Procurement Management System

Database Specification:

Purpose, Business Problems Addressed, Business Rules, Design decisions

Ankur Chauhan, Aparna Shelar, Roshni Dhami, Rahul Bhiwande, Sudipta Saha, Urvi Aryamane

Database Purpose:

The purpose of Procurement Management database is to maintain the data used to support the procurement process of an organization by establishing and maintain strategic thinking in all procurement efforts with high scalability. This will result in responsible direction of organization's resources, streamlining of department cost utilization by engaging the lucrative suppliers and effective budget allocation thereby promoting greater spend efficiency while reducing the costs and analysing the quality of goods and services. The system will be used by the administration staff only and will not have redundant data.

Business Problems Addressed:

- Preventing unauthorized purchase in the organization as the order tickets needs to be approved by the manager.
- Analysing the expense of each department and creating a budget accordingly.
- Helping management create strategy to control spending, for department going over the allocated budget.
- Choosing efficient supplier by selecting product based on rating/available discounts
- Storing purchase order and invoicing details for easy auditing.
- Preventing wasteful purchasing as admin will check the inventory before processing an order.
- Well-organized procurement process is ensured by using catalogue to maintain details of each product for effective management of the products provided by the supplier.

Business Rules:

- The General staff has 0 or more tickets
- Tickets generated by the general staff will follow approval work flow
- The Manager manages at least 1 general staff
- One manager is associated with only one dept
- A manager can have 0 or more requisitions
- One admin manages all the requisitions
- Admin will check the inventory, approve the requisition and then place the order
- One product can have multiple suppliers
- One supplier can have multiple products
- One category has multiple products
- One dept can have multiple contracts with the supplier based on the categories
- One order has one invoice

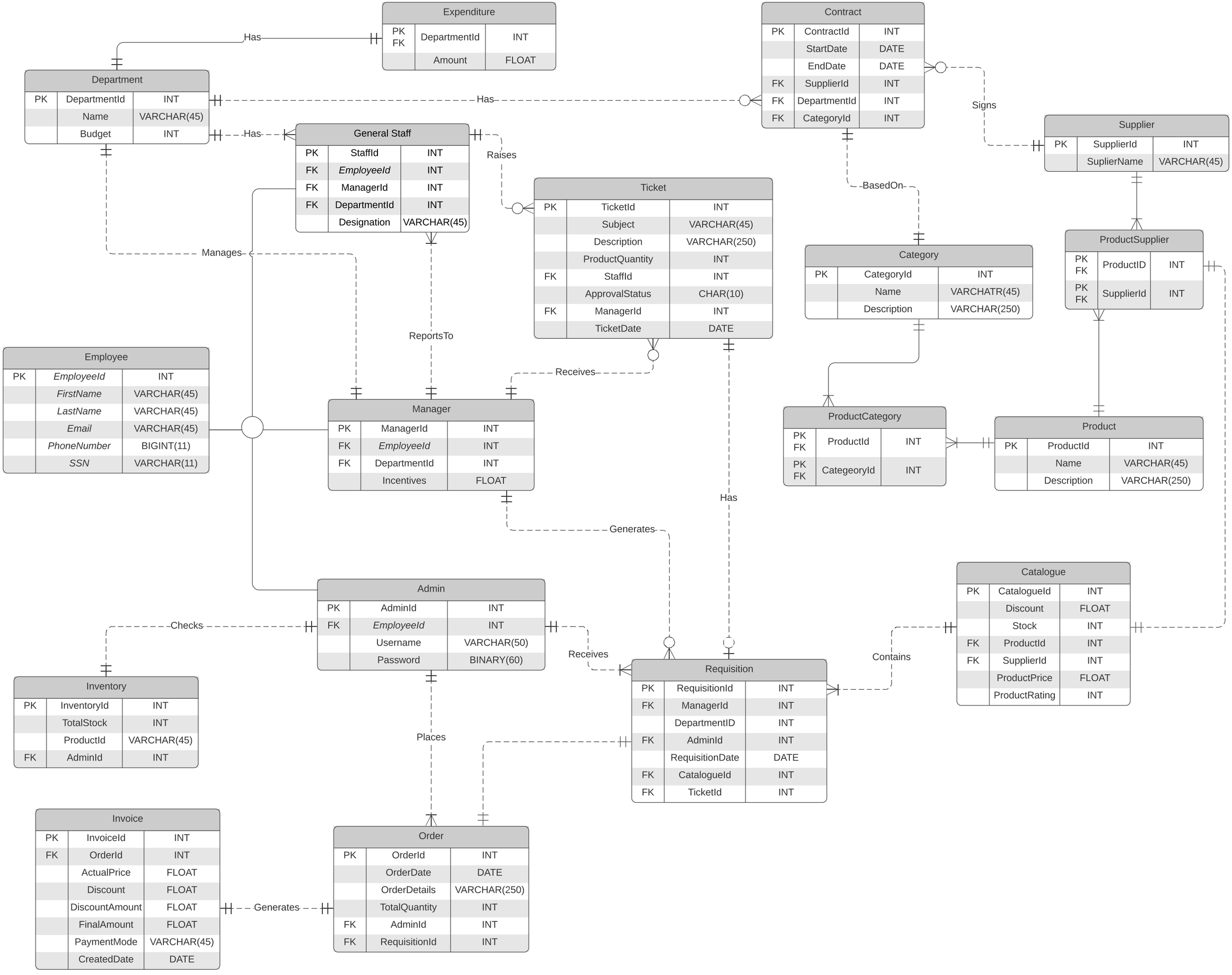
- One product can have multiple catalogues
- One ticket will have only one product
- One employee will have only one contact number

