计算机科学与技术学院_数据库系统_课程实验报告

实验题目：综合实验 1		学号：202300130183
日期：2024/11/29	班级：23 级人工智能	姓名：宋浩宇
Email：2367651943@qq.com		

实验软件和硬件环境：

实验软件：

系统：Windows 11 家庭中文版 23H2 22631.4317

编辑器：Visual Studio Code

数据库：SQL Server 2022

数据库管理工具：Microsoft SQL Server Management Studio 18

硬件环境：

CPU：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz

内存：32.0 GB (31.6 GB 可用)

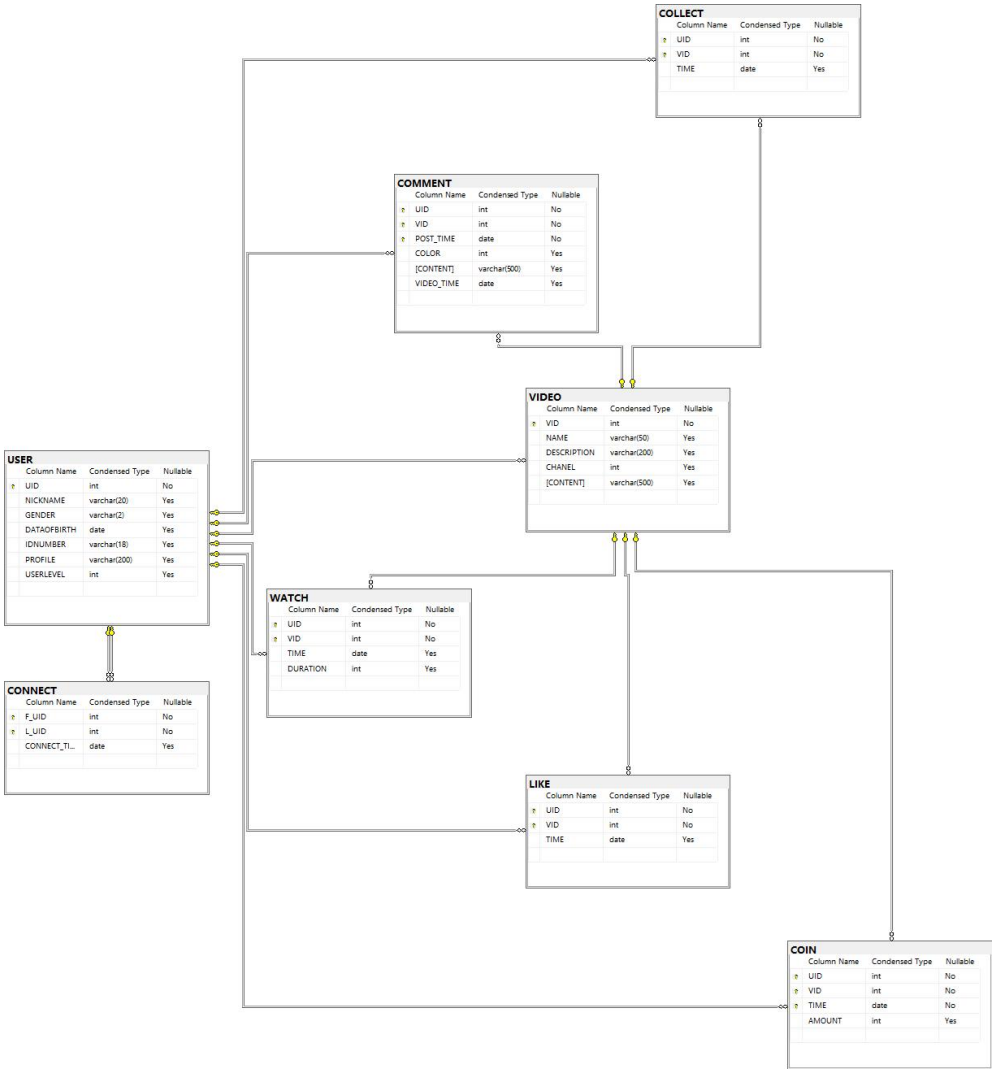
磁盘驱动器：NVMe WD_BLACKSN850X2000GB

实验步骤：

实验分为四个部分，每个部分分为模式图和 SQL，其中的模式图均是使用 Microsoft SQL Server Management Studio 18 中的 Database Diagram 工具生成的，SQL 语句均使用 Visual Studio Code 进行辅助高亮处理。

