



Northeastern

GROUP 8

Online Shopping Database

Data Management and Database Design
Instructor – Simon Wang

Agenda

- 1 Database Design Document
- 2 Database ERD
- 3 Database Implementation
- 4 Reporting Views & Visualization
- 5 Conclusions
- 6 Q & A



Design Document - Database Purpose



1 Track and report customers retention and preference



2 Make efficient marketing plan



3 Improve profits and sales

Design Document - Business Problems



Provide information to improve profits and sales.



Allow staff to track customer order to make sure the items arrived on time and keep the customer satisfied.



Analyze data for administrative staff to make effective decision when purchase inventory items.



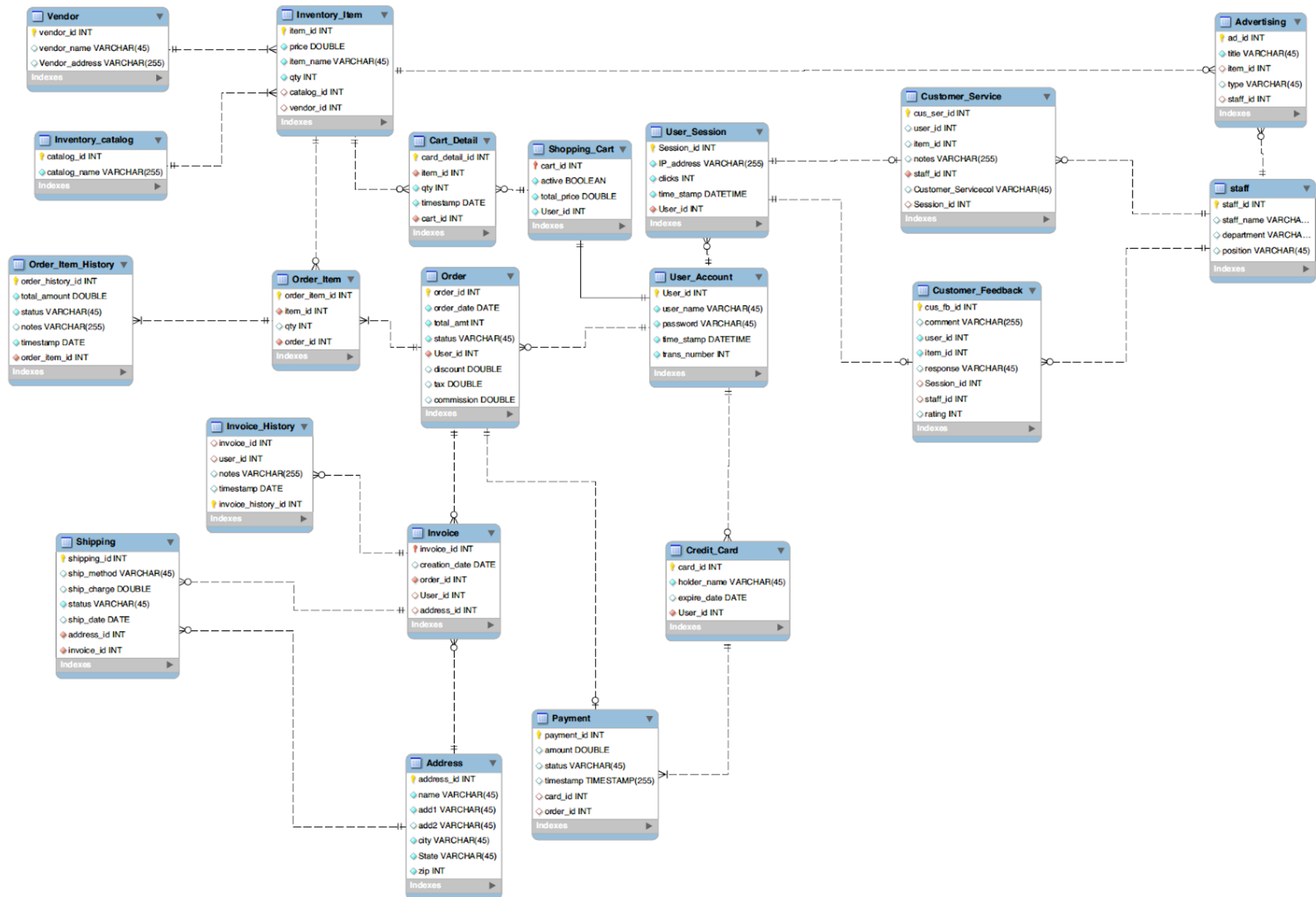
Collect and analyze customers feedback to improve the quality and efficiency of customer service.

