg: encrypted with 3072-bit RSA key, ID 0FE6E7743515393F, created 2023-10-30
      "Ishan <ishanprabhune@yahoo.co.in>"
```

It will prompt for a passphrase

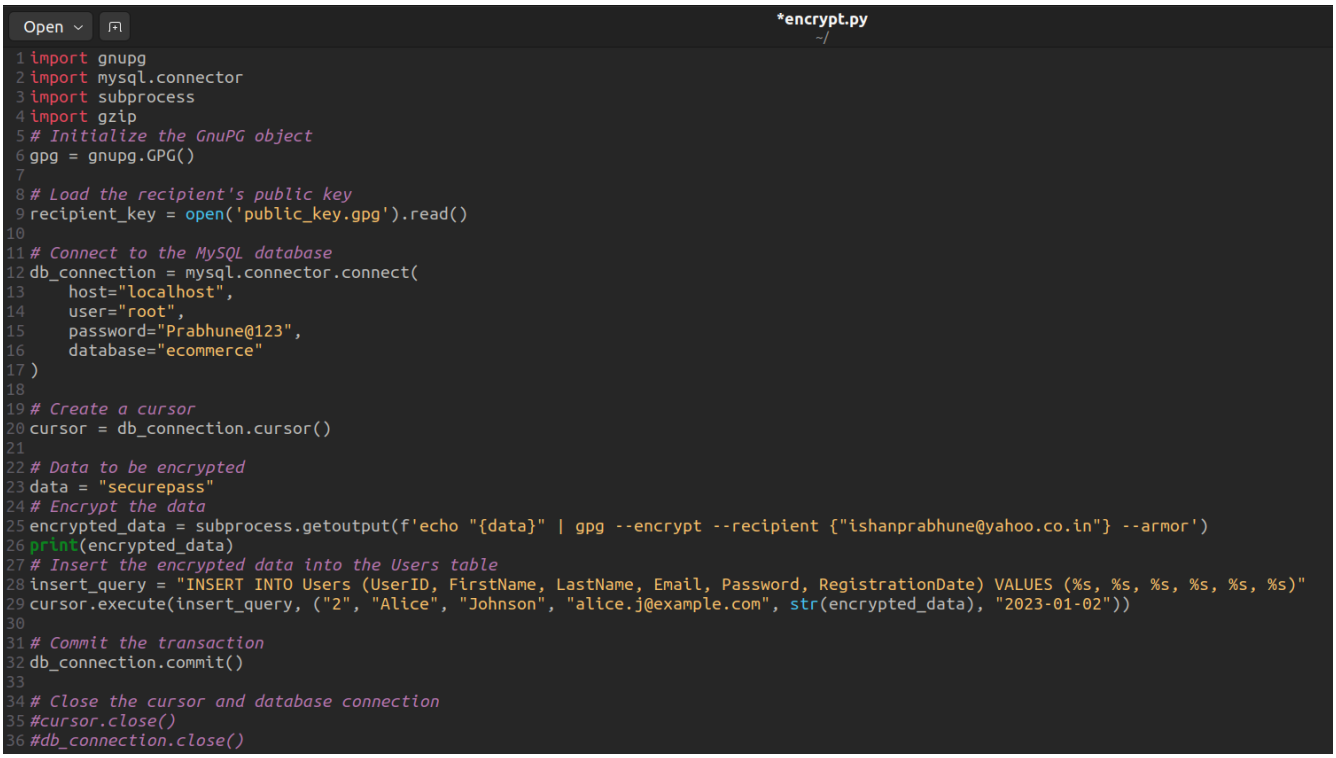


This will decrypt the file.




```sql
-- MySQL dump 10.13  Distrib 8.0.34, for Linux (x86_64)
--
-- Host: localhost    Database: ecommerce
-- ------------------------------------------------------
-- Server version       8.0.34-0ubuntu0.22.04.1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `OrderDetails`
--

DROP TABLE IF EXISTS `OrderDetails`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `OrderDetails` (
  `OrderDetailID` int NOT NULL,
  `OrderID` int DEFAULT NULL,
  `ProductID` int DEFAULT NULL,
  `Quantity` int DEFAULT NULL,
  `Subtotal` decimal(10,2) DEFAULT NULL,
  PRIMARY KEY (`OrderDetailID`),
  KEY `OrderID` (`OrderID`),
  KEY `ProductID` (`ProductID`),
  CONSTRAINT `OrderDetails_ibfk_1` FOREIGN KEY (`OrderID`) REFERENCES `Orders` (`OrderID`),
  CONSTRAINT `OrderDetails_ibfk_2` FOREIGN KEY (`ProductID`) REFERENCES `Products` (`ProductID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `OrderDetails`
--

LOCK TABLES `OrderDetails` WRITE;
/*!40000 ALTER TABLE `OrderDetails` DISABLE KEYS */;
```

Python script to integrate GnuPG encryption into the database operations
This script will encrypt the password for UserID = 2 in the Users table.