第一部分，哔哩哔哩：

模式图为：



SQL 为：

```
USE FOR_HOME_WORK;
GO


CREATE SCHEMA BILIBILI AUTHORIZATION "dbo";
GO
```

```
CREATE TABLE BILIBILI."USER"(
    UID INT PRIMARY KEY,
    NICKNAME VARCHAR(20),
    GENDER VARCHAR(2),
    DATAOFBIRTH DATE,
    IDNUMBER VARCHAR(18),
    PROFILE VARCHAR(200),
    USERLEVEL INT
);
GO
```

```
CREATE TABLE BILIBILI."CONNECT"(
    F_UID INT,
    L_UID INT,
    CONNECT_TIME DATE,
    PRIMARY KEY (F_UID, L_UID),
    FOREIGN KEY (F_UID) REFERENCES BILIBILI.[USER](UID),
    FOREIGN KEY (L_UID) REFERENCES BILIBILI.[USER](UID)
);
GO
```

```
CREATE TABLE BILIBILI."VIDEO"(
    VID INT PRIMARY KEY,
    NAME VARCHAR(50),
    DESCRIPTION VARCHAR(200),
    CHANEL INT,
    CONTENT VARCHAR(500),
    FOREIGN KEY (CHANEL) REFERENCES BILIBILI.[USER](UID)
);
GO
```

```
CREATE TABLE BILIBILI."COIN"(
    UID INT,
    VID INT,
    TIME DATE,
    AMOUNT INT,
    PRIMARY KEY (UID, VID, TIME),
    FOREIGN KEY (UID) REFERENCES BILIBILI.[USER](UID),
    FOREIGN KEY (VID) REFERENCES BILIBILI.VIDEO(VID)
);
GO
```

```
CREATE TABLE BILIBILI."COMMENT"(
    UID INT,
    VID INT,
    POST_TIME DATE,
    COLOR INT,
    CONTENT VARCHAR(500),
    VIDEO_TIME DATE,
    PRIMARY KEY (UID, VID, POST_TIME),
    FOREIGN KEY (UID) REFERENCES BILIBILI.[USER](UID),
    FOREIGN KEY (VID) REFERENCES BILIBILI.VIDEO(VID)
);
GO
```