Database ERD

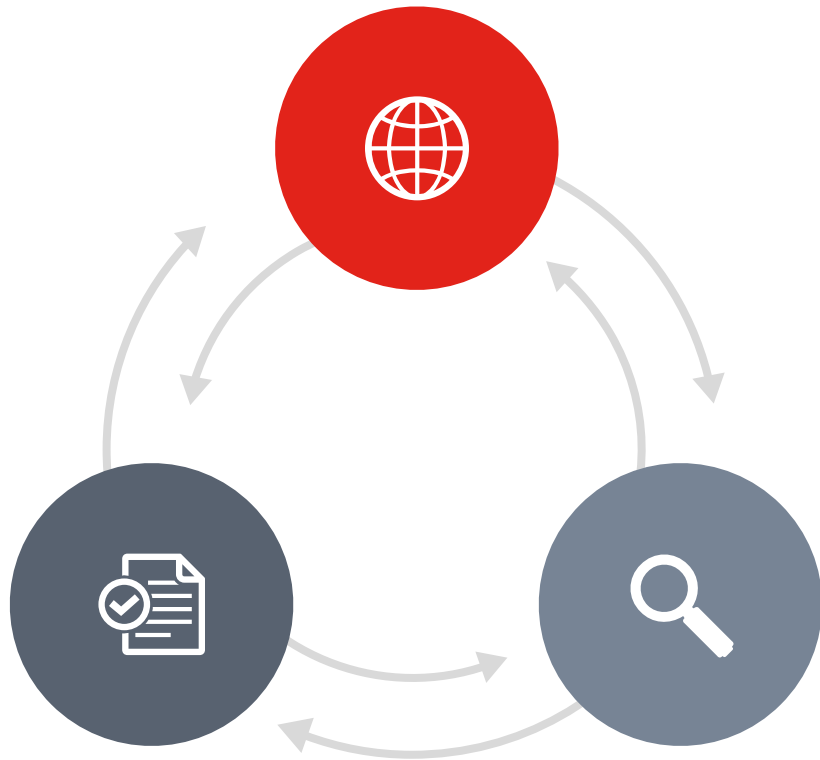
ERD



[DOCUMENT LINK](#)



Database Implementation



Create Database Online Shopping



Create Tables(20 tables)



Input Data(every table has 10 data)

Database Implementation – Computed Column

```
create database OnlineShoppingMall
```

```
USE OnlineShoppingMall
```

```
CREATE TABLE OrderHeader
```

```
(
```

```
    OrderID INT IDENTITY,
    OrderDate DATETIME,
    TotalAmount decimal(10,2) NOT NULL,
    Status VARCHAR(255) NOT NULL,
    UserID INT NOT NULL,
    Discount decimal(10,2),
    Tax decimal(10,2),
    PRIMARY KEY (OrderID),
    FOREIGN KEY (UserID) REFERENCES User_Account(User_id)
```

```
)
```

```
ALTER TABLE OrderHeader ALTER COLUMN Discount decimal(10,2) NOT NULL
```

```
ALTER TABLE OrderHeader ALTER COLUMN Tax decimal(10,2) NOT NULL
```

```
CREATE FUNCTION FU_TotalDue(@OrderID INT)
```

```
RETURNS DECIMAL(10,2)
```

```
AS
```

```
BEGIN
```

```
    DECLARE @TotalDue DECIMAL(10,2);
    SELECT @TotalDue = CAST( (TotalAmount + Tax - Discount) AS decimal(10,2))
    FROM OrderHeader
    WHERE OrderID = @OrderID
    RETURN @TotalDue
```

```
END
```

```
ALTER TABLE OrderHeader
```

```
ADD TotalDue AS (dbo.FU_TotalDue(OrderID));
```