```python
import gnupg
import mysql.connector
import subprocess
import gzip
# Initialize the GnuPG object
gpg = gnupg.GPG()

# Load the recipient's public key
recipient_key = open('public_key.gpg').read()

# Connect to the MySQL database
db_connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Prabhune@123",
    database="ecommerce"
)

# Create a cursor
cursor = db_connection.cursor()

# Data to be encrypted
data = "securepass"
# Encrypt the data
encrypted_data = subprocess.getoutput(f'echo "{data}" | gpg --encrypt --recipient {"ishanprabhune@yahoo.co.in"} --armor')
print(encrypted_data)
# Insert the encrypted data into the Users table
insert_query = "INSERT INTO Users (UserID, FirstName, LastName, Email, Password, RegistrationDate) VALUES (%s, %s, %s, %s, %s, %s)"
cursor.execute(insert_query, ("2", "Alice", "Johnson", "alice.j@example.com", str(encrypted_data), "2023-01-02"))

# Commit the transaction
db_connection.commit()

# Close the cursor and database connection
#cursor.close()
#db_connection.close()
```

Running python script. This is the encrypted password.

```
ishan@hpx360:~$ python3 encrypt.py
-----BEGIN PGP MESSAGE-----

hQGMAw/m53Q1FTk/AQv/a+QtIBlB2XpsCzPpX5uMYvzMNUsdC6kiIwMX7oo5SY2/
mf9zB1lpwsD5TksvXPnG4nb41q7O7u3LVTy5eLuNZttMqvxa1VXa3p4ZWkX2yKYd
hgufpmIC3vJnlBQmyDTZGrNEIy14TSpfFafrR/GyUB/le6fBb2OlTvQOYt4CyNUr
p0umB/OueQYmDSbx3aOS1944w64aUyeXgZGuJaOcKxYIGMcUpRwoCoxiRHDFMqQ6
JIWdvQXgM1xnBNO5/rb1EvKsKSP4dUWsDLm6do5eeiGFO4c30yqI3SVGKM9fZfDn
KNE1rWANhtrvALJE/r2Fck9bQy1ppBh4/OL02Nw622dnERa+39SNSeEANo95DK21
NSM+31WgDhkMKzyhkLPPlgyZZr8ajkCaTffvpfZcrqR6XuNVietWyN9gi9B7OsGw
EtfXYKA9QJpTht4PmA46dm1uZB1MlwwJr11w3OIEBrnrYXKJIsjla2KI8BhL0gxb
YeuLWyJBxBd2osTOudFK0kMBquWzUcNAWHLWy2CBdZzma1U+w26Mj9S+BOtHOJo1
1KdtVTllceGJQ3VF6lroARXSOzam6QRyzYYRjSxeknx1aU36
=Vtg5
-----END PGP MESSAGE-----
```

Script has encypted the password data for UserID =2.

Data is not visible in the table as it is encrypted for UserID = 2.

UserID	FirstName	LastName	Email	Password	RegistrationDate
0	Ishan	Prabhune	ishan.p@example.com	password1234	2023-01-09
1	John	Doe	john.doe@example.com	password123	2023-01-01
2	Alice	Johnson	alice.j@example.com		2023-01-02
3	Michael	Brown	michael.b@example.com		2023-02-28
4	Emily	Smith	emily.s@example.com		2023-03-15
5	William	Davis	william.d@example.com	qwerty	2023-04-05
6	Emma	Martinez	emma.m@example.com	abc123	2023-04-18
7	James	Anderson	james.a@example.com	p@ssw0rd	2023-05-09
8	Olivia	Garcia	olivia.g@example.com	letmein	2023-06-22
9	Liam	Hernandez	liam.h@example.com	welcome123	2023-07-03
10	Ava	Lopez	ava.l@example.com	123456	2023-08-14

Script for decryption

```
# Select encrypted data from the Customers table
select_query = "SELECT Password FROM Users WHERE UserID = 2"
cursor.execute(select_query)
encrypted_data = cursor.fetchone()[0]

# Decrypt the data
decrypted_data = gpg.decrypt(encrypted_data, passphrase='prabhune')

# Use the decrypted data
print(decrypted_data.data)

# Close the cursor and database connection
cursor.close()
db_connection.close()
```

Script has decrypted the Password Data for UserID = 2

UserID	FirstName	LastName	Email	Password	RegistrationDate
0	Ishan	Prabhune	ishan.p@example.com	password1234	2023-01-09
1	John	Doe	john.doe@example.com	password123	2023-01-01
2	Alice	Johnson	alice.j@example.com	securepass	2023-01-02
3	Michael	Brown	michael.b@example.com		2023-02-28
4	Emily	Smith	emily.s@example.com		2023-03-15
5	William	Davis	william.d@example.com	qwerty	2023-04-05
6	Emma	Martinez	emma.m@example.com	abc123	2023-04-18
7	James	Anderson	james.a@example.com	p@ssw0rd	2023-05-09
8	Olivia	Garcia	olivia.g@example.com	letmein	2023-06-22
9	Liam	Hernandez	liam.h@example.com	welcome123	2023-07-03
10	Ava	Lopez	ava.l@example.com	123456	2023-08-14