IE304 – Production and Service Information Systems

Project 1 – Part 2

"Academic integrity is expected of all students of METU at all times, whether in the presence or absence of members of the faculty. Understanding this, I declare that I shall not give, use, or receive unauthorized aid in this study."

Abderrahman Harkat	2397479
Alperen Oktay Şahin	2305373
Atakan Berk	2444438
Enes Gök	2444628
Islam Valehli	2349371
Youssef Nsouli	2487494

10/05/2023

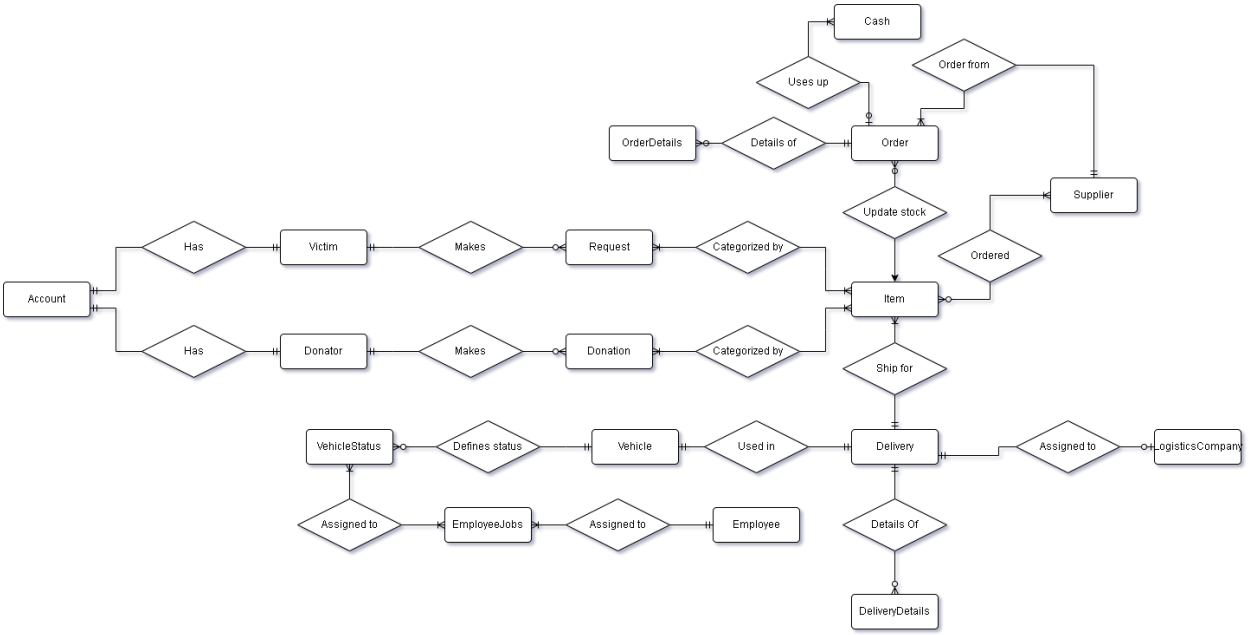

Figure 1: EER Diagram explaining the relationships between entities

Required Tables: (Primary keys in **Bold**)

Victim:

RequesterID

AccountID

Name

Surname

DistrictID

Request:

RequestID

RequesterID

RequestTime

DeliveryTime

Status

RequestedItemID

RequestedItemAmount

Feedback

Donator:

DonatorID

AccountID

Name

Surname

DistrictID

Donation:

DonationID

RequestID

DonatedItemID

DonatedAmount

DonatedCurrencyID

Item:

ItemID

ItemName

AmountAvailable

Cash:

CurrencyID

CurrencyName

RateToTL

AmountAvailable

Supplier:

SupplierID

SupplierName

DistrictID

Order:

OrderID

SupplierID

OrderPrice

OrderDetails:

OrderID

ProductID

SupplierID

AmountOrdered

Employee:

EmployeeID

EmployeeName

EmployeeSurname

EmployeeJobs:

EmployeeID

AssignedVehicle

Status

VehicleInfo:

VehicleID

VehicleName

VehicleModel

LicenseNo

VehicleStatus:

VehicleID

Status

CourierID

Delivery:

DeliveryID

SupplierID

RequesterID

LogisticsCompany

DeliveryTime

EmployeeID

Price

DeliveryDetails:

DeliveryID

ItemID

ItemAmount

LogisticsCompany:

CompanyID

CompanyName

DistrictID

Price

District:

DistrictID

DistrictName

PostalCode

Region

Street

Building

Account:

AccountID

Username

Password

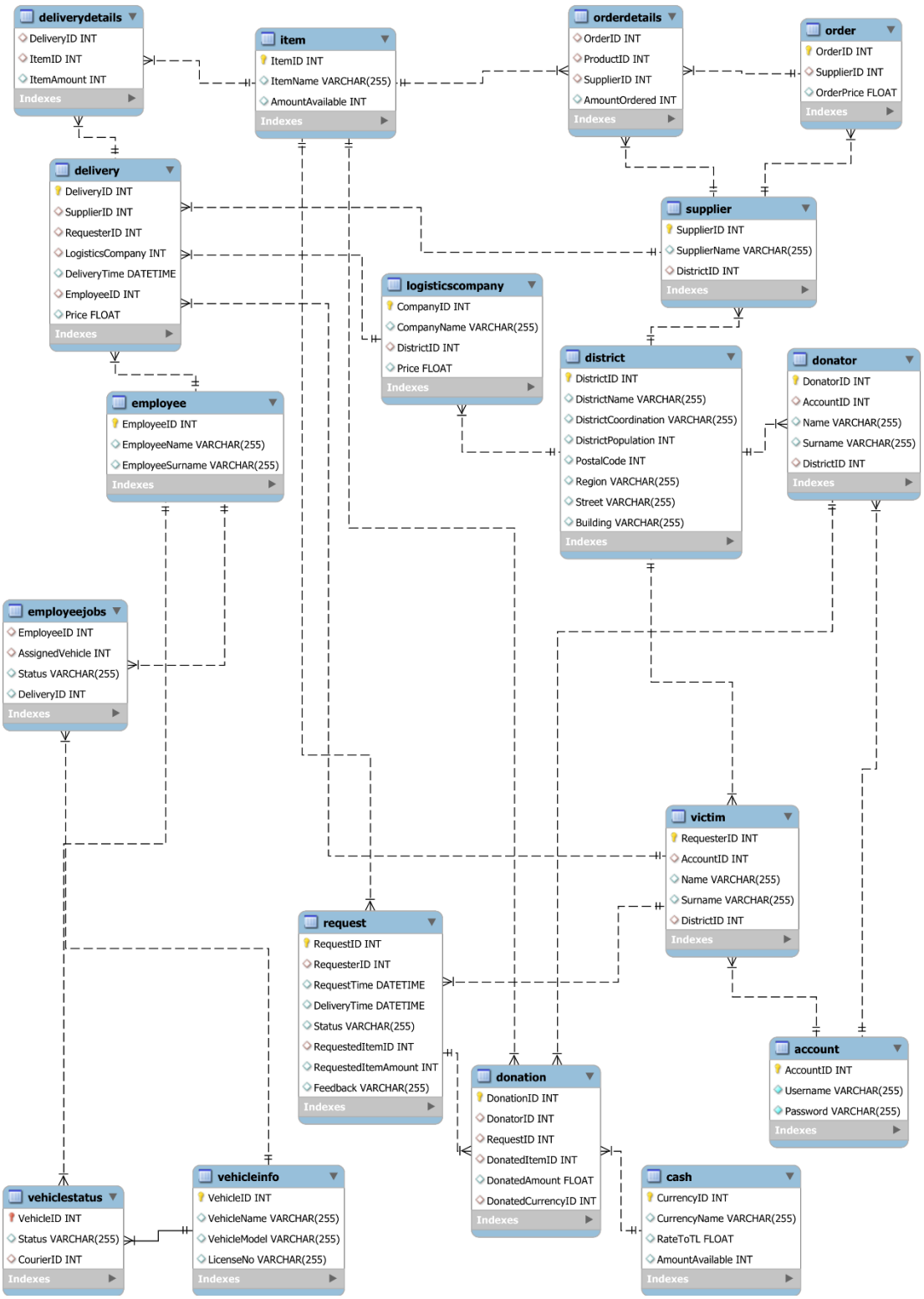

Figure 2: Relational Diagram of S.O.S.'s Database

SQL Query Code:

```sql
DROP DATABASE IF EXISTS `SOS IS`;
CREATE DATABASE IF NOT EXISTS `SOS IS`;
USE `SOS IS`;

CREATE TABLE IF NOT EXISTS `Account` (
  AccountID INT NOT NULL AUTO_INCREMENT,
  Username VARCHAR(255) NOT NULL,
  `Password` VARCHAR(255) NOT NULL,
  PRIMARY KEY (AccountID)
);

CREATE TABLE IF NOT EXISTS District (
  DistrictID INT NOT NULL,
  DistrictName VARCHAR(255),
  DistrictCoordination VARCHAR(255),
  DistrictPopulation INT,
  PostalCode INT,
  Region VARCHAR(255),
  Street VARCHAR(255),
  Building VARCHAR(255),
  PRIMARY KEY (DistrictID)
);

CREATE TABLE IF NOT EXISTS Item (
  ItemID INT NOT NULL,
  ItemName VARCHAR(255),
  AmountAvailable INT,
  PRIMARY KEY (ItemID)
);

CREATE TABLE IF NOT EXISTS Cash (
  CurrencyID INT NOT NULL,
  CurrencyName VARCHAR(255),
  RateToTL FLOAT,
  AmountAvailable INT,
  PRIMARY KEY (CurrencyID)
);

CREATE TABLE IF NOT EXISTS Victim (
  RequesterID INT NOT NULL,
  AccountID INT,
  `Name` VARCHAR(255),
  Surname VARCHAR(255),
  DistrictID INT,
  PRIMARY KEY (RequesterID),
  FOREIGN KEY (AccountID) REFERENCES Account(AccountID),
  FOREIGN KEY (DistrictID) REFERENCES District(DistrictID)
);

CREATE TABLE IF NOT EXISTS Request (
  RequestID INT NOT NULL,
  RequesterID INT,
```

```
  RequestTime DATETIME,
  DeliveryTime DATETIME,
  `Status` VARCHAR(255),
  RequestedItemID INT,
  RequestedItemAmount INT,
  Feedback VARCHAR(255),
  PRIMARY KEY (RequestID),
  FOREIGN KEY (RequesterID) REFERENCES Victim(RequesterID),
  FOREIGN KEY (RequestedItemID) REFERENCES Item(ItemID)
);

CREATE TABLE IF NOT EXISTS Donator (
  DonatorID INT NOT NULL,
  AccountID INT,
  `Name` VARCHAR(255),
  Surname VARCHAR(255),
  DistrictID INT,
  PRIMARY KEY (DonatorID),
  FOREIGN KEY (AccountID) REFERENCES Account(AccountID),
  FOREIGN KEY (DistrictID) REFERENCES District(DistrictID)
);

CREATE TABLE IF NOT EXISTS Donation (
  DonationID INT NOT NULL,
  DonatorID INT,
  RequestID INT,
  DonatedItemID INT,
  DonatedAmount FLOAT,
  DonatedCurrencyID INT,
  PRIMARY KEY (DonationID),
  FOREIGN KEY (RequestID) REFERENCES Request(RequestID),
  FOREIGN KEY (DonatedItemID) REFERENCES Item(ItemID),
  FOREIGN KEY (DonatedCurrencyID) REFERENCES Cash(CurrencyID),
  FOREIGN KEY (DonatorID) REFERENCES Donator(DonatorID)
);

CREATE TABLE IF NOT EXISTS Supplier (
  SupplierID INT NOT NULL,