	123 OrderID	🕒 OrderDate	123 TotalAmount	ABC Status	123 UserID	123 Discount	123 Tax	123 TotalDue
1	1	2018-11-01 00:00:00	207.50	p	1	0.00	21.00	228.50
2	2	2018-11-02 00:00:00	500.00	u	1	0.00	50.00	550.00
3	3	2018-11-02 00:00:00	500.00	u	2	55.00	50.00	495.00
4	4	2018-11-07 00:00:00	10.00	u	2	0.00	1.00	11.00
5	5	2018-11-07 00:00:00	39.00	p	4	0.00	4.00	43.00
6	6	2018-11-07 00:00:00	10.00	u	4	0.00	1.00	11.00
7	7	2018-11-08 00:00:00	1,000.00	u	4	55.00	100.00	1,045.00
8	8	2018-11-08 00:00:00	4,999.00	p	5	99.00	500.00	5,400.00
9	9	2018-11-10 00:00:00	60.00	p	3	0.00	6.00	66.00
10	10	2018-11-16 00:00:00	999.00	u	6	9.00	100.00	1,090.00
11	11	2016-11-16 00:00:00	999.00	u	6	9.00	100.00	1,090.00
12	12	2014-01-10 00:00:00	60.00	p	9	0.00	6.00	66.00
13	13	2016-11-16 00:00:00	999.00	u	7	9.00	100.00	1,090.00
14	14	2017-11-10 00:00:00	6,000.00	p	3	0.00	600.00	6,600.00
15	15	2014-01-10 00:00:00	60.00	p	8	0.00	6.00	66.00
16	16	2016-11-16 00:00:00	999.00	u	7	9.00	100.00	1,090.00
17	17	2016-11-10 00:00:00	1,600.00	p	10	0.00	160.00	1,760.00
18	18	2014-01-10 00:00:00	2,000.00	p	8	0.00	200.00	2,200.00
19	19	2018-11-16 00:00:00	999.00	u	9	9.00	100.00	1,090.00
20	20	2018-11-16 00:00:00	999.00	u	8	9.00	100.00	1,090.00



Database Implementation – Table-level Function

```
CREATE TABLE Payment
```

```
(  
    PaymentID INT IDENTITY,  
    TotalAmount decimal(10,2) NOT NULL,  
    Status VARCHAR(255) NOT NULL,  
    PayTime TIMESTAMP,  
    CardID BIGINT,  
    OrderID INT,  
    PRIMARY KEY (PaymentID),  
    FOREIGN KEY (OrderID) REFERENCES OrderHeader(OrderID).  
)
```

```
INSERT INTO Payment  
(TotalAmount, Status, CardID, OrderID)  
VALUES  
(2000000, 'Succeed', 10000000000000000, 1)
```



Query execution failed

原因:

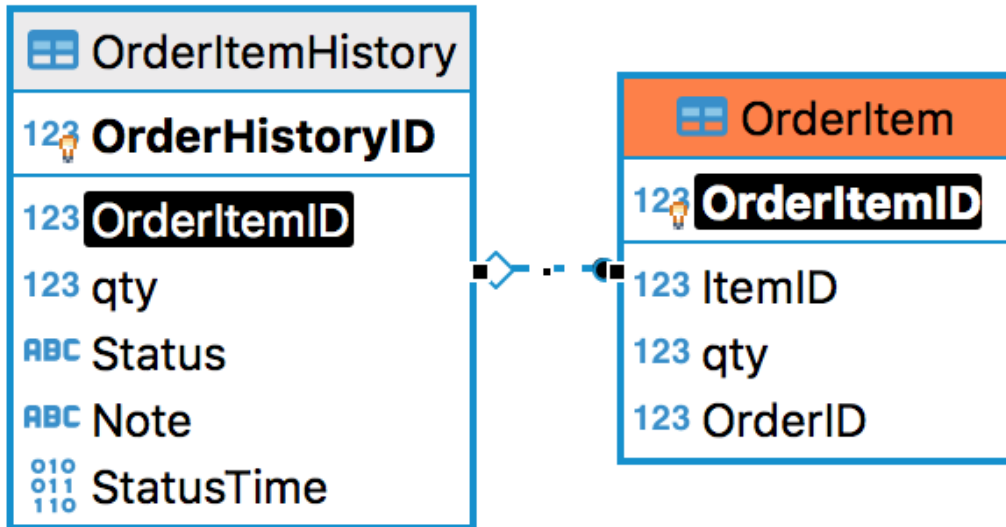
SQL 错误 [547] [23000]: The INSERT statement conflicted with the CHECK constraint "MaxTotalAmount". The conflict occurred in database "OnlineShoppingMall", table "dbo.Payment", column 'OrderID'.

```
BEGIN  
    DECLARE @flag BIT = 1;  
    IF (  
        SELECT ISNULL(SUM(TotalAmount), 0) FROM Payment WHERE @OrderID = OrderID  
    ) <= (SELECT TotalDue FROM OrderHeader WHERE @OrderID = OrderID)  
    BEGIN  
        SELECT @flag = 0  
    END  
    RETURN @flag  
END
```

```
ALTER TABLE Payment ADD CONSTRAINT MaxTotalAmount CHECK (dbo.FU_CheckTotalAmount(OrderID) = 0);
```