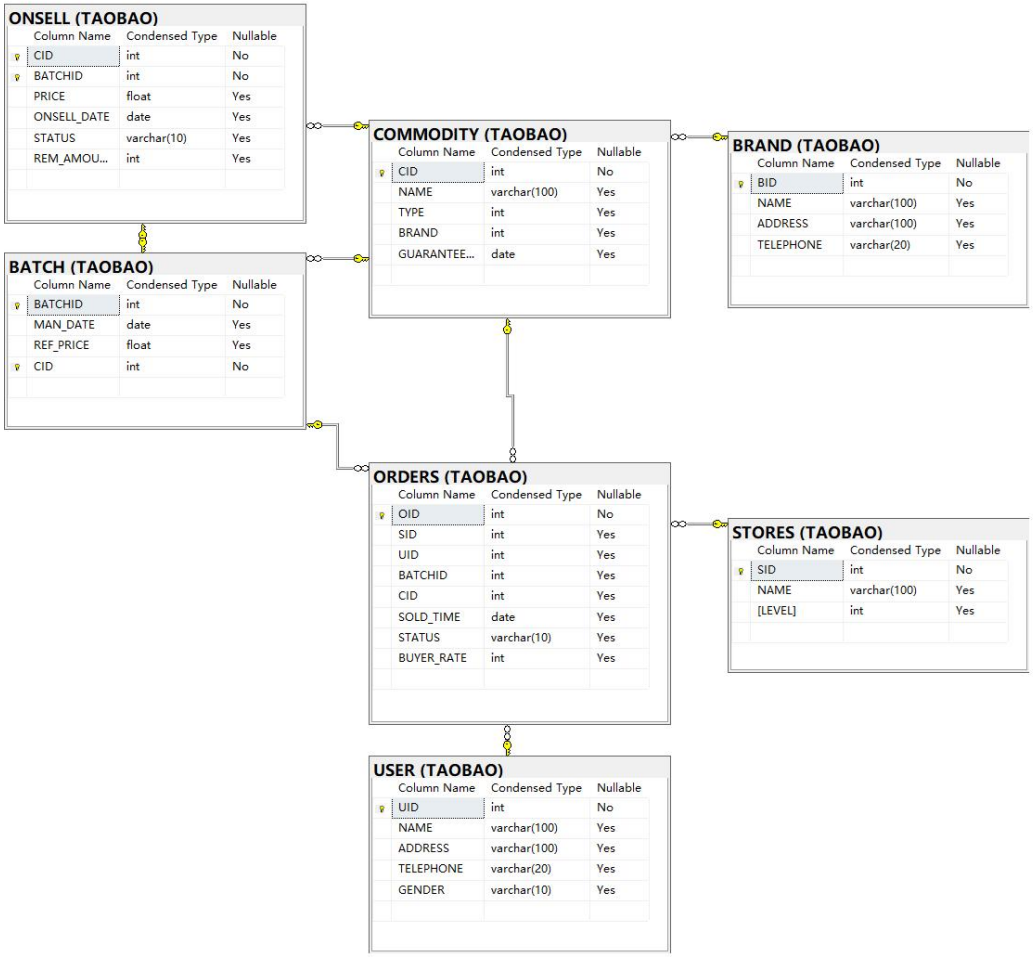
```
CREATE TABLE BILIBILI."LIKE"(
    UID INT,
    VID INT,
    TIME DATE,
    PRIMARY KEY (UID, VID),
    FOREIGN KEY (UID) REFERENCES BILIBILI.[USER](UID),
    FOREIGN KEY (VID) REFERENCES BILIBILI.VIDEO(VID)
);
GO
```

```
CREATE TABLE BILIBILI."COLLECT"(
    UID INT,
    VID INT,
    TIME DATE,
    PRIMARY KEY (UID, VID),
    FOREIGN KEY (UID) REFERENCES BILIBILI.[USER](UID),
    FOREIGN KEY (VID) REFERENCES BILIBILI.VIDEO(VID)
);
GO
```

```
CREATE TABLE BILIBILI."WATCH"(
    UID INT,
    VID INT,
    TIME DATE,
    DURATION INT,
    PRIMARY KEY (UID, VID),
    FOREIGN KEY (UID) REFERENCES BILIBILI.[USER](UID),
    FOREIGN KEY (VID) REFERENCES BILIBILI.VIDEO(VID)
);
GO
```