  SupplierName VARCHAR(255),
  DistrictID INT,
  PRIMARY KEY (SupplierID),
  FOREIGN KEY (DistrictID) REFERENCES District(DistrictID)
);

CREATE TABLE IF NOT EXISTS `Order` (
  OrderID INT NOT NULL,
  SupplierID INT,
  OrderPrice FLOAT,
  PRIMARY KEY (OrderID),
  FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)
);

CREATE TABLE IF NOT EXISTS OrderDetails (
  OrderID INT,
  ProductID INT,
  SupplierID INT,
  AmountOrdered INT,
```

```
  FOREIGN KEY (OrderID) REFERENCES `Order`(OrderID),
  FOREIGN KEY (ProductID) REFERENCES Item(ItemID),
  FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)
);

CREATE TABLE IF NOT EXISTS VehicleInfo (
  VehicleID INT NOT NULL,
  VehicleName VARCHAR(255),
  VehicleModel VARCHAR(255),
  LicenseNo VARCHAR(255),
  PRIMARY KEY (VehicleID)
);

CREATE TABLE IF NOT EXISTS Employee (
  EmployeeID INT NOT NULL,
  EmployeeName VARCHAR(255),
  EmployeeSurname VARCHAR(255),
  EmployeeJobs VARCHAR(255),
  AssignedVehicle INT,
  `Status` VARCHAR(255),
  PRIMARY KEY (EmployeeID),
  FOREIGN KEY (AssignedVehicle) REFERENCES VehicleInfo(VehicleID)
);

CREATE TABLE IF NOT EXISTS EmployeeJobs (
EmployeeID INT,
AssignedVehicle INT,
`Status` VARCHAR(255),
DeliveryID INT,
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

CREATE TABLE IF NOT EXISTS VehicleStatus (
  VehicleID INT,
  `Status` VARCHAR(255),
  CourierID INT,
  PRIMARY KEY (VehicleID),
  FOREIGN KEY (VehicleID) REFERENCES VehicleInfo(VehicleID),
  FOREIGN KEY (CourierID) REFERENCES Employee(EmployeeID)
);

CREATE TABLE IF NOT EXISTS LogisticsCompany (
  CompanyID INT NOT NULL,
  CompanyName VARCHAR(255),
  DistrictID INT,
  Price FLOAT,
  PRIMARY KEY (CompanyID),
  FOREIGN KEY (DistrictID) REFERENCES District(DistrictID)
);

CREATE TABLE IF NOT EXISTS Delivery (
  DeliveryID INT NOT NULL,
  SupplierID INT,
  RequesterID INT,
  LogisticsCompany INT,
  DeliveryTime DATETIME,
  EmployeeID INT,
```

```
  Price FLOAT,
  PRIMARY KEY (DeliveryID),
  FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID),
  FOREIGN KEY (RequesterID) REFERENCES Victim(RequesterID),
  FOREIGN KEY (LogisticsCompany) REFERENCES LogisticsCompany(CompanyID),
  FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

CREATE TABLE IF NOT EXISTS DeliveryDetails (
  DeliveryID INT,
  ItemID INT,
  ItemAmount INT,
  PRIMARY KEY (DeliveryID, ItemID),
  FOREIGN KEY (DeliveryID) REFERENCES Delivery(DeliveryID),
  FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
);
```