Database Implementation – Trigger



```
CREATE TRIGGER TR_OrderHis
ON OrderItem
AFTER INSERT
AS
BEGIN
    DECLARE @OrderItemID INT
    DECLARE @qty INT
    SELECT @OrderItemID = OrderItemID FROM Inserted
    SELECT @qty = qty FROM Inserted
    INSERT INTO OrderItemHistory
    (OrderItemID,qty,Status,Note)
VALUES
    (@OrderItemID,@qty,'purchased','')
END
```

Reporting View 1

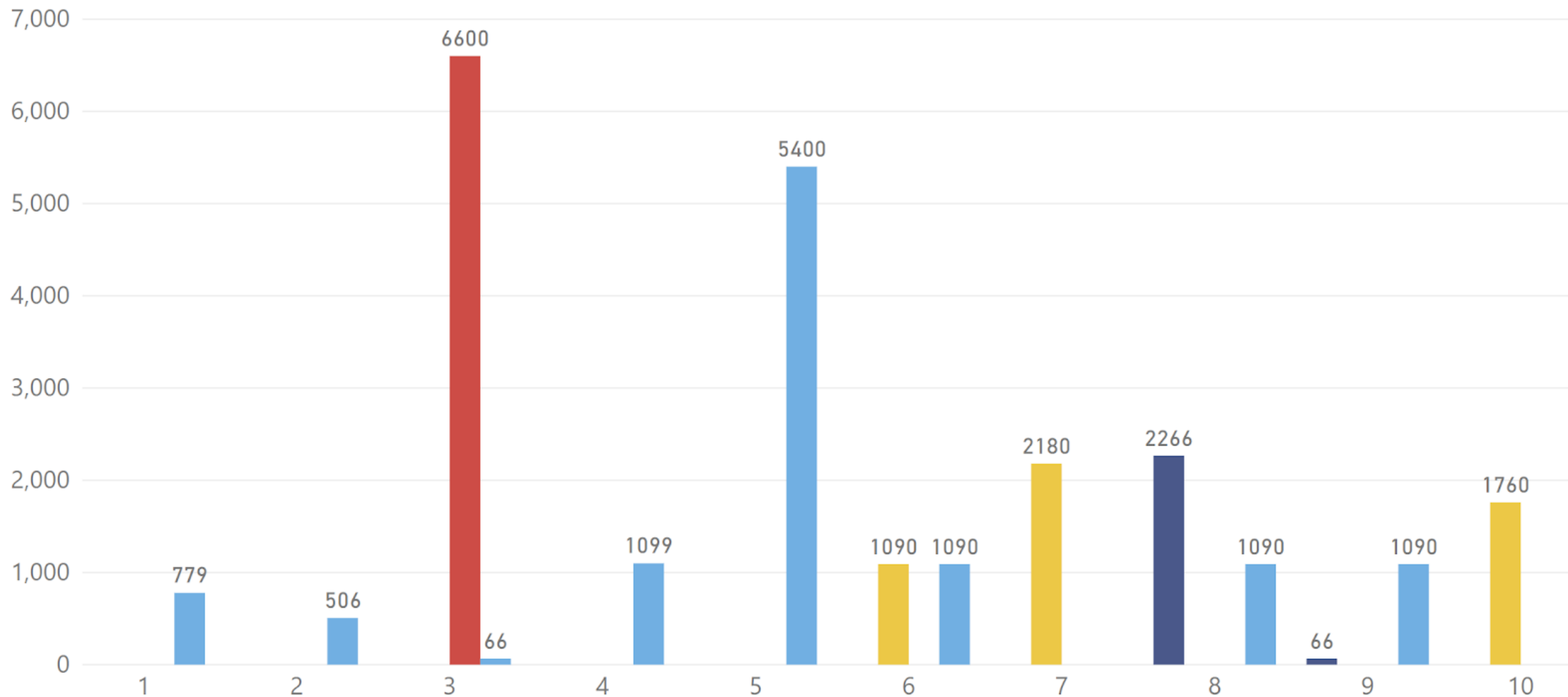
```
--View 1: Customer Purchase
CREATE VIEW CustomerPurchase_vw
AS
SELECT ua.User_id UserID, user_name,
       ISNULL(YEAR(OrderDate),0) PurchaseYear,
       ISNULL(SUM(TotalDue),0) TotalDue
FROM OrderHeader o
FULL JOIN User_Account ua
ON ua.User_id = o.UserID
GROUP BY ua.User_id, user_name, CUBE(YEAR(OrderDate))
```