第二部分，淘宝：

模式图为：



SQL 为：

```
USE FOR_HOME_WORK;
GO


CREATE SCHEMA [TAOBAO] AUTHORIZATION DBO;
GO
```

```
CREATE TABLE TAOBAO.[BRAND] (
    BID INT PRIMARY KEY,
    NAME VARCHAR(100),
    ADDRESS VARCHAR(100),
    TELEPHONE VARCHAR(20)
);
GO
```

```
CREATE TABLE TAOBAO.[COMMODITY] (
    CID INT PRIMARY KEY,
    NAME VARCHAR(100),
    TYPE INT,
    BRAND INT,
    GUARANTEE_PERIOD DATE,
    FOREIGN KEY (BRAND) REFERENCES TAOBAO.BRAND(BID)
);
GO
```

```
CREATE TABLE TAOBAO.[BATCH](
    BATCHID INT,
    MAN_DATE DATE,
    REF_PRICE FLOAT,
    CID INT,
    PRIMARY KEY(CID,BATCHID),
    FOREIGN KEY(CID) REFERENCES TAOBAO.COMMODITY(CID)
);
GO
```

```
CREATE TABLE TAOBAO.[STORES](
    SID INT PRIMARY KEY,
    NAME VARCHAR(100),
    LEVEL INT
);
GO
```

```
CREATE TABLE TAOBAO.[USER](
    UID INT PRIMARY KEY,
    NAME VARCHAR(100),
    ADDRESS VARCHAR(100),
    TELEPHONE VARCHAR(20),
    GENDER VARCHAR(10)
);
GO
```

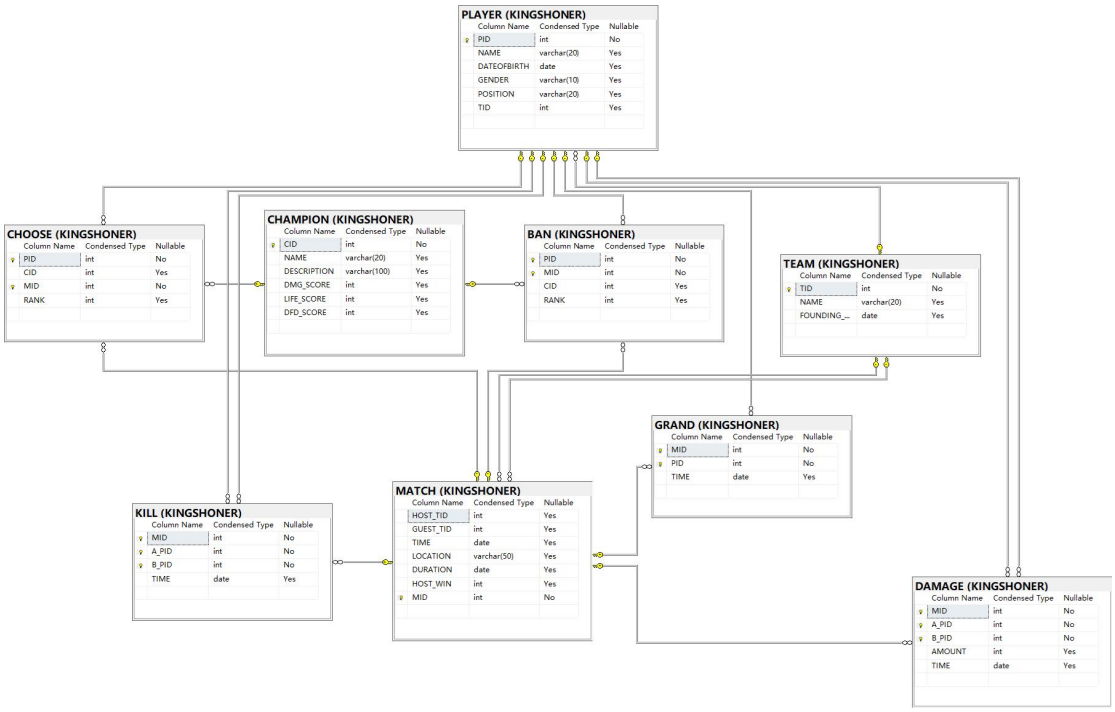
```
CREATE TABLE TAOBAO.[ONSELL](
    CID INT,
    BATCHID INT,
    PRICE FLOAT,
    ONSELL_DATE DATE,
    STATUS VARCHAR(10),
    REM_AMOUNT INT,
```

```
    PRIMARY KEY(CID, BATCHID),
    FOREIGN KEY(CID) REFERENCES TAOBAO.COMMODITY(CID),
    FOREIGN KEY(CID,BATCHID) REFERENCES TAOBAO.BATCH(CID,BATCHID)
);
GO
```

```
CREATE TABLE TAOBAO.[ORDERS](
    OID INT PRIMARY KEY,
    SID INT,
    UID INT,
    BATCHID INT,
    CID INT,
    SOLD_TIME DATE,
    STATUS VARCHAR(10),
    BUYER_RATE INT NULL,
    FOREIGN KEY(UID) REFERENCES TAOBAO.[USER](UID),
    FOREIGN KEY(CID, BATCHID) REFERENCES TAOBAO.BATCH(CID, BATCHID),
    FOREIGN KEY(CID) REFERENCES TAOBAO.COMMODITY(CID),
    FOREIGN KEY(SID) REFERENCES TAOBAO.STORES(SID)
);
GO
```

