Student ID:22101071d

Student Name: Zhu Jin Shun

**Question 1: Functional Dependency and Normalization**

a)

Top FD:

**PlantNo** 🡪 RecorderPoint, QtyOnHand

**ItemNo, PlantNo** 🡪 ItemDesc

**OrderNo**🡪 ShipAddr,OrderDate,CustNo

**ItemNo, OrderNo**🡪LineNo, QtyOrdered, QtyOustanding

**CustNo** 🡪 CustBal, CustDiscount

Bottom FD:

**LineNo, OrderNo**🡪 ItemNo, QtyOrdered, QtyOustanding

**ItemNo, OrderNo** 🡪 LineNo

b)

Underline🡪Primary Key

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| OrderNo | ItemNo | PlantNo | LineNo | Orderdate | ShipAddr | QtyOrdered | QtyOutstanding | CustNo | ItemDesc | QtyOnHand | ReorderPoint | CustBal | CustDiscount |
| O001 | I001 | P001 | L001 | 2024/10/5 | PQ306 | 10 | 5 | C001 | IA | 50 | RP01 | 1000 | 5% |
| O001 | I002 | P001 | L002 | 2024/10/5 | PQ306 | 5 | 2 | C001 | IB | 30 | RP01 | 1000 | 5% |
| O002 | I001 | P002 | L001 | 2024/10/6 | PQ307 | 20 | 10 | C002 | IA | 60 | RP02 | 1500 | 10% |
| O002 | I002 | P002 | L002 | 2024/10/6 | PQ307 | 15 | 15 | C002 | IC | 40 | RP02 | 1500 | 10% |

c)

2NF Table:

**OrderNo** 🡪 ShipAddr, OrderDate, CustNo

|  |  |  |  |
| --- | --- | --- | --- |
| OrderNo | ShipAddr | OrderDate | CustNo |
|  |  |  |  |

**ItemNo, OrderNo**🡪LineNo, QtyOrdered, QtyOustanding

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ItemNo | OrderNo | LineNo | QtyOrdered | QtyOustanding |
|  |  |  |  |  |

**ItemNo** 🡪 ItemDesc

|  |  |
| --- | --- |
| ItemNo | ItemDesc |
|  |  |

**LineNo, OrderNo**🡪 ItemNo, QtyOrdered, QtyOustanding

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LineNo | OrderNo | ItemNo | QtyOrdered | QtyOustanding |
|  |  |  |  |  |

**PlantNo** 🡪 RecorderPoint，QtyOnHand

|  |  |  |
| --- | --- | --- |
| PlantNo | RecorderPoint | QtyOnHand |
|  |  |  |

**CustNo** 🡪 CustBal, CustDiscount

|  |  |  |
| --- | --- | --- |
| CustNo | CustBal | CustDiscount |
|  |  |  |

d)

3NF Table

**OrderNo** 🡪 ShipAddr, OrderDate, CustNo，CustBal, CustDiscount

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OrderNo | ShipAddr | OrderDate | CustNo | CustBal | CustDiscount |
|  |  |  |  |  |  |

**OrderNo, ItemNo, LineNo** 🡪 QtyOrdered, QtyOutstanding

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderNo | ItemNo | LineNo | QtyOrdered | QtyOutstanding |
|  |  |  |  |  |

**ItemNo** 🡪 ItemDesc

|  |  |
| --- | --- |
| ItemNo | ItemDesc |
|  |  |

**PlantNo** 🡪 RecorderPoint, QtyOnHand

|  |  |  |
| --- | --- | --- |
| PlantNo | RecorderPoint | QtyOnHand |
|  |  |  |

e)

Possible BCNF Tables:

**OrderNo** 🡪 ShipAddr, OrderDate, CustNo，CustBal，CustDiscount

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OrderNo | ShipAddr | OrderDate | CustNo | CustBal | CustDiscount |
|  |  |  |  |  |  |

**OrderNo, ItemNo , LineNo** 🡪 QtyOrdered, QtyOutstanding

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderNo | ItemNo | LineNo | QtyOrdered | QtyOutstanding |
|  |  |  |  |  |

**ItemNo** 🡪 ItemDesc

|  |  |
| --- | --- |
| ItemNo | ItemDesc |
|  |  |

**PlantNo** 🡪 RecorderPoint, QtyOnHand

|  |  |  |
| --- | --- | --- |
| PlantNo | RecorderPoint | QtyOnHand |
|  |  |  |

f)

**ShipAddr** 🡪 CustNo，CustBal,CustDiscount

**PlantNo** 🡪 RecorderPoint, QtyOnHand

**ItemNo, PlantNo** 🡪 ItemDesc

**OrderNo**🡪 ShipAddr,OrderDate

**ItemNo, OrderNo**🡪LineNo, QtyOrdered, QtyOustanding

Bottom FD:

**LineNo, OrderNo**🡪 ItemNo, QtyOrdered, QtyOustanding

I won’t say it reasonable, as ShipAddr won’t be unique for every customer, customer from same family will have same ShipAddr but different CustNo. If ShipAddr determines CustNo, customer from the same family will all have same CustNo which won’t be accurate and unique for the relationship design process. Also, as OrderNo defines CustNo previously, changing to let ShipAddr defining CustNo won’t change the original relationship as OrderNo still determines ShipAddr, which means OrderNo can still determine CustNo. So having ShipAddr to determine CustNo I would say isn’t reasonable.

