**SMARTPROC**

**PROCUREMENT INVENTORY MANAGEMENT SYSTEM**

A picture containing text

Description automatically generated

**Project by:**

1. *Pranali Shenvi*
2. *Priyal Bamnikar*
3. *Srushti Gawade*

|  |  |  |
| --- | --- | --- |
| **Table of Contents** | | |
| Sr. no. | Topic: | Page No. |
| 1. | Background |  |
| 2. | Problem |  |
| 3. | Solution |  |
| 4. | Data Dictionary |  |
| 5. | Business Rules |  |
| 6. | Database System Infrastructure |  |
| 7. | Table Creation |  |
| 8. | Insertion of sample data |  |
| 9. | Answering business questions |  |
| 10. | Logbook |  |

**Background:**

**Problem:**

**Solution:**

**Conceptual Diagram:**

Diagram

Description automatically generated

**Logical Diagram:**

Diagram, schematic

Description automatically generated

**Data Dictionary:**

1. Addresses: *This table contains Stakeholder Project addresses*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Address\_ID | int | Not Null |  | Primary key of addresses table |
|  | Street\_Address\_1 | varchar(50) | Not Null |  | Street address of stakeholder project |
|  | Street\_Address\_2 | varchar(50) | Not Null |  | Secondary address line of stakeholder project |
|  | City | varchar(20) | Not Null |  | City corresponding to the address |
|  | States | varchar(20) | Not Null |  | State corresponding to the address |
|  | Zip | int | Not Null |  | Zip corresponding to the address |

1. CreditCards: *This table contains Credit card details of Project Managers*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Credit\_card\_id | int | Not Null |  | Primary key of credit cards table |
| Unique | Credit\_card\_no | int | Not Null |  | Credit card number of project manager |
|  | Expiration\_Month | int | Not Null |  | Expiration month on card |
|  | Expiration\_Year | int | Not Null |  | Expiration Year on card |
|  | Cardholder\_Name | varchar(30) | Not Null |  | Name of cardholder |

1. EmployeeTeams: *Team Names of Contingent Workers*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Employee\_Team\_id | int | Not Null |  | Primary key of Employeeteams table |
|  | Employee\_Team\_name | varchar(20) | Not Null |  | Name of the team |

1. Shippers: *Details of Shippers*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Shipper\_ID | int | Not Null |  | Primary Key of Shippers table |
|  | Shipper\_Name | varchar(20) | Not Null |  | Name of Shipper |

1. BankAccounts: *Bank Account details of Project Managers*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Bank\_id | int | Not Null |  | Primary Key of BankAccounts table |
| Unique | BankAccount\_No | int | Not Null |  | Bank Account Number |
|  | Routing\_No | varchar(10) | Not Null |  | Routing number of the bank |
|  | Accountholder\_Name | varchar(30) | Not Null |  | Bank accountholder’s name |

1. Projects: *Details of ongoing Stakeholder Projects*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Project\_ID | int | Not Null |  | Primary key of Projects table |
|  | Project\_Name | varchar(20) | Not Null |  | Name of the project |
| Foreign Key | Project\_Address\_ID | int | Not Null | Addresses:  Address\_ID | Address of Project: Foreign key to addresses table |

1. Teams: *Details of Stakeholder Teams*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Teams\_ID | int | Not Null |  | Primary key of Teams table |
|  | Team\_Name | varchar(20) | Not Null |  | Name of the team |
| Foreign Key | Project\_ID | int | Not Null | Projects:  Project ID | Assigned Project: Foreign key to Projects table |

1. Stakeholders: *Details of Stakeholders*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Stakeholder\_ID | int | Not Null |  | Primary key of Stakeholders table |
|  | Stakeholder\_Firstname | varchar(20) | Not Null |  | First name of stakeholder |
|  | Stakeholder\_Lastname | varchar(20) | Not Null |  | Last name of stakeholder |
| Unique | Stakeholder\_Email\_ID | varchar(30) | Not Null |  | Email ID of stakeholder |
|  | Stakeholder\_Contact | varchar(15) | Not Null |  | Contact no. of Stakeholder |
|  | Stakeholder\_Title | varchar(30) | Not Null |  | Title od Stakeholder |
| Foreign Key | Stakeholder\_Team\_ID | int | Not Null | Teams:  Team\_Lead\_ID | Assigned team: Foreign key to Teams Table |
| Foreign Key | Stakeholder\_Team\_Lead | int | Not Null | Stakeholders:  Stakeholder\_ID | Team lead id: Foreign key to Stakeholders Table |

1. Employees: *Details of Contingent Workers*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Employee\_ID | int | Not Null |  | Primary Key of Employee table |
|  | Employee\_Title | varchar(30) | Not Null |  | Title of employee |
|  | Employee\_Firstname | varchar(20) | Not Null |  | First name of Employee |
|  | Employee\_Lastname | varchar(20) | Not Null |  | Last name of Employee |
| Unique | Employee\_Email | varchar(30) | Not Null |  | Email ID of Employee |
|  | Employee\_Contact | varchar(15) | Not Null |  | Contact no. Contact |
| Foreign Key | Employee\_Team\_ID | int | Not Null | EmployeeTeams:  Employee\_Team\_ID | Assigned team: Foreign key to Employeeteams table |

1. Orders: *Table containing Orders*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Order\_ID | int | Not Null |  | Primary key of Orders table |
|  | PO\_Number | varchar(5) | Not Null |  | Purchase order number |
|  | Order\_Date | date | Not Null |  | Date of order placed |
| Foreign Key | Stakeholder\_ID | int | Not Null | Stakeholders:  Stakeholder\_ID | ID of Stakeholder who placed the order: Foreign key to Stakeholders table |
| Foreign Key | Employee\_ID | int | Not Null | Employees:  Employee\_ID | ID of Employee who processed the order: Foreign key to Employee table |
| Foreign Key | Shipper | int | Not Null | Shippers:  Shipper\_ID | Assigned Shipper: Foreign key to shipper table |