Design Decision:

Entity Name	Why the entity is included	How entities are related to other entities
Employee	This is one of the most important entity in Procurement Management Database System. This entity will contain information about all the employees of organisation irrespective of their team, role etc. Employee table will store personal information of each employee such as their Name, Department, phone number, EmailID etc.	Employee entity is a superclass of the subclass entities admin, manager and general staff. Employee Entity's primary key EmployeeID relates it to Ticket Entity so that ticket raised by each employee in organisation can be tracked.
Staff	Staff Entity plays an important role in the overall procurement process as they generate a ticket, which is then passed onto the manager for further approval. Staff entity acts as an initial generator of procurement request.	As one of the subclass entities of this database, staff entity is mainly associated with ticket, department, and manager entity. Staff entity contains keys such as ManagerID so as to indicate who manages the particular employee. It also contains DepartmentID to indicate the staff's department. The most important key which this entity contains is the TicketID displaying which staff has generated the ticket.
Manager	Manager entity will hold information about Manager and the department he works in. Manager need to approve the ticket to process it further. Approval is required in order to satisfy the business rule which states the procurement request needs to go through approval workflow	Manager entity is subclass of employee entity and is associated with Department entity via primary key DepartmentID, which indicate the Manager's department. Manager is related to ticket entity and general staff entity as manager will receive the tickets, generated by employees he is managing. Manager entity is also connected with Requisition entity as manger has the authority to convert a ticket into requisition.
Admin	Admin entity contains admin information who is responsible to checks the inventory for existing products and then places the final order which will help in preventing wasteful purchases. Admin is a sole authorized individual managing most of the order placement, so we have assigned him username and password.	As the core entity of this database, admin is associated with several entities in our database. AdminID acts as a foreign key in Inventory. Manager will send the requisition to Admin to check the inventory so we have connected Admin Entity to Manger table. Once the requisition is approved by Admin, he can create an order for the same so we have connected Order table.
Ticket	One of the business rules for our Procurement Management System	This is one of the busiest entities in entire database. This table will witness most

	Database is each requisition will go through the approval workflow to efficiently manage organisation's resources and prevention of unauthorized purchase. This can be achieved by creating ticket for each request raised by the general staff and then allowing it to go through approval workflow. Ticket Entity will store information about procurement requests such as product name, quantity, staff who raised it and manager details. Manager details will include ManagerID and his approval status.	transactions in its lifetime. After successful approval workflow, requisition will be created against TicketID. This will contain foreign keys such ManagerID and DepartmentID which will be helpful for generating reports for tickets generated by each department and manager approval status for particular ticket. Tickets having manager approval status as approved will be created as requisitions
Requisition	To avoid wasteful purchasing as stated in the business rule, organisation will check the availability of each purchase before processing. For each ticket created in organisation, for procurement of products, requisitions will be created for approval by the admin team. Admin team will check the availability of the products against the organisation's inventory and if the product is not available in inventory then the requisition will be approved.	Requisition entity is directly related to the admin entity for its approval status. AdminID is a foreign key in requisition table which will store admin member's ID who has approved or rejected the requests depending on availability of the products in the inventory. It has a relation with the Manager entity and order entity which identifies who has raised this request and for which requisition the order has been placed respectively. Requisition entity is connected to Catalogue table via CatalogueID, which will be used to determine the item was selected from which catalogue.
Category	Category entity will contain broad description of products needed in the organization. This entity will list all categories such as Electronics, Stationary, Edibles etc. Since one category can have multiple products so we create an associative entity between category and product.	Category entity is related to product entity as each product belongs to particular category listed in category entity. As it is creating many-to-many relationship with product entity associative entity ProducCategory is created. Also each category will have a contract so we have connected it with Contract entity.
Product	Product entity will contain information of all products falling under the selected category along with its description. One supplier can have many products so we have used associative entity between category and supplier.	Product entity is one of the core entities in our database which will be in relation with other entities such as supplier and category. Each product belongs to one of the categories listed in category entity to manifest the same we have created associative entity between product and category. Product and category are creating many-to-many relationship. Also, each product has a supplier hence it is creating many-to-many relationship with supplier entity. We have created ProductSupplier associative entity
Catalogue	Catalogue will display all the available item based on the product selected by the user. This entity will display price, discount, supplier information, rating for	Catalogue entity is directly related to associative entity ProdcutSupplier entity for Product and SupplierID. Catalogue will have entry for each product-supplier entry in