第三部分，王者荣耀联赛：

模式图为：



SQL 为：

```
USE FOR_HOME_WORK;
GO


CREATE SCHEMA [KINGSHONER] AUTHORIZATION DBO;
GO
```

```
CREATE TABLE KINGSHONER.[TEAM](
    TID INT PRIMARY KEY,
    NAME VARCHAR(20),
    FOUNDING_DATE DATE
);
GO
```

```
CREATE TABLE KINGSHONER.[PLAYER](
    PID INT PRIMARY KEY,
    NAME VARCHAR(20),
    DATEOFBIRTH DATE,
    GENDER VARCHAR(10),
    POSITION VARCHAR(20),
    TID INT,
    FOREIGN KEY (TID) REFERENCES KINGSHONER.[TEAM](TID)
);
GO
```

```
CREATE TABLE KINGSHONER.[MATCH](
    HOST_TID INT,
    GUEST_TID INT,
    TIME DATE,
    LOCATION VARCHAR(50),
    DURATION DATE,
    HOST_WIN INT,
    MID INT PRIMARY KEY,
    FOREIGN KEY (HOST_TID) REFERENCES KINGSHONER.[TEAM](TID),
    FOREIGN KEY (GUEST_TID) REFERENCES KINGSHONER.[TEAM](TID)
);
GO
```

```
CREATE TABLE KINGSHONER.[CHAMPION](
    CID INT PRIMARY KEY,
    NAME VARCHAR(20),
    DESCRIPTION VARCHAR(100),
    DMG_SCORE INT,
    LIFE_SCORE INT,
    DFD_SCORE INT
```

```
);
GO
```

```
CREATE TABLE KINGSHONER.[CHOOSE](
    PID INT,
    CID INT,
    MID INT,
    RANK INT,
    PRIMARY KEY (PID, MID),
    FOREIGN KEY (PID) REFERENCES KINGSHONER.[PLAYER](PID),
    FOREIGN KEY (CID) REFERENCES KINGSHONER.[CHAMPION](CID),
    FOREIGN KEY (MID) REFERENCES KINGSHONER.[MATCH](MID)
);
GO
```

```
CREATE TABLE KINGSHONER.[BAN](
    PID INT,
    MID INT,
    CID INT,
    RANK INT,
    FOREIGN KEY (PID) REFERENCES KINGSHONER.[PLAYER](PID),
    FOREIGN KEY (MID) REFERENCES KINGSHONER.[MATCH](MID),
    FOREIGN KEY (CID) REFERENCES KINGSHONER.[CHAMPION](CID),
    PRIMARY KEY (PID, MID)
);
GO
```

```
CREATE TABLE KINGSHONER.[DAMAGE](
    MID INT,
    A_PID INT,
    B_PID INT,
    AMOUNT INT,
    TIME DATE FOREIGN KEY (MID) REFERENCES KINGSHONER.[MATCH](MID),
    FOREIGN KEY (A_PID) REFERENCES KINGSHONER.[PLAYER](PID),
    FOREIGN KEY (B_PID) REFERENCES KINGSHONER.[PLAYER](PID),
    PRIMARY KEY (MID, A_PID, B_PID)
);
GO
```