	UserID	user_name	PurchaseYear	TotalDue
1	1	James	2018	778.50
2	1	James	0	778.50
3	2	Steve	2018	506.00
4	2	Steve	0	506.00
5	3	Amy	2017	6600.00
6	3	Amy	2018	66.00
7	3	Amy	0	6666.00
8	4	Mark	2018	1099.00
9	4	Mark	0	1099.00
10	5	Nicola	2018	5400.00
11	5	Nicola	0	5400.00
12	6	Elizabeth	2016	1090.00
13	6	Elizabeth	2018	1090.00
14	6	Elizabeth	0	2180.00
15	7	Susan	2016	2180.00
16	7	Susan	0	2180.00
17	8	Kelly	2014	2266.00
18	8	Kelly	2018	1090.00
19	8	Kelly	0	3356.00
20	9	David	2014	66.00
21	9	David	2018	1090.00
22	9	David	0	1156.00

Visualization Example 1

Customer Purchase by Year

PurchaseYear ● 2014 ● 2016 ● 2017 ● 2018



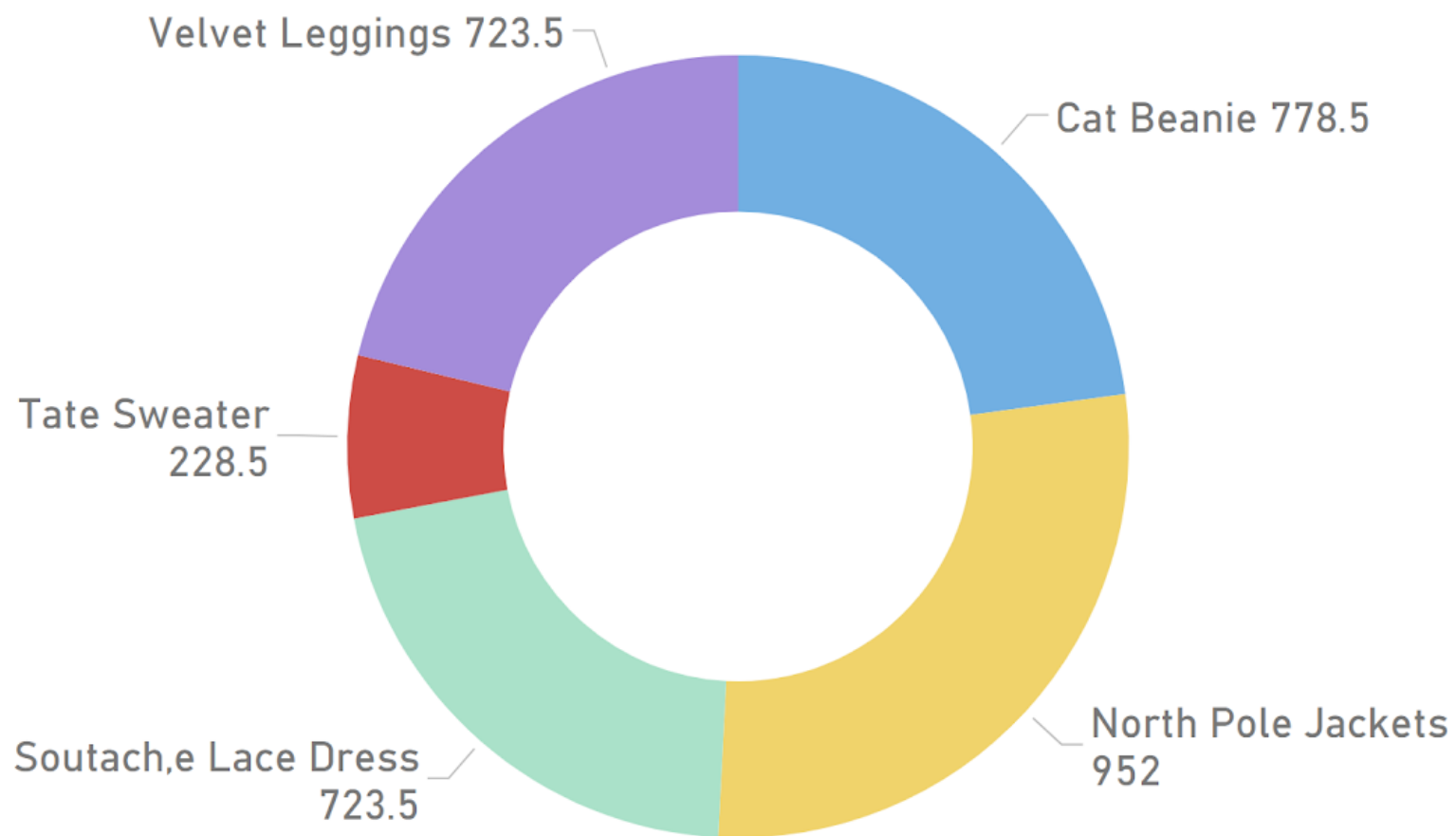
Reporting View 2

```
--View 2: Item Purchase
DROP VIEW ItemPurchase_vw
CREATE VIEW ItemPurchase_vw
AS
SELECT ii.item_id ItemID, item_name,price,
       ISNULL(YEAR(OrderDate),0) PurchaseYear,
       ISNULL(SUM(TotalDue),0) TotalDue
FROM OrderItem oi
FULL JOIN Inventory_Item ii
ON ii.item_id = oi.ItemID
FULL JOIN OrderHeader oh
ON oi.OrderID = oh.OrderID
GROUP BY CUBE(ii.item_id),item_name,price,YEAR(OrderDate)
```