	each item. Manager can select efficient supplier based on rating/available discount for each product offered by various suppliers. This is eventually help in achieving our mission statement of efficient use of organisations resources.	ProductSupplier entity. Catalogue is also related to Manager entity as before creating requisition, manager will view catalogue of each product and select supplier who is providing best offer.
Supplier	Organisation is also interested in tracking supplier information to create contract with efficient supplier based on offers they are providing of products. This entity will contain all the supplier information.	Supplier entity is related to product entity by creating many-to-many relationship. To avoid that we have created ProdcutSupplier as an associative entity. It is also in relation with contract entity as supplier will sign a contract with organisation for good evidentiary value.
Order	This is also one of the most widely used table in our database as each approved requisition will turn into purchase order. This entity will contain order information	Order entity is related to two main entities in our databases such as Admin and Invoice entity. Order entity contains AdminID as foreign key which is referencing Admin entity's primary key for its value. Also, Invoice will be created against each order hence they are related via OrderID.
Invoice	One of the business goals of our database is to track final spend of each department against approved budget. Invoice is created against each purchase order made by employees. It will track information like Price, Discount, Discounted amount and final amount. We will use this entity to analyze the total sent done by each department.	Invoice is directly related to Order entity as every order will have a invoice for it. They are related via OrderID
Contract	To achieve one of our business goals i.e. helping the management in creating a strategy to control spending for each department we need to choose efficient supplier and create long term contract. Contract entity is used to track contract detail between supplier and organisation.	Contract creation requires two parties and, in our database, contract will be created between department and suppliers. Contract entity has non identifying relationship with category entity as CategoryID is foreign key in the Contract table.
Expenditure	This entity will track expenditure of each department in the organisation. This will help us update the budget of each department accordingly.	Expenditure entity is in identifying relationship with department entity. Department entity contains budget information which can use to analysing expenses of each department.
Department	Department entity is one of the core entities in our procurement management database as organisation as it contains details like department budget. This attribute is important to satisfy our purpose to analyze the expense of each department and update the budget accordingly in future.	As the core entity of this database department entity associated with many entities in our database. Department entity is related to contract, expenditure and Manager entities. All the entities will retrieve department details and budget details from this entity.


Expenditure
PK FK
DepartmentId
INT
Amount
FLOAT
Has
Contract
PK
ContractId
INT
StartDate
DATE
EndDate
DATE
FK
SupplierId
INT
FK
DepartmentId
INT
FK
CategoryId
INT
Signs
Department
PK
DepartmentId
INT
Name
VARCHAR(45)
Budget
INT
Has
Has
Supplier
PK
SupplierId
INT
SuplierName
VARCHAR(45)
General Staff
PK
StaffId
INT
FK
EmployeeId
INT
FK
ManagerId
INT
FK
DepartmentId
INT
Designation
VARCHAR(45)
Raises
Ticket
PK
TicketId
INT
Subject
VARCHAR(45)
Description
VARCHAR(250)
ProductQuantity
INT
FK
StaffId
INT
ApprovalStatus
CHAR(10)
FK
ManagerId
INT
TicketDate
DATE
BasedOn
Category
PK
CategoryId
INT
Name
VARCHATR(45)
Description
VARCHAR(250)
ProductSupplier
PK FK
ProductID
INT
PK FK
SupplierId
INT
Manages
ReportsTo
Receives
Employee
PK
EmployeeId
INT
FirstName
VARCHAR(45)
LastName
VARCHAR(45)
Email
VARCHAR(45)
PhoneNumber
BIGINT(11)
SSN
VARCHAR(11)
Manager
PK
ManagerId
INT
FK
EmployeeId
INT
FK
DepartmentId
INT
Incentives
FLOAT
ProductCategory
PK FK
ProductId
INT
PK FK
CategeoryId
INT
Product
PK
ProductId
INT
Name
VARCHAR(45)
Description
VARCHAR(250)
Has
Generates
Catalogue
PK
CatalogueId
INT
Discount
FLOAT
Stock
INT
FK
ProductId
INT
FK
SupplierId
INT
ProductPrice
FLOAT
ProductRating
INT
Admin
PK
AdminId
INT
FK
EmployeeId
INT
Username
VARCHAR(50)
Password
BINARY(60)
Checks
Receives
Contains
Inventory
PK
InventoryId
INT
TotalStock
INT
ProductId
VARCHAR(45)
FK
AdminId
INT
Requisition
PK
RequisitionId
INT
FK
ManagerId
INT
DepartmentID
INT
FK
AdminId
INT
RequisitionDate
DATE
FK
CatalogueId
INT
FK
TicketId
INT
Places
Invoice
PK
InvoiceId
INT
FK
OrderId
INT
ActualPrice
FLOAT
Discount
FLOAT
DiscountAmount
FLOAT
FinalAmount
FLOAT
PaymentMode
VARCHAR(45)
CreatedDate
DATE
Generates
Order
PK
OrderId
INT
OrderDate
DATE
OrderDetails
VARCHAR(250)
TotalQuantity
INT
FK
AdminId
INT
FK
RequisitionId
INT