1. Payments: *Details of Order Payments*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Payment\_ID | int | Not Null |  | Primary key of Payments table |
|  | Payment\_Type | varchar(5) | Not Null |  | Type of Payment: BT or CC |
|  | Payment\_Date | datetime | Not Null |  | Date of Payment |
| Foreign Key | Supplier\_ID | int | Not Null | Suppliers:  Supplier\_ID | Who got paid: Foreign key to Suppliers table |
| Foreign Key | Order\_ID | int | Not Null | Orders:  Order\_ID | Order corresponding to payment: Foreign key to Orders table |

1. Suppliers: *Details of Suppliers*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Supplier\_ID | int | Not Null |  | Primary key of Supplier table |
|  | Supplier\_Name | varchar(20) | Not Null |  | Name of Supplier |
|  | Supplier\_Contact\_FName | varchar(20) | Not Null |  | First Name of contact on supplier side |
|  | Supplier\_Contact\_LName | varchar(30) | Not Null |  | Last Name of contact on supplier side |
|  | Supplier\_Contact\_Email | varchar(30) | Not Null |  | Email ID of contact on supplier side |
|  | Supplier\_Contact | varchar(15) | Not Null |  | Phone of contact on supplier side |
| Foreign Key | Supplier\_Address\_ID | int | Not Null | SupplierAddress:  Supplier\_Address\_ID | Address of Supplier: Foreign key to Supplier Addresses table |

1. SupplierAddress: *Details on Supplier Addresses*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Supplier\_Address\_ID | int | Not Null |  | Primary key of Supplier Addresses table |
|  | Supplier\_Address\_1 | varchar(50) | Not Null |  | Street address of supplier |
|  | Supplier\_Address\_2 | varchar(50) | Not Null |  | Secondary street address of supplier |
|  | Supplier\_City | varchar(20) | Not Null |  | City corresponding to supplier address |
|  | Supplier\_State | varchar(20) | Not Null |  | State corresponding to supplier address |
|  | Supplier\_Zip | int | Not Null |  | Zip corresponding to supplier address |

1. CreditCardPayment: *Associate table about credit card payments*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key, Foreign Key | Payment\_ID | int | Not Null | Payments:  Payment\_ID | Primary key of Payments table |
| Primary Key, Foreign Key | Credit\_Card\_ID | int | Not Null | CreditCards:  Credit\_Card\_ID | Primary key of Credit cards table |

1. BankTransferPayment: *Associate table about Bank transfer payments*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key, Foreign Key | Payment\_ID | int | Not Null | Payments:  Payment\_ID | Primary Key of Payments table |
| Primary Key, Foreign Key | Bank\_ID | int | Not Null | BankAccounts:  Bank\_ID | Primary key of Bank Accounts table |

1. Products: *Product Directory*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key | Product\_ID | int | Not Null |  | Primary key of Products table |
|  | Product\_Name | varchar(50) | Not Null |  | Name of Product |
|  | Product\_Price | int | Not Null |  | Price of Product |
|  | Product\_Reserve | int | Not Null |  | Reserve amount of Product |
|  | Product\_Discount | int | Not Null |  | Discount offered on the product |
|  | Lead\_Time | int | Not Null |  | Predicted delivery time |
| Foreign Key | Supplier\_ID | int | Not Null | Suppliers:  Supplier\_ID | Sold by supplier: Foreign key to Suppliers table |

1. OrderLines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attributes** | **Data Type** | **Nullable** | **Foreign key constraints** | **Description** |
| Primary Key, Foreign Key | Order\_ID | int | Not Null | Orders:  Order\_ID | Primary Key of Orders table |
| Primary Key, Foreign Key | Product\_ID | int | Not Null | Products:  Product\_ID | Primary key of Products table |
|  | Product\_Name | varchar(50) | Not Null |  | Name of Product |
|  | Product\_Price | int | Not Null |  | Price of Product |
|  | Product\_Quantity | int | Not Null |  | Quantity of Product ordered |

**Business Rules:**

* One stakeholder can place one or many orders, but an order can be placed by only one stakeholder
* One Employee can process one or many orders, but an order can be processed by only one Employee
* An order contains one or many products and a product can also be a part of one or many orders
* A supplier manufactures one or many products, but a product can be manufactured by only one supplier
* A supplier can request one or many payments, but one payment is only for one supplier
* One order can be paid via many payments, but one payment is only for one order
* Payments can only be via Bank Transfer or Credit Card
* Employees can only be a part of one of the three teams: Sourcing, Procurement, Contracting
* A team can have many stakeholders and similarly a project can be worked upon by many teams
* There are only four sippers of choice: FedEx, UPS, USPS and DHL

**Database System Infrastructure:**

We used three tools to create our database: MS SQL, Draw.io, PowerApps.

Draw.io:

Draw.io is used to create and share diagrams within a web browser. All entities and relations are clearly shown in the table. Primary keys and foreign keys are clearly written. Data type and is also added when SQL database is constructed.

MS SQL:

MS SQL is the physical database and is used to create queries and maintain the database. We are using MS SQL to use queries for creating tables, inserting data in them and using functions like joins and pivot etc.

PowerApps:

We are using PowerApps to build customized application that will connect our database and the data stored in it and perform various functions to solve business problems.