Results		Messages			
	ItemID	item_name	price	PurchaseYear	TotalDue
1	1	Soutach, e Lace Dress	50.00	2018	723.50
2	2	North Pole Jackets	28.00	2018	952.00
3	4	Velvet Leggings	65.00	2018	723.50
4	5	Tate Sweater	120.00	2018	228.50
5	7	Cat Beanie	15.00	2018	778.50

Visualization Example 2

Sales Analysis by Item's Name(2018)



Item Name	Percentage
North Pole Jackets	27.95%
Cat Beanie	22.86%
Soutach,e Lace Dress	21.24%
Velvet Leggings	21.24%
Tate Sweater	6.71%

Reporting View 3

--View 3: Classify Customer

```
CREATE VIEW CustomerLevel_vw
```

```
AS
```

```
SELECT User_id UserID, user_name, trans_number,
```

```
CASE
```

```
    WHEN trans_number = 1 THEN 'normal'
```

```
    WHEN trans_number > 1 AND trans_number < 10 THEN 'silver'
```

```
    WHEN trans_number >= 10 AND trans_number <100 THEN 'gold'
```

```
    ELSE 'platinum'
```

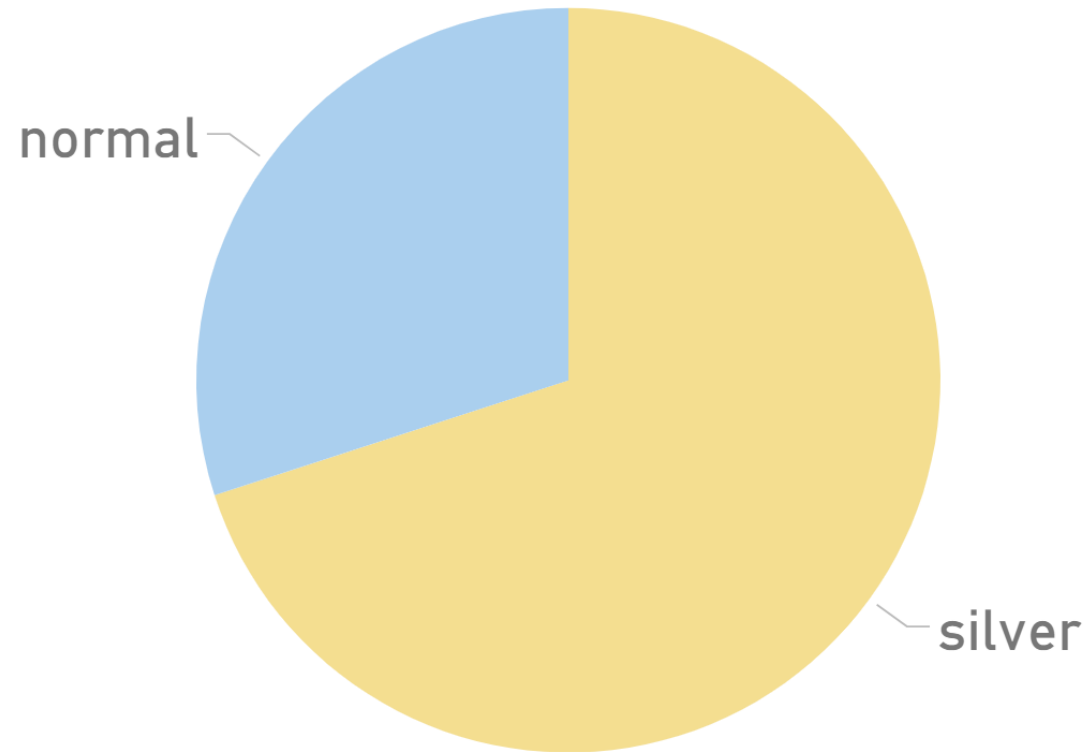
```
END AS customerLevel
```

```
FROM User_Account ua
```

	UserID	user_name	trans_number	customerLevel
1	1	James	5	silver
2	2	Steve	1	normal
3	3	Amy	1	normal
4	4	Mark	1	normal
5	5	Nicola	2	silver
6	6	Elizabeth	3	silver
7	7	Susan	2	silver
8	8	Kelly	4	silver
9	9	David	2	silver
10	10	Lucas	2	silver

Visualization Example 3

Customer Level



Customer Level	User Count
silver	7
normal	3



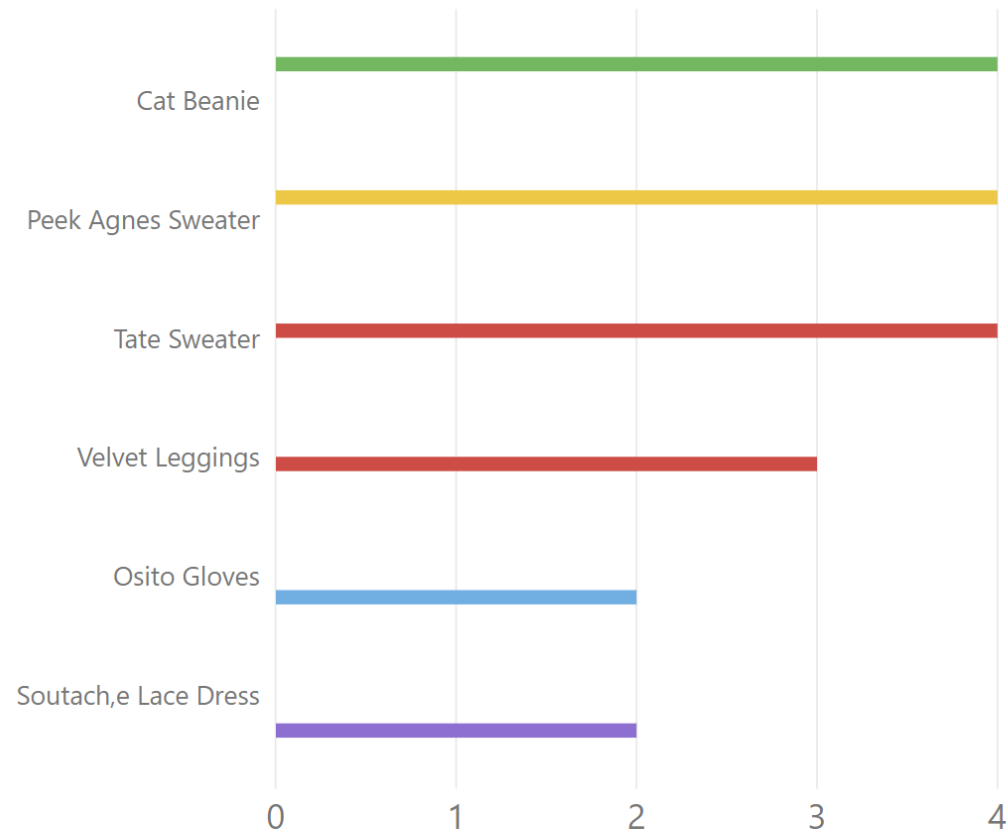
Reporting View 4

```
--View 4: Rating Item  
CREATE VIEW Item_Rating_vw  
AS  
SELECT Item_ID, AVG(Rating) Rating  
FROM Customer_Feedback  
GROUP BY Item_ID
```

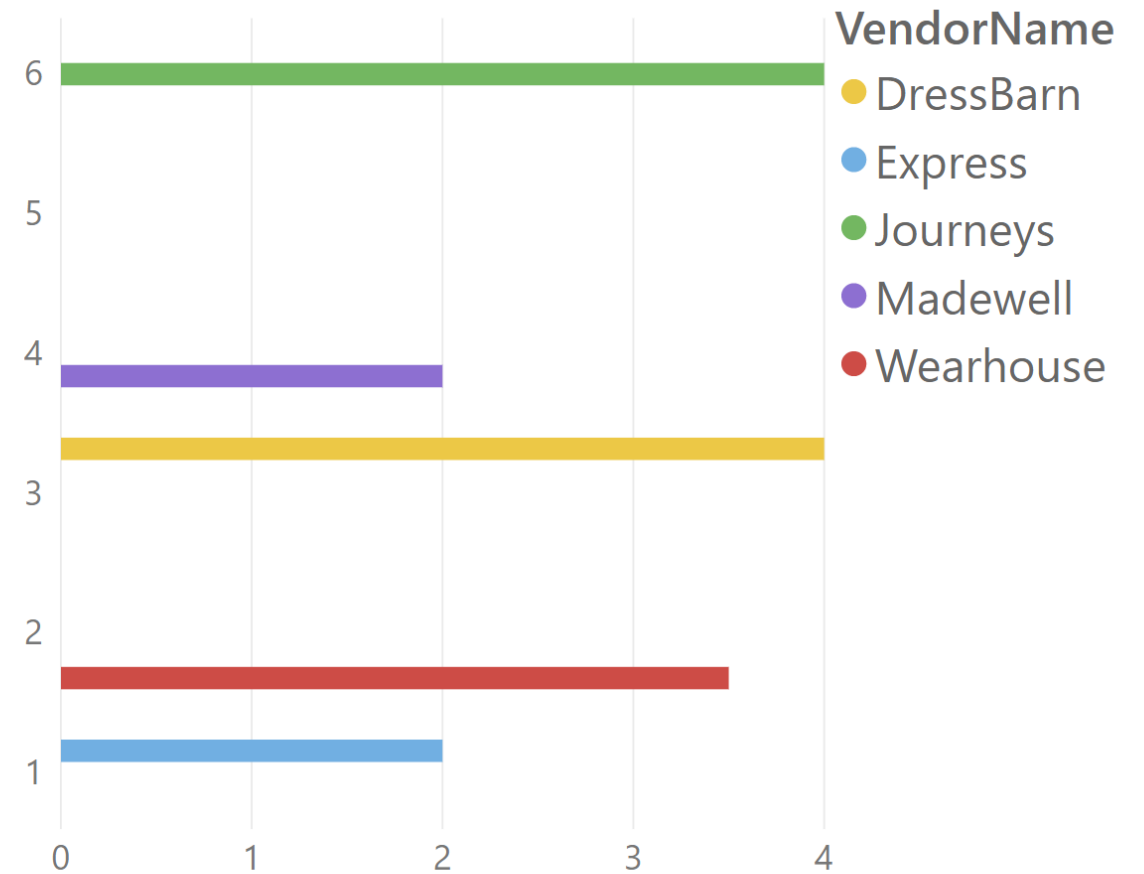
Results			Messages
	Item_ID	Rating	
1	1	2	
2	3	4	
3	4	3	
4	5	4	
5	7	4	
6	8	2	

Visualization Example 4

Item Rating



Vendor Rating



Conclusions

- 1 ▶ **We presented a database which records online shopping experience**
- 2 ▶ **From user information, the order history, inventory shipping details, to online advertisements, the database collects detailed information in analyzing customer shopping preferences**
- 3 ▶ **By analyzing data with SQL queries and presenting in Power BI, we can target specific users for advertisement, understand annual sales**



Thank you!
Q & A