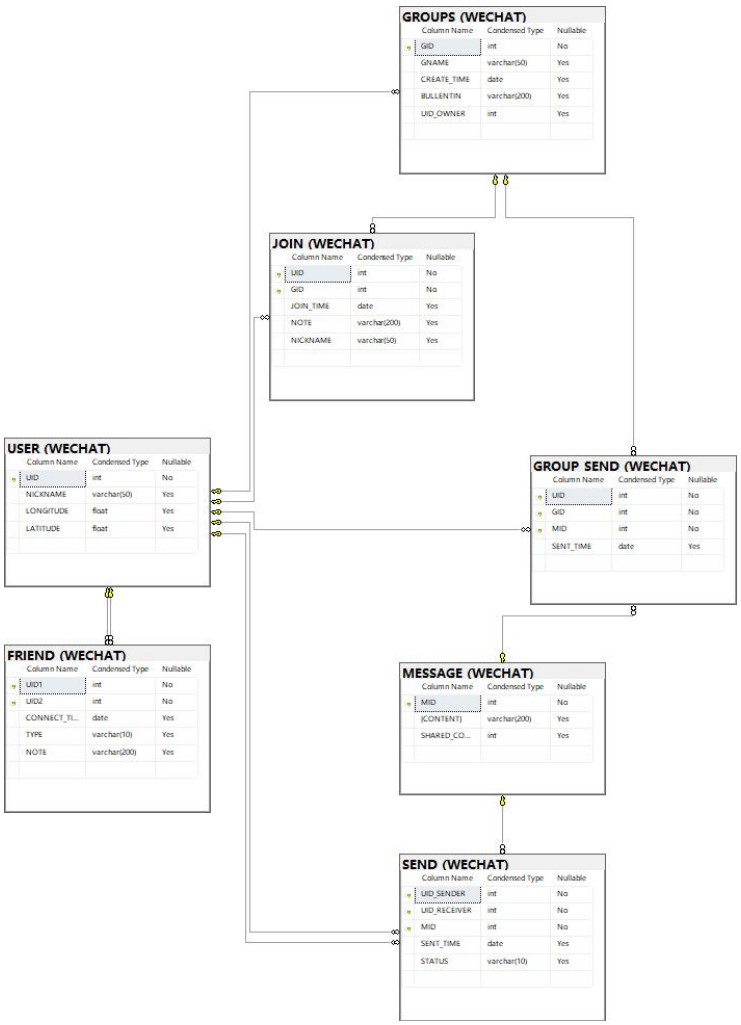
```
CREATE TABLE KINGSHONER.[KILL](
    MID INT,
    A_PID INT,
    B_PID INT,
    TIME DATE,
```

```
    FOREIGN KEY (MID) REFERENCES KINGSHONER.[MATCH](MID),
    FOREIGN KEY (A_PID) REFERENCES KINGSHONER.[PLAYER](PID),
    FOREIGN KEY (B_PID) REFERENCES KINGSHONER.[PLAYER](PID),
    PRIMARY KEY (MID, A_PID, B_PID)
);
GO
```

```
CREATE TABLE KINGSHONER.[GRAND](
    MID INT,
    PID INT,
    TIME DATE,
    FOREIGN KEY (MID) REFERENCES KINGSHONER.[MATCH](MID),
    FOREIGN KEY (PID) REFERENCES KINGSHONER.[PLAYER](PID),
    PRIMARY KEY (MID, PID)
);
GO
```