**Table Creation:**

create DATABASE project

GO

use project

GO

----------------DOWN-------------------

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_OrderLines\_Bank\_ID')

alter table OrderLines drop constraint fk\_OrderLines\_Bank\_ID

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_OrderLines\_Payment\_ID')

alter table OrderLines drop constraint fk\_OrderLines\_Payment\_ID

GO

drop table if exists OrderLines

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_BankTransferPayment\_Bank\_ID')

alter table BankTransferPayment drop constraint fk\_BankTransferPayment\_Bank\_ID

GO

drop table if exists BankTransferPayment

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_BankTransferPayment\_Payment\_ID')

alter table BankTransferPayment drop constraint fk\_BankTransferPayment\_Payment\_ID

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_CreditCardPayment\_Credit\_Card\_ID')

alter table CreditCardPayment drop constraint fk\_CreditCardPayment\_Credit\_Card\_ID

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_CreditCardPayment\_Payment\_ID')

alter table CreditCardPayment drop constraint fk\_CreditCardPayment\_Payment\_ID

GO

drop table if exists CreditCardPayment

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Suppliers\_Supplier\_Address\_ID')

alter table Suppliers drop constraint fk\_Suppliers\_Supplier\_Address\_ID

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Payments\_Supplier\_ID')

alter table Payments drop constraint fk\_Payments\_Supplier\_ID

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Payments\_Order\_ID')

alter table Payments drop constraint fk\_Payments\_Order\_ID

GO

drop table if exists Payments

GO

drop table if exists Products

Go

drop table if exists Suppliers

GO

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Orders\_Shipper')

alter table Orders drop constraint fk\_Orders\_Shipper

GO

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Orders\_Employee\_ID')

alter table Orders drop constraint fk\_Orders\_Employee\_ID

GO

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Orders\_Stakeholder\_ID')

alter table Orders drop constraint fk\_Orders\_Stakeholder\_ID

GO

drop table if exists Orders

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Employees\_Employee\_Team\_ID')

alter table Employees drop constraint fk\_Employees\_Employee\_Team\_ID

GO

drop table if exists Employees

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Stakeholders\_Stakeholder\_Team\_ID')

alter table Stakeholders drop constraint fk\_Stakeholders\_Stakeholder\_Team\_ID

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Stakeholder\_Team\_Lead\_ID')

alter table Stakeholders drop constraint fk\_Stakeholder\_Team\_Lead\_ID

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Teams\_Project\_Team\_Lead\_ID')

alter table Teams drop constraint fk\_Teams\_Project\_Team\_Lead\_ID

GO

drop table if exists Stakeholders

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Teams\_Project\_Project\_ID')

alter table Teams drop constraint fk\_Teams\_Project\_Project\_ID

GO

drop table if exists Teams

GO

if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

where CONSTRAINT\_NAME='fk\_Projects\_Project\_Address\_ID')

alter table Projects drop constraint fk\_Projects\_Project\_Address\_ID

GO

drop table if exists Projects

GO

drop table if exists BankAccounts

Go

drop table if exists SupplierAddresses

GO

drop table if exists Shippers

GO

drop table if exists EmployeeTeams

GO

drop table if exists CreditCards

GO

drop table if exists Addresses

GO

------------------UP-------------------

------create table Addresses-----------

--done

create table Addresses(

Address\_id int IDENTITY not null,

Street\_Address\_1 VARCHAR(50) not NULL,

Street\_Address\_2 VARCHAR(50) not NULL,

City VARCHAR(20) not NULL,

States VARCHAR(20) not NULL,

Zip int not NULL

constraint pk\_Addresses\_Address\_id PRIMARY key (Address\_id)

)

GO

------create table CreditCards-----------

create table CreditCards(

Credit\_card\_id int IDENTITY not null,

Credit\_card\_No int not NULL,

Expiration\_Month int not NULL,

Expiration\_Year int not NULL,

Cardholder\_Name VARCHAR(30) not NULL

constraint pk\_CreditCards\_Credit\_card\_id PRIMARY key (Credit\_card\_id),

constraint u\_CreditCards\_Credit\_card\_No UNIQUE(Credit\_card\_No)

)

GO

------create table EmployeeTeams-----------

--done

create table EmployeeTeams(

Employee\_Team\_id int IDENTITY not null,

Employee\_Team\_name VARCHAR(20) not NULL

constraint pk\_EmployeeTeams\_Employee\_Team\_id PRIMARY key (Employee\_Team\_id)

)

GO

------create table Shippers-----------

--done

create table Shippers(

Shipper\_ID int IDENTITY not null,

Shipper\_Name VARCHAR(20) not NULL

constraint pk\_Shippers\_Shipper\_id PRIMARY key (Shipper\_id)

)

GO

------create table BankAccounts-----------

--done

create table BankAccounts(

Bank\_id int IDENTITY not null,

BankAccount\_No int not NULL,

Routing\_No varchar(10) not NULL,

Accountholder\_Name VARCHAR(30) not NULL

constraint pk\_BankAccounts\_Bank\_id PRIMARY key (Bank\_id),

constraint u\_BankAccounts\_BankAccount\_No UNIQUE(BankAccount\_No)

)

GO

------create table Projects-----------

--done

create table Projects(

Project\_ID int IDENTITY not null,

Project\_Name VARCHAR(20) not NULL,

Project\_Address\_ID int not null

constraint pk\_Projects\_Project\_id PRIMARY key (Project\_id)

)

GO

alter table Projects

add constraint fk\_Projects\_Project\_Address\_ID foreign key(Project\_Address\_ID)

REFERENCES Addresses(Address\_id)

GO

------create table Teams-----------

create table Teams(

Team\_ID int IDENTITY not null,

Team\_Name VARCHAR(20) not NULL,

Project\_ID int not null,

constraint pk\_Teams\_Team\_ID PRIMARY key (Team\_ID)

)

GO

alter table Teams

add constraint fk\_Teams\_Project\_Project\_ID foreign key(Project\_ID)

REFERENCES Projects(Project\_ID)

GO

------create table Stakeholders-----------

create table Stakeholders(

Stakeholder\_ID int IDENTITY not null,

Stakeholder\_Firstname VARCHAR(20) not NULL,

Stakeholder\_Lastname VARCHAR(20) not NULL,

Stakeholder\_Team\_ID int not null,

Stakeholder\_Email\_ID VARCHAR(30) not NULL,

Stakeholder\_Contact VARCHAR(15) not NULL,

Stakeholder\_Title VARCHAR(30) not NULL,

Stakeholder\_Team\_Lead int not null

constraint pk\_Stakeholders\_Stakeholder\_ID PRIMARY key (Stakeholder\_ID),

constraint u\_Stakeholders\_Stakeholder\_Email\_ID UNIQUE(Stakeholder\_Email\_ID)