**Question 2: Securing Application Passwords**

a)

Symmetric encryption can be used to store passwords through the two-tiered key based architecture. The encryption key is responsible for encrypting user passwords before they are stored in the database, ensuring that only encrypted versions are saved, and plaintext passwords are never exposed. The encryption key can be used to perform encryption which is to convert plaintext into ciphertext and decryption which to convert ciphertext back into plaintext. The master key is stored and managed by an external security module and is used to encrypt the encryption key itself. So even the attacker can have access with the encryption key, they can’t unlock it and access the passwords because they don’t have the master key, meanwhile, the users will still have access to the master key and could unlock the encryption key with their own passwords.

The storage of Relevant Keys:

The TDE tablespace encryption Key is encrypted and can only be encrypted by the TDE master encryption key.

The TDE master encryption key’s information and encrypted values of column encryption will be obtained in an external security module like the TPM or HSM.

b)

This method can be immune to dictionary attacks.

The reason is because this method ensures the passwords are encrypted on client side before sending to database, so the server never sees the plaintext data. Only the client application has access to encryption keys which can encrypt or decrypt data, so when attackers perform dictionary attack, they can only see encrypted values instead of accessing plaintext passwords or passwords hashes.

**Question 3: Securing Data-in-motion in MySQL**

a)

SSL: Not in use

Server version: 8.0.39 MySQL Community Server - GPL

文本

描述已自动生成

b)

I chose mode REQUIRED

SSL: Cipher in use is TLS\_AES\_256\_GCM\_SHA384

文本

描述已自动生成

c)

Issuer: CN=MySQL\_Server\_8.0.39\_Auto\_Generated\_CA\_Certificate

文本

描述已自动生成

Certificate Screenshot

文本

描述已自动生成

SHOW VARIABLES LIKE '%ssl%' screenshot

d)

Command 1:

(CA): openssl ecparam -out C:\myCA\private\ca\_private\_key.pem -name prime256v1 -genkey

(Server 1: Localhost):

openssl ecparam -out C:\myCA\private\localhost\_private\_key.pem -name prime256v1 -genkey

(Server 2: IAmAHacker.com):

openssl ecparam -out C:\myCA\private\IAmAHacker.com\_private\_key.pem -name prime256v1 -genkey

Command 2

(Server 1: Localhost):

openssl req -new -key C:\myCA\private\localhost\_private\_key.pem -out C:\myCA\localhost.csr -config C:\myCA\private\openssl.cnf

(Server 2: IAmAHacker.com):

openssl req -new -key C:\myCA\private\IAmAHacker.com\_private\_key.pem -out C:\myCA\IAmAHacker.csr -config C:\myCA\private\openssl.cnf

Command 3:

(CA certificate): openssl req -x509 -new -key C:\myCA\private\ca\_private\_key.pem -out C:\myCA\ca\_certificate.pem -days 365 -config C:\myCA\private\openssl.cnf

Command 4:

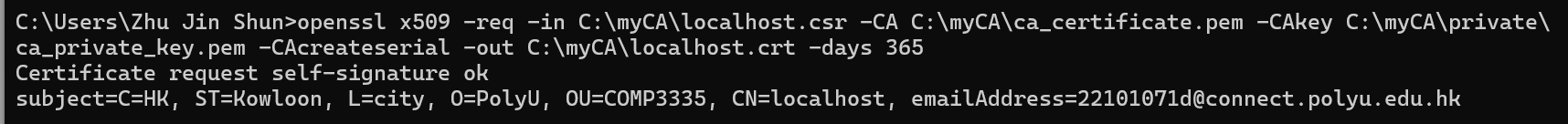
(Server 1: Localhost):

openssl x509 -req -in C:\myCA\localhost.csr -CA C:\myCA\ca\_certificate.pem -CAkey C:\myCA\private\ca\_private\_key.pem -CAcreateserial -out C:\myCA\localhost.crt -days 365

(Server 2: IAmAHacker.com):

openssl x509 -req -in C:\myCA\IAmAHacker.csr -CA C:\myCA\ca\_certificate.pem -CAkey C:\myCA\private\ca\_private\_key.pem -CAcreateserial -out C:\myCA\IAmAHacker.crt -days 365

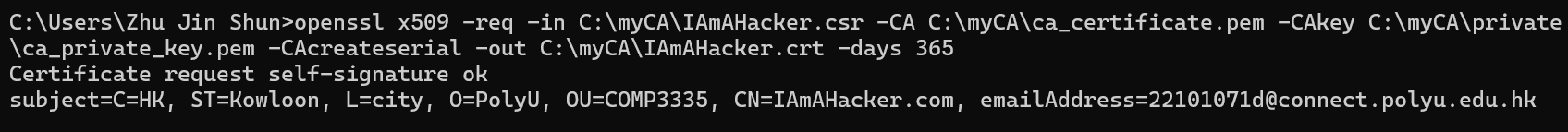
Certificate 1:



图形用户界面, 文本, 应用程序

描述已自动生成

Certificate 2:



图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

e)

Lines Added to mysqld:

文本

描述已自动生成

Lines Added to mysql:

文本

描述已自动生成

f)

All two connections failed due to ERROR 2026 (HY000): SSL connection error: error:0A000086:SSL routines::certificate verify failed.

文本

描述已自动生成

The --ssl-mode setting plays an important role in encrypted connections in MySQL servers, it determines the level of verification on the SSL certificate, which is improving the security level of data transmission.

The effects caused on the verification can be different due to the mode chosen by the user. Stricter mode like the VERIFY\_IDENTITY and VERIFY\_CA provides higher security level of protection than DISABLED mode. Higher security mode can ensure the verification of the certificate through a recognized CA and ensuring that the hostname matches with the certificate details to prevent attacks.