第四部分，微信：

模式图为：



SQL 为：

```
USE FOR_HOME_WORK;
GO

CREATE SCHEMA [WECHAT] AUTHORIZATION DBO;
GO
```

```
CREATE TABLE WECHAT.[USER](
    UID INT PRIMARY KEY,
    NICKNAME VARCHAR(50),
    LONGITUDE FLOAT,
    LATITUDE FLOAT
);
GO
```

```
CREATE TABLE WECHAT.[FRIEND](
    UID1 INT,
    UID2 INT,
    CONNECT_TIME DATE,
    TYPE VARCHAR(10),
    NOTE VARCHAR(200),
    PRIMARY KEY(UID1, UID2),
    FOREIGN KEY(UID1) REFERENCES WECHAT.[USER](UID),
    FOREIGN KEY(UID2) REFERENCES WECHAT.[USER](UID)
);
GO
```

```
CREATE TABLE WECHAT.[MESSAGE](
    MID INT PRIMARY KEY,
    CONTENT VARCHAR(200),
    SHARED_COUNT INT DEFAULT 1,
);
GO
```

```
CREATE TABLE WECHAT.[SEND](
    UID_SENDER INT,
    UID_RECEIVER INT,
    MID INT,
    SENT_TIME DATE,
    STATUS VARCHAR(10),
    PRIMARY KEY(UID_SENDER, UID_RECEIVER, MID),
    FOREIGN KEY(UID_SENDER) REFERENCES WECHAT.[USER](UID),
    FOREIGN KEY(UID_RECEIVER) REFERENCES WECHAT.[USER](UID),
    FOREIGN KEY(MID) REFERENCES WECHAT.[MESSAGE](MID)
);
```

```
GO
```

```
create trigger trg_UpdateSharedCount
on Wechat.[send]
AFTER INSERT
AS
BEGIN
    UPDATE Wechat.[message]
    SET SHARED_COUNT = SHARED_COUNT + 1
    WHERE MID IN (SELECT MID FROM inserted)
END;
GO
```

```
CREATE TABLE WECHAT.[GROUPS](
    GID INT PRIMARY KEY,
    GNAME VARCHAR(50),
    CREATE_TIME DATE,
    BULLENTIN VARCHAR(200),
    UID_OWNER INT,
    FOREIGN KEY(UID_OWNER) REFERENCES WECHAT.[USER](UID)
);
GO
```

```
CREATE TABLE WECHAT.[JOIN](
    UID INT,
    GID INT,
    JOIN_TIME DATE,
    NOTE VARCHAR(200),
    NICKNAME VARCHAR(50),
    PRIMARY KEY(UID, GID),
    FOREIGN KEY(UID) REFERENCES WECHAT.[USER](UID),
    FOREIGN KEY(GID) REFERENCES WECHAT.[GROUPS](GID)
);
GO
```

```
CREATE TABLE WECHAT.[GROUP_SEND](
    UID INT,
    GID INT,
    MID INT,
    SENT_TIME DATE,
    PRIMARY KEY(UID, GID, MID),
    FOREIGN KEY(UID) REFERENCES WECHAT.[USER](UID),
    FOREIGN KEY(GID) REFERENCES WECHAT.[GROUPS](GID),
    FOREIGN KEY(MID) REFERENCES WECHAT.[MESSAGE](MID)
```

```
);
GO
```

```
CREATE TRIGGER TRG_UpdateGroupMessageCount
ON WECHAT.[GROUP_SEND]
AFTER INSERT
AS
BEGIN
    UPDATE WECHAT.[GROUPS]
    SET SHARED_COUNT = SHARED_COUNT + 1
    WHERE MID IN (SELECT MID FROM inserted)
END;
```

```
CREATE TABLE WECHAT.[MOMENT](
    MOID INT PRIMARY KEY,
    UID INT,
    CONTENT VARCHAR(200),
    POST_TIME DATE,
    TYPE VARCHAR(10),
    FOREIGN KEY(UID) REFERENCES WECHAT.[USER](UID)
);
GO
```