)

GO

alter table Stakeholders

add constraint fk\_Stakeholders\_Stakeholder\_Team\_ID foreign key(Stakeholder\_Team\_ID)

REFERENCES Teams(Team\_ID)

GO

alter table Stakeholders

add constraint fk\_Stakeholder\_Team\_Lead\_ID foreign key(Stakeholder\_Team\_Lead)

REFERENCES Stakeholders(Stakeholder\_ID)

GO

------create table Employees-----------

--done

create table Employees(

Employee\_ID int IDENTITY not null,

Employee\_Title VARCHAR(30) not null,

Employee\_Firstname VARCHAR(20) not NULL,

Employee\_Lastname VARCHAR(20) not NULL,

Employee\_Email VARCHAR(30) not NULL,

Employee\_Contact VARCHAR(15) not NULL,

Employee\_Team\_ID int not null

constraint pk\_Employees\_Employee\_ID PRIMARY key (Employee\_ID),

constraint u\_Employees\_Employee\_Email UNIQUE(Employee\_Email)

)

GO

alter table Employees

add constraint fk\_Employees\_Employee\_Team\_ID foreign key(Employee\_Team\_ID)

REFERENCES EmployeeTeams(Employee\_Team\_ID)

GO

------create table Orders-----------

create table Orders(

Order\_ID int IDENTITY not null,

PO\_Nmber VARCHAR(5) not null,

Order\_Date DATE ,

Stakeholder\_ID int not null,

Employee\_ID int not null,

Shipper int not null,

constraint pk\_Orders\_Order\_ID PRIMARY key (Order\_ID)

)

GO

alter table Orders

add constraint fk\_Orders\_Stakeholder\_ID foreign key(Stakeholder\_ID)

REFERENCES Stakeholders(Stakeholder\_ID)

GO

alter table Orders

add constraint fk\_Orders\_Employee\_ID foreign key(Employee\_ID)

REFERENCES Employees(Employee\_ID)

GO

alter table Orders

add constraint fk\_Orders\_Shipper foreign key(Shipper)

REFERENCES Shippers(Shipper\_ID)

GO

------create table Payments-----------

--done

create table Payments(

Payment\_ID int IDENTITY not null,

Payment\_Type VARCHAR(5) not NULL,

Payment\_Date date not NULL,

Supplier\_ID int not null,

Order\_ID int not null

constraint pk\_Payments\_Payment\_ID PRIMARY key (Payment\_ID)

)

GO

alter table Payments

add constraint fk\_Payments\_Order\_ID foreign key(Order\_ID)

REFERENCES Orders(Order\_ID)

GO

------create table Suppliers-----------

--done

create table Suppliers(

Supplier\_ID int IDENTITY not null,

Supplier\_Name VARCHAR(20) not NULL,

Supplier\_Contact\_FName VARCHAR(20) not NULL,

Supplier\_Contact\_LName VARCHAR(30) not NULL,

Supplier\_Contact\_Email VARCHAR(30) not NULL,

Supplier\_Contact VARCHAR(15) not NULL,

Supplier\_Address\_ID int not null

constraint pk\_Suppliers\_Supplier\_ID PRIMARY key (Supplier\_ID)

)

GO

alter table Payments

add constraint fk\_Payments\_Supplier\_ID foreign key(Supplier\_ID)

REFERENCES Suppliers(Supplier\_ID)

GO

------create table SupplierAddresses-----------

--done

create table SupplierAddresses(

Supplier\_Address\_ID int IDENTITY not null,

Supplier\_Address\_1 VARCHAR(50) not NULL,

Supplier\_Address\_2 VARCHAR(50) not NULL,

Supplier\_City VARCHAR(20) not NULL,

Supplier\_State VARCHAR(20) not NULL,

Supplier\_Zip int not NULL

constraint pk\_SupplierAddresses\_Address\_id PRIMARY key (Supplier\_Address\_ID)

)

GO

alter table Suppliers

add constraint fk\_Suppliers\_Supplier\_Address\_ID foreign key(Supplier\_Address\_ID)

REFERENCES SupplierAddresses(Supplier\_Address\_ID)

GO

------create table CreditCardPayment-----------

create table CreditCardPayment(

Payment\_ID int not null,

Credit\_Card\_ID int not null,

constraint pk\_CreditCardPayment\_Payment\_ID\_Credit\_Card\_ID PRIMARY key (Payment\_ID,Credit\_Card\_ID)

)

GO

alter table CreditCardPayment

add constraint fk\_CreditCardPayment\_Payment\_ID foreign key(Payment\_ID)

REFERENCES Payments(Payment\_ID)

GO

alter table CreditCardPayment

add constraint fk\_CreditCardPayment\_Credit\_Card\_ID foreign key(Credit\_Card\_ID)

REFERENCES CreditCards(Credit\_Card\_ID)

GO

------create table BankTransferPayment-----------

create table BankTransferPayment(

Payment\_ID int not null,

Bank\_ID int not null,

constraint pk\_BankTransferPayment\_Payment\_ID\_Bank\_ID PRIMARY key (Payment\_ID,Bank\_ID)

)

GO

alter table BankTransferPayment

add constraint fk\_BankTransferPayment\_Payment\_ID foreign key(Payment\_ID)

REFERENCES Payments(Payment\_ID)

GO

alter table BankTransferPayment

add constraint fk\_BankTransferPayment\_Bank\_ID foreign key(Bank\_ID)

REFERENCES BankAccounts(Bank\_ID)

