A Database Management System

for

Books-A-Thousand Bookstores

CSC 540 Database Management concepts and Systems

Project Report #1

Team Rdm
Abhay Gupta(agupta15), Rohit Kandhal(rkandha), Sagar Jauhari(sjauhar), Raghav Tripathi(rtripat)

List of assumptions taken for Books-A-Thousand store

1. Warehouse of the bookstore has merchandise purchased only from the vendors.
2. Each merchandise is supplied to the store by only one vendor.
3. Books-A-Thousand buys merchandise (i.e. warehouse stock) only from vendors who are under contract.
4. Store has salesperson who helps customers with merchandise selection and once customer selects, salesperson can swipe that merchandise for further order processing.
5. Billing cycle for each vendor and each customer is first day of month.
6. There is only one warehouse for the complete chain of Books-A-Thousand stores.
7. Age of a person (staff, customer) is not kept as a separate attribute but calculated from date of birth.
8. Isolation level for the database system is set to serializable.
9. When a particular order is shipped by the vendor then we assume that order will be received safely by the store and warehouse database will be updated with the merchandises received.
10. Merchandise purchased from a vendor is sold at some profit by Books-A-Thousand thus prices of a merchandise sold by vendor and sold by Books-A-Thousand store may vary.
11. Purchased made by a customer can be assisted by atmost one salesperson.
12. Usage record for vendor or customer is not specifically maintained as a separate entity in database. All orders placed by customer or vendor will be in customer order or vendor orders, and after successful completion of an order these would be used to retrieve usage required.
13. Merchandise ID in the warehouse will remain same as that shipped by the vendor i.e. No two vendors can use the same merchandise ID for their books.
14. A store doesn't maintain any inventory of its own instead all merchandise are fetched from the central warehouse.

Added functionalities (if time permits):

1. A customer can make payment of any amount. It may be less than his outstanding balance or more than that. Extra credit will be adjusted in next month billing cycle.
2. A customer can be a preferred customer (as decided by the publishing house) and all preferred customers will be given a specific discount on each purchase.
3. Generate book recommendation list based on purchase history for a specific age group.
4. List of top ten bestsellers books.
5. Monthly discount on specific books.

1. **Problem description and why database is better than other storage like a simple set of files for this task**

   The database system being designed in this project is for management of a chain of bookstores named “Book-A-Thousands”. Like many modern bookstores, Book-A-Thousands requires the database system to perform simple Custom Relationship Management(CRM) and Warehouse management operations. The database stores information about customers who have a Book-A-Thousands account, the orders placed by a customer and maintain billing and payment information for a customer. Moreover, the database should also store information about contracted vendors of the chain, enable the Book-A-Thousands warehouse to place order with vendors, manage vendor payments and store merchandise available at the warehouse. It will also maintain information about employees of a Book-A-Thousand bookstore.

   A database is a better idea than a file system because:

   1. **Difficulty in accessing data:** If we chose to store data in files, then each file will have data particular to an entity like customer, vendor, orders etc and becomes very difficult to access information which require correlation between 2 entities like “How many orders has a customer placed and what is their status?”. While the real power of relational database lies in ability to link and extract information from multiple tables to answer specific questions.
   2. **Data redundancy and inconsistency:** Storing data in files might lead to duplication of data across files and cause update and delete anomaly. A database handles such situations very well.
   3. **Integrity problems:** Database can enforce consistency constraints on the data stored in it. Example, when the status of a vendor order changes to “Received”, the merchandise tuple must be added to the warehouse contents. This would be very difficult to maintain in a file based system
   4. **Concurrent-access anomalies:** Database is built to allow multiple users to both read and update data simultaneously through use of transactions. Whereas while using a file based storage system, it will be the onus of the programmer to allow concurrent execution by usage of reader and writer locks which can be exclusive to a user.

2. **Intended classes of users:**

   a. **Billing Staff**: staff responsible for billing of customers
   b. **Sales Department**: staff responsible for assisting customers in buying books
   c. **Payroll Department**: staff managing the accounts of employees.
   d. **Accounting Department**: store account management like vendor contract information, policies etc.
   e. **Business Intelligence Department**: For analysing the data/transaction made through the store and working out strategies to promote business.
   f. **Inventory and warehousing Department**: Maintaining data about merchandise stored in warehouse, tracking vendor orders etc.

3. **Five things about which we need to store information:**

    a. **Customers**: Store customer name, date of birth, gender, address, phone number etc
    b. **Vendor**: Store Vendor Name, contact information, orders to a vendor, payments made to a vendor
    c. **Merchandise**: Store merchandise name, price, author etc
    d. **Billing & Order information**: Stores and tracks customer orders, vendor orders, payment made by customer, payments made to a vendor
    e. **Staff:** Store information about staff working in a Books-A-Thousand store, their name, contact information, job title, department name etc.

4. **Realistic situation where using the database system will require to perform two operations Placing a new order and order cancellation**:

    For a new customer, new account will be created with the basic information about the customer by the customer accountant and customer can request for an order of one or more merchandise to the executive (*Information Processing)*. Billing of that order will be automatically added to that customer's account using the customer details present in user account (*Maintaining Billing account)*. In the end of monthly billing cycle customer will be notified about the pending payment and due date for the payment. Moreover, a user can request for the cancellation of a particular merchandise and that order will not be shipped and billed to the customer. Existing customers can simply request for a particular merchandise and system will automatically retrieve shipping and billing details from the account details.

5. **APIs required for each Tasks and Operations listed in specification.**

Operation	**Function name**	**Input**	**Output**
Information Processing	Enter / update staff information	Employee personal information, role, salary, date of joining,	New/updated employee record in database
	Enter / update customer information	Customer personal information including contact details, shipping address, billing information	New/updated customer record in database
	Enter / update contracts	vendor details, contract details	Successful contract information insertion/update
	Enter/update warehouse information	books catalogue including quantity of books available	New/updated warehouse database
	Enter/ update vendors information	Vendor contact details, books catalogue available with that vendor, billing information	New/updated vendor record in database
	Delete staff/ customer/ contract/vendor/ merchandise in warehouse record	staff/ contract/ customer/ vendor/ merchandise to be deleted from warehouse (specifically merchandises to be deleted)	Mentioned entry deleted from the database
	Check Available Books	a) No parameter or b) Specific book	a) All available books in warehouse or b) Specific book available in warehouse or not
	Order book	Book(s) detail(s), customer details, salesperson assisted	Order for book placed, billing record added to customer, warehouse database updated
Vendor usage record	Enter purchase record	Vendor details, merchandise list (or order from vendor details)	Order added to Vendor billing information.
	Update purchase record	vendor details, merchandise list (or order from vendor details)	Vendor billing record updated
	Cancel purchase record	Vendor purchase order id	Deletes the specified vendor purchase order and successfully updates billing record of vender.
	Show purchase record	vendor id, time period	shows the purchases made from the specified vendor over a specified time

			period
Maintaining billing accounts	Generate bill for customer	customer id, billing month(optional)	Generate bill for the specified month(by default last month) for the specified customer
	Generate bill for vendor	vendor id, billing month(optional)	Generate bill for the specific month(by default last month) for the specified vendor
	Report/maintain payment for customer	Customer id, amount,date	Update billing record for a customer with payment details
	Report/maintain payment for vendor	Vendor id, amount, date	Update billing record for a vendor with payment details
Reports	Purchase history for a customer	customer id, period	Generate purchase history over a period for a given customer.
	Salesperson report	salesperson id, time period	Generate report of all the customer assisted by a particular salesperson during a time period
	Get all vendor details	storename (books-a-thousand in this case)	Generate report of all the vendors details, books-a-thousand has contract with.
	Get staff details based on role	a) Role or b) No parameter	a) Returns details about all the staff with the specified role or b) All staff grouped by the role

