2.1

This table is not in any normal form (unnormalized), as there are repeating groups (Invoice-No, Client-No, Client-Name).

2.2  
First normalise into 1NF  
Remove repeating groups (Invoice-No, Client-No, Client-Name)  
Identify Primary key  
Identify Dependencies   
  
Normalise to 2NF  
Remove Partial dependencies  
  
Normalise to 3NF  
Remove transative dependencies  
  
2.3  
(Invoice-No, Client-No, Client-Name, Product-No, Product-Name, Product-Price, Quantity)  
filling in record 2 of (Invoice-No, Client-No, Client-Name) with (1234,4321, John) respectively.

2.4  
Invoice-No, Client-No, Product-No 🡪 Client-Name, Product-Name, Product-Price, Quantity

Client-No🡪 Client-Name  
Product-No 🡪 Product-Name, Product-Price  
Invoice-No🡪 Client-No, Product-No  
Invoice-No, Product-No 🡪 Quantity

2.5

Client(Client-No, ClientName)  
Product(Product-No, ProductName, Product-Price)  
InvoiceLine(Invoice-No, Product-No, Quantity)  
Invoice(InvoiceNo, CLientNo, Product-No)  
  
2.6 There are no transative dependencies, therefore 2.5 is in 3NF, therefore  
3NF is  
  
Client(Client-No, ClientName)  
Product(Product-No, ProductName, Product-Price)  
InvoiceLine(Invoice-No, Product-No, Quantity)  
Invoice(InvoiceNo, CLientNo, Product-No)  
  
2.7  
Converting human readable tables, to DBMS readable tables.  
Higher abstraction…. Unknown look textbook  
  
2.8  
slides…  
  
3.1  
Create Table Client(  
Client-No char(10) UNIQUE NOT NULL,  
Client-Name char(50) NOT NULL,  
PRIMARY KEY(CLIENT-No)  
);

Create Table Product(

Product-No char(10) UNIQUE NOT NULL,   
ProductName char(50) NOT NULL,   
Product-Price NUMERIC (10,2) NOT NULL,  
PRIMARY KEY(PRODUCT-No)

);

Create Table Ivoice(

InvoiceNo char(10) UNIQUE NOT NULL,   
Client-No char(10) NOT NULL,  
Product-No char(10) NOT NULL,  
PRIMARY KEY(InvoiceNo),  
FOREIGN KEY(Client-No) REFERENCES Client,  
FOREIGN KEY(Product-No) REFERENCES Product

);

Create Table InvoiceLine(

InvoiceNo char(10) NOT NULL,   
Product-No char(10) NOT NULL,  
Quantity int NOT NULL,  
PRIMARY KEY(InvoiceNo, Product-No),  
FOREIGN KEY(Invoice-No) REFERENCES Invoice,  
FOREIGN KEY(Product-No) REFERENCES Product

);

3.2

INSERT INTO PRODUCT  
VALUES (‘1234567890’,’GPU’, 1000.50)  
  
3.3  
SELECT SUM(Quantity)  
FROM INvoiceLine

3.4  
SELECT SUM(Quantity\* ProductNo.ProductPrice)  
FROM   
Invoice INNER JOIN InvoiceLine  
ON Invoice.InvoiceNo = InvoiceLine.InvoiceNo

Invoice INNER JOIN Product  
ON InvoiceLine.ProductNo = Product.ProductNo

InvoiceNo char(10) NOT NULL,   
Product-No char(10) NOT NULL,