GO

------create table Products-----------

--done

create table Products(

Product\_ID int IDENTITY not null,

Product\_Name VARCHAR(50) not NULL,

Product\_Price int not NULL,

Product\_Reserve int not NULL,

Product\_Discount int not NULL,

Lead\_Time int not NULL,

Supplier\_ID int not null

constraint pk\_Products\_Product\_ID PRIMARY key (Product\_ID)

)

GO

alter table Products

add constraint fk\_Products\_Supplier\_Supplier\_ID foreign key(Supplier\_ID)

REFERENCES Suppliers(Supplier\_ID)

GO

------create table OrderLines-----------

create table OrderLines(

Order\_ID int not null,

Product\_ID int not null,

Product\_Name VARCHAR(50) not NULL,

Product\_Price int not NULL,

Product\_Quantity int not NULL

constraint pk\_OrderLines\_Payment\_ID\_Bank\_ID PRIMARY key (Order\_ID,Product\_ID)

)

GO

alter table OrderLines

add constraint fk\_OrderLines\_Payment\_ID foreign key(Order\_ID)

REFERENCES Orders(Order\_ID)

GO

alter table OrderLines

add constraint fk\_OrderLines\_Bank\_ID foreign key(Product\_ID)

REFERENCES Products(Product\_ID)

GO

**Insertion of sample data:**

----------------INSERTS-------------------

------insert into table Addresses-----------

insert into Addresses

(Street\_Address\_1,Street\_Address\_2,City,States,Zip)

VALUES

('436 Columbus Avenue','Next to Chase','Syracuse','NY','13210'),

('567 Akerman Avenue','Next to BOA','NYC','NY','13250'),

('343 Lancaster Avenue','Next to Fedex','Olympia','NJ','13211'),

('757 Maryland Avenue','Next to UPS','Syracuse','NY','13210'),

('222 Hawthorne Avenue','Next to Walmart','Syracuse','NY','13210'),

('888 Richard Avenue','Next to Pricerite','Boston','MA','14333'),

('222 Livingston Avenue','Next to Hendricks','Syracuse','NY','13210'),

('441 Kensington Avenue','Next to Alto Cinco','Los Angeles','CA','15321'),

('909 Ostrom Avenue','Next to SU','Syracuse','NY','13210'),

('245 Bell Avenue','Next to Chase','Los Angeles','CA','15321')

GO

------insert into table CreditCards-----------

insert into CreditCards

(Credit\_card\_No,Expiration\_Month,Expiration\_Year,Cardholder\_Name)

VALUES

('2444222','01','2023','Mikhail Pinto'),

('2446422','08','2027','Pranali Shenvi'),

('2444202','12','2025','Ruchak Vira')

GO

------insert into table EmployeeTeams-----------

insert into EmployeeTeams

(Employee\_Team\_name)

VALUES

('Billing'),

('HR'),

('Finance'),

('IT'),

('Support'),

('Enrollment'),

('Cloud'),

('Tech'),

('Analytics'),

('Presentations')

GO

GO

------insert into table Shippers-----------

insert into Shippers

(Shipper\_Name)

VALUES

('UPS'),

('Fedex'),

('USPS'),

('DHL')

GO

------insert into table BankAccounts-----------

insert into BankAccounts

(BankAccount\_No,Routing\_No,Accountholder\_Name)

VALUES

('3456322','CS0927','Mikhail Pinto'),

('6677442','RO8329','Pranali Shenvi'),

('3424563','LD0393','Ruchak Vira')

GO

------insert into table Projects-----------

insert into Projects

(Project\_Name,Project\_Address\_ID)

VALUES

('BCBS','9'),

('Verizon','3'),

('Blake','8'),

('T-Mobile','1'),

('Walgreens','2'),

('SU','5'),

('Orange','7'),

('ZS','6'),

('GM','4'),

('Seagate','10')

GO

------insert into table SupplierAddresses-----------

insert into SupplierAddresses

(Supplier\_Address\_1,Supplier\_Address\_2,Supplier\_City,Supplier\_State,Supplier\_Zip)

VALUES

('466 Columbus Avenue','Next to Chase','Syracuse','NY','13210'),

('568 Akerman Avenue','Next to Chase','New York','NY','13250'),

('343 Lancaster Avenue','Next to Fedex','Olympia','NJ','13210'),

('754 Maryland Avenue','Next to Fedex','Syracuse','NY','13210'),

('221 Hawthorne Avenue','Next to Walmart','Syracuse','NY','13210')

GO

------insert into table Employees-----------

insert into Employees

(Employee\_Firstname,Employee\_Lastname,Employee\_Email,Employee\_Contact,Employee\_Team\_ID, Employee\_Title)

VALUES

('Rachel','Greene','raGreene@gmail.com','31599206309','8','Sourcing Manager'),

('Jung','Ho','Jnh@gmail.com','31599006309','2','Sourcing Associate'),

('Pranav','Sheth','psheth189@gmail.com','31599208809','4', 'Procurement Associate'),

('Ashish','Waghmare','awagh1@gmail.com','31109206309','7','Procurement Manager'),

('Ankita','Agarwal','aagrawal@gmail.com','31599204409','1', 'Contracts Associate'),

('Harsh','Athavale','haathawale@gmail.com','31599206209','5', 'Contracts Manager'),

('Naruto','Uzumaki','hiddenleaf\_123@gmail.com','31339206309','10', 'Procurement Associate'),

('Sasuke','Uchiha','rogue\_ninja@gmail.com','31599208809','9','Contracts Associate'),

('Eren','Yaeger','titan12@gmail.com','31508206309','6', 'Sourcing Associate'),

('Misa','Amane','kira2@gmail.com','31592306309','3', 'Group Manager')

GO

------insert into table Suppliers-----------

insert into Suppliers

(Supplier\_Name,Supplier\_Contact\_FName,Supplier\_Contact\_LName,Supplier\_Contact\_Email,

Supplier\_Contact,Supplier\_Address\_ID)

VALUES

('Ballistic Machines','Bill','Gray','Billie@gmail.com','3109928933','2'),

('Houser Electronics','John','Greene','Teled@gmail.com','3109921133','3'),

('3D Microtec','Agatha ','Christie','ASI\_emp@gmail.com','3119928933','5'),

('ProServ Servers','Nick','Fury','US\_Demp@gmail.com','3109922233','4'),

('Teledyne Electronics','Peter','Dinklage','WpetEmp@gmail.com','3009928933','1')

GO

------insert into table Teams-----------

insert into Teams

(Team\_Name,Project\_ID)

VALUES

('Seagate Deployment','10'),

('Verizon Execution','2'),

('GM Research','9')

GO

------insert into table Stakeholders-----------

insert into Stakeholders

(Stakeholder\_Firstname,Stakeholder\_Lastname,Stakeholder\_Team\_ID,Stakeholder\_Email\_ID,Stakeholder\_Contact,

Stakeholder\_Title, Stakeholder\_Team\_Lead)

VALUES

('Pranali','Shenvi','1','psehnvi@gmail.com','3159803790', 'Project Manager','1'),

('Mikhail','Pinto','2','mike112@gmail.com','3159803791', 'Project Manager','2'),

('Ruchak','Vira','3','rnvira47@gmail.com','3159803792', 'Project Manager','3'),

('Neel','Samant','1','neelsa123@gmail.com','3159803793', 'Research Scientist','1'),

('Pranjali','Shenvi','1','prahsen@gmail.com','3159803794', 'Software Engineer','1'),

('Usha','Sen','1','usen@gmail.com','3159803795', 'Hardware Engineer','1'),

('Manik','Tiles','2','mati@gmail.com','3159803796', 'Hardware Engineer','2'),

('Rupa','Saarees','2','rusa@gmail.com','3159803797', 'Software Engineer','2'),

('Kachrina','Kemph','2','kake@gmail.com','3159803798', 'Research Scientist','2'),

('Dola','Tum','3','dolun@gmail.com','3159803799', 'Software Engineer','3'),

('Nurinder','Moody','3','nurinchamp@gmail.com','3159803700', 'Research Scientist','3'),

('Pappu','Gandhi','3','pape@gmail.com','3159803701', 'Hardware Engineer','3'),

('Saif Ali','Ramprasad','1','nawab@gmail.com','3159803702', 'Data Analyst','1'),

('Raja','Hindustani','2','emirkhun@gmail.com','3159803703', 'Data Analyst','2'),

('Mader','Codh','3','hogatu@gmail.com','3159803704', 'Data Analyst','3')

------insert into table Orders-----------

insert into Orders

(PO\_Nmber,Order\_Date,Stakeholder\_ID,Employee\_ID,Shipper)

VALUES

('367','2018-12-31','4','6','2'),

('639','2019-10-31','5','10','4'),

('892','2018-07-10','9','4','3'),

('938','2019-06-08','13','9','1'),

('739','2019-11-30','15','2','1'),

('999','2018-06-23','14','1','2'),

('157','2017-05-31','7','7','3'),

('493','2018-10-12','8','5','3'),

('902','2018-08-19','10','8','2'),

('678','2018-05-30','6','3','1')

GO

------insert into table Payments-----------

insert into Payments

(Payment\_Type,Payment\_Date,Supplier\_ID, Order\_ID)

VALUES

('CC','2021-11-30','1','1'),

('BT','2020-05-06','1','2'),

('CC','2020-12-10','2','3'),

('CC','2020-03-02','2','4'),

('CC','2020-12-31','2','5'),

('BT','2021-11-12','3','6'),

('BT','2020-12-31','3','7'),

('CC','2020-04-05','4','8'),

('BT','2020-06-07','4','9')

GO

------insert into table Products-----------

insert into Products

(Product\_Name,Product\_Price,Product\_Reserve,Product\_Discount,Lead\_Time,Supplier\_ID)

VALUES

('Servers','100000','50','5','14','1'),

('Probes','200','1000','40','4','1'),

('Tablithe Waifer','4500','500','20','5','2'),

('3-DM Pro Neurometer','26000','35','10','5','2'),

('Oscilloscope','53000','120','15','14','3'),

('3-D Printers','21000','190','5','8','3'),

('Inferometer','32000','100','8','9','3'),

('Plasma Cutter','17000','400','0','7','4'),

('Pro-Gen Graphite Cable','400','5600','50','15','4'),

('Ultrasound Machine','67000','70','25','20','5')

GO

------insert into table BankTransferPayment-----------

insert into BankTransferPayment

(Payment\_ID,Bank\_ID)

VALUES

('7','2'),

('2','1'),

('6','2'),

('9','3')

GO

------insert into table CreditCardPayment-----------

insert into CreditCardPayment

(Payment\_ID,Credit\_Card\_ID)

VALUES

('1','1'),

('8','2'),

('3','2'),

('5','3'),

('4','1')

GO

------insert into table OrderLines-----------

insert into OrderLines

(Order\_ID,Product\_ID,Product\_Name,Product\_Price,Product\_Quantity)

VALUES

('1','1','Servers','100000','3'),

('2','1','Servers','100000','8'),

('1','2','Probes','200','6'),

('2','2','Probes','200','3'),

('3','3','Tablithe Wafer','4500','4'),

('4','3','Tablithe Wafer','4500','2'),

('4','4','3-DM Pro Neurometer','26000','7'),

('5','4','3-DM Pro Neurometer','26000','9'),

('6','5','Oscilloscope','53000','8'),

('6','6','3-D Printers','21000','6'),

('7','6','3-D Printers','21000','10'),

('6','7','Inferometer','32000','6'),

('8','8','Plasma Cutter','17000','8'),

('8','9','Pro-gen Graphite cable','400','5'),

('9','9','Pro-gen Graphite cable','400','3'),

('10','10','Ultrasound Machine','67000','4')

GO

**Answering business questions:**

--1 We would like a list of product names with product id and type along with a count of product quantities for each shipper names. There should be a column for each of the Shippers.

This is helpful for the stakeholder to know how many orders are shipped by which Shipper. This will help the stakeholders understand which shippers are reliable and determine which shippers can be used for mass deliveries and which shippers should be used for small order deliveries.

First, we use joins to combine the required tables: Orders and Shippers. We then use pivot to arrange the product quantities according to the shipper using count of Product\_Quantity. Because of this, finally we get the product\_IDs and the Product Names with respect to each shipper.

with pivot\_source3 as (

    selectol.Product\_ID,ol.Product\_Name,s.Shipper\_Name,ol.Product\_Quantity

    from OrderLines ol

    join Orders o on ol.Order\_ID=o.Order\_ID

    join Shippers s on o.Shipper=s.Shipper\_ID

)

select \* from pivot\_source3 pivot(

    count(Product\_Quantity) for Shipper\_Name IN("Fedex","DHL","UPS","USPS")

)as pivot\_table3

Table

Description automatically generated

--2 What is the spend of each Stakeholder team?

This can be considered as the most important business question. Stakeholder is the biggest part of the procurement as it procures different products. It is important to track the spend and determine how the budget can be used. This can be used to know how much a team on a project is spending to get it done.

First, we are using joins to connect the tables: Orders, OrderLines and Products. We are using the partition function to partition over Team\_Name and taking the sum of the product of product prices and the product quantities.

We are also creating a View for this query.

In the result, we are getting the total spend for each Team\_Name.

CREATE VIEW Spend\_By\_StakeholderTeam

AS

Select distinct t.Team\_Name, sum((ol.Product\_Price\*ol.Product\_Quantity)) OVER (PARTITION byt.Team\_Name) as TotalSpend

from Teams t JOIN Stakeholders s

on t.Team\_ID=s.Stakeholder\_Team\_ID

JOIN Orders o on s.Stakeholder\_ID = o.Stakeholder\_ID

JOIN OrderLines ol on o.Order\_ID=ol.Order\_ID

JOIN Products p on ol.Product\_ID=p.Product\_ID

Table

Description automatically generated

--3 Which orders have not been paid for?

This is also one of the most important business questions and can be taken as the credit worthiness of an organization. No supplier wants to sell their products for free. And this query helps us to determine how many and which orders have not been paid for. With the help of this approach the stakeholder can successfully settle all dues and can be a part of a healthy procurement system. We are using joins to connect tables: Stakeholders, Orders, Teams and Payments. We are using a where clause to fetch the data where Order\_ID is NULL. This will give us the output for those orders which have not been paid.

Select o.Order\_ID, s.Stakeholder\_Firstname + ' ' +s.Stakeholder\_Lastname as StakeholderName,

s1.Stakeholder\_Firstname + ' ' + s1.Stakeholder\_Lastname as TeamLead, t.Team\_Name, o.Order\_Date

from Teams t JOIN Stakeholders s

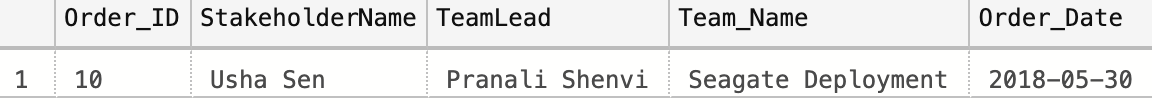
on t.Team\_ID = s.Stakeholder\_Team\_ID

join Stakeholders s1 on s1.Stakeholder\_ID = s.Stakeholder\_Team\_Lead

join Orders o on s.Stakeholder\_ID = o.Stakeholder\_ID

left join Payments p on o.Order\_ID = p.Order\_ID

where p.Order\_ID is NULL



--4 How many orders correspond to each zip code?

This is more of an internal optimization business question, one of the most important features of procurement management is internal inventory management. With the help of this approach the stakeholder organization can determine exactly which projects or warehouses require more inventory products. Thus, it can help in an efficient control of the inventory and decrease lead times. Here we are using joins to connect the tables: Teams, Stakeholders, Orders and Addresses. We are using pivot to fetch data from Zip and plot it in a tabular form in such a way that we can see the number of orders in each zip code.

with pivot\_source AS

(

    Select Zip

    from Addresses a join Projects p

    on a.Address\_id=p.Project\_Address\_ID

    join Teams t on p.Project\_ID=t.Project\_ID

    JOIN Stakeholders s on t.Team\_ID=s.Stakeholder\_Team\_ID

    JOIN Orders o on s.Stakeholder\_ID=o.Stakeholder\_ID

)

SELECT \* from pivot\_source PIVOT (

    COUNT(zip) for Zip in ("15321","13210","13211")

) as pivot\_table

Table

Description automatically generated

--5 How many suppliers correspond to each zip code?

This can be another solution to the supply chain problem. Stakeholders need the products to get delivered as soon as possible but due to some unforeseen reasons or natural issues, it is not always the case. Thus, it is important to determine where exactly the suppliers are delivering the products from. This could help in determining which supplier to choose if there is an urgent requirement of product. We are using joins to combine the tables: SupplierAddresses and Suppliers. Then we are using the pivot function to fetch the data in every zip code and plot it in a tabular form in such a way that each zip code corresponds to the number of suppliers.

with pivot\_source5 AS

(

    Select s.Supplier\_ID, s.Supplier\_Name, sa.Supplier\_Zip

    from SupplierAddresses sa join Suppliers s

    on sa.Supplier\_Address\_ID=s.Supplier\_Address\_ID

)

SELECT \* from pivot\_source5 PIVOT (

    COUNT(Supplier\_Zip) for Supplier\_Zip in ("13210","13211","13250")

) as pivot\_table5

Table

Description automatically generated

--6 Create list of product names, price and payment method?

Sometimes, in order to analyze the demand, it is important to determine what type of payment method is used on an order. A credit card payment means the order needs to be urgently delivered while a bank transfer payment might mean the product is expensive and can take up to longer lead times, here, we can filter out the products using payment methods. Here, we are using joins to connect the tables Products, Suppliers and Payments. We are using the pivot function to fetch the data according to the Product price in such a way that the product name is displayed with the payment method as bank Transfer or Credit Card.

with pivot\_source1 AS

(

    Select p.Product\_Name, p.Product\_Price ,pa.Payment\_Type

    from Products p join Suppliers s

    on p.Supplier\_ID = s.Supplier\_ID

    join Payments pa on s.Supplier\_ID=pa.Supplier\_ID

)

SELECT \* from pivot\_source1 PIVOT (

    sum(Product\_Price) for Payment\_Type in ("BT","CC")

) as pivot\_table1

**Table

Description automatically generated**

**Interface Implementation:**

**Diagram

Description automatically generated**

The screen above shows the design of our home screen. There are 4 clickable buttons: ‘I am a Stakeholder’, ‘I am an Employee’, ‘I am a Supplier’ and ‘Payment Portal’. Clicking these buttons take you to the respective screens depending upon the user.

**Diagram, engineering drawing

Description automatically generated**

The above screen shows the interface of the Stakeholder. This page pops up when you click on the ‘I am a stakeholder’ button on the Homepage. This page has 2 clickable buttons: ‘Stakeholder directory’ and ‘Place an Order’.

**A picture containing table

Description automatically generated**

This page pops up when you click on ‘Stakeholder directory’ on the Stakeholder page. This page gives you the option to go through the details of each stakeholder including their role and email ID. This list is searchable and sortable.

**Graphical user interface

Description automatically generated with medium confidence**

This page pops up when you click on ‘Place an order’ button on the Stakeholder screen. This page is an insert form which lets the Stakeholder place an order based on the requirements of the team. This form lets the stakeholder choose the employee who is going to process the order and the shipper who is going to be assigned. Once, an order is placed, it will pop up on the Orders screen.

**Diagram, engineering drawing

Description automatically generated**

The above screen shows the interface of the Employee. This page pops up when you click on the ‘I am an Employee’ button on the Homepage. This page has 2 clickable buttons: ‘Employee directory’ and ‘Orders Assigned’.

**Table

Description automatically generated with low confidence**

This page pops up when you click on ‘Employee directory’ on the Employee page. This page gives you the option to go through the details of each Employee including their role and email ID. This list is searchable and sortable.

**Table

Description automatically generated**

This page pops up when you click on ‘Orders Assigned’ on the Employee page. This page gives you the option to go through the details of each Order including the date when the order was placed and Purchase Order Number. This list is searchable and sortable.

**Graphical user interface, application

Description automatically generated**

This page pops up when you click on any order on the ‘Orders Assigned’ page. This page gives you the option to go through the details of each Order including the date when the order was placed and Purchase Order Number, Shipper ID, Stakeholder ID. This is a viewable form. There is also an option to process the order.

**Text, table

Description automatically generated with medium confidence**

This page pops up when you click on ‘I am a Supplier’ on the home screen. This page gives you the option to go through the details of each Supplier including the name of the contact on the supplier side and email ID. This list is searchable and sortable.

**Table

Description automatically generated**

This page pops up when you click on ‘Payment Portal’ on the Home screen. This page gives you the option to go through the details of each Order including the date when the order was placed and Purchase Order Number. This list is searchable and sortable.

**Graphical user interface, application

Description automatically generated**

This page pops up when you click on any order on the ‘Orders pending payment’ page. This page gives you the option to go through the details of each Order including the date when the order was placed and Purchase Order Number, Shipper ID, Stakeholder ID. This is a viewable form. There is an option to Pay for the order by card or by Bank transfer.

**Graphical user interface

Description automatically generated with medium confidence**

This page pops up when you click on ‘Pay by Card’ on the ‘Orders pending payment’ page. This page contains a dropdown list of the names of 3 project managers who are responsible for the payment of an order. Furthermore, there is a text box to input the card number along with expiration month and year of the card. This page is a form that creates a new row in the payments table.

**A picture containing graphical user interface

Description automatically generated**

This page pops up when you click on ‘Pay by Bank Account’ on the ‘Orders pending payment’ page. This page contains a Text box to enter one of the names of 3 project managers who are responsible for the payment of an order. Furthermore, there is a text box to input the Bank Account Number along with Routing Number. This page is a form that creates a new row in the payments table.

**Graphical user interface, application

Description automatically generated**

This page pops up when a payment is done either by Card or Bank Transfer.

Link to the video demonstration of the SMARTPROC App using PowerApps:

<https://video.syr.edu/media/t/1_rbp57zjs>

**Logbook:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Tasks** | **Implemented By:** |
| 1. | Proposing the initial project concept: Coming up with the topic for the project after using the websites for reference and brainstorming | Pranali |
| 2. | Confirming the final project topic: Revising the current project topic by adding a few entities and other constraints | Pranali, Priyal, Srushti |
| 3. | Designing the Conceptual Diagram | Srushti |
| 4. | Designing the Logical Diagram | Pranali, Priyal |
| 5. | Creating the database and the tables | Priyal |
| 6. | Inserting sample data values in the tables | Priyal, Srushti |
| 7. | Answering business questions: Coming up with the possible ideas of business questions and implementing them using pivots, joins etc. | Pranali, Priyal |
| 8. | PowerApps: Initial making of the basic screens with buttons | Srushti |
| 9. | PowerApps: Adding the extra functions and connections in the screens and finalizing the PowerApps | Srushti, Pranali |
| 10. | Project Report: Initial building of the report template, adding the code and the screenshots of the PowerApps UI | Srushti |
| 11. | Project Report: Writing about the project details like the background and the information about the business questions | Pranali |
| 12. | Project Report: Completing the Project Report for the remaining pieces | Priyal |
| 13. | PowerPoint Presentation | Pranali, Priyal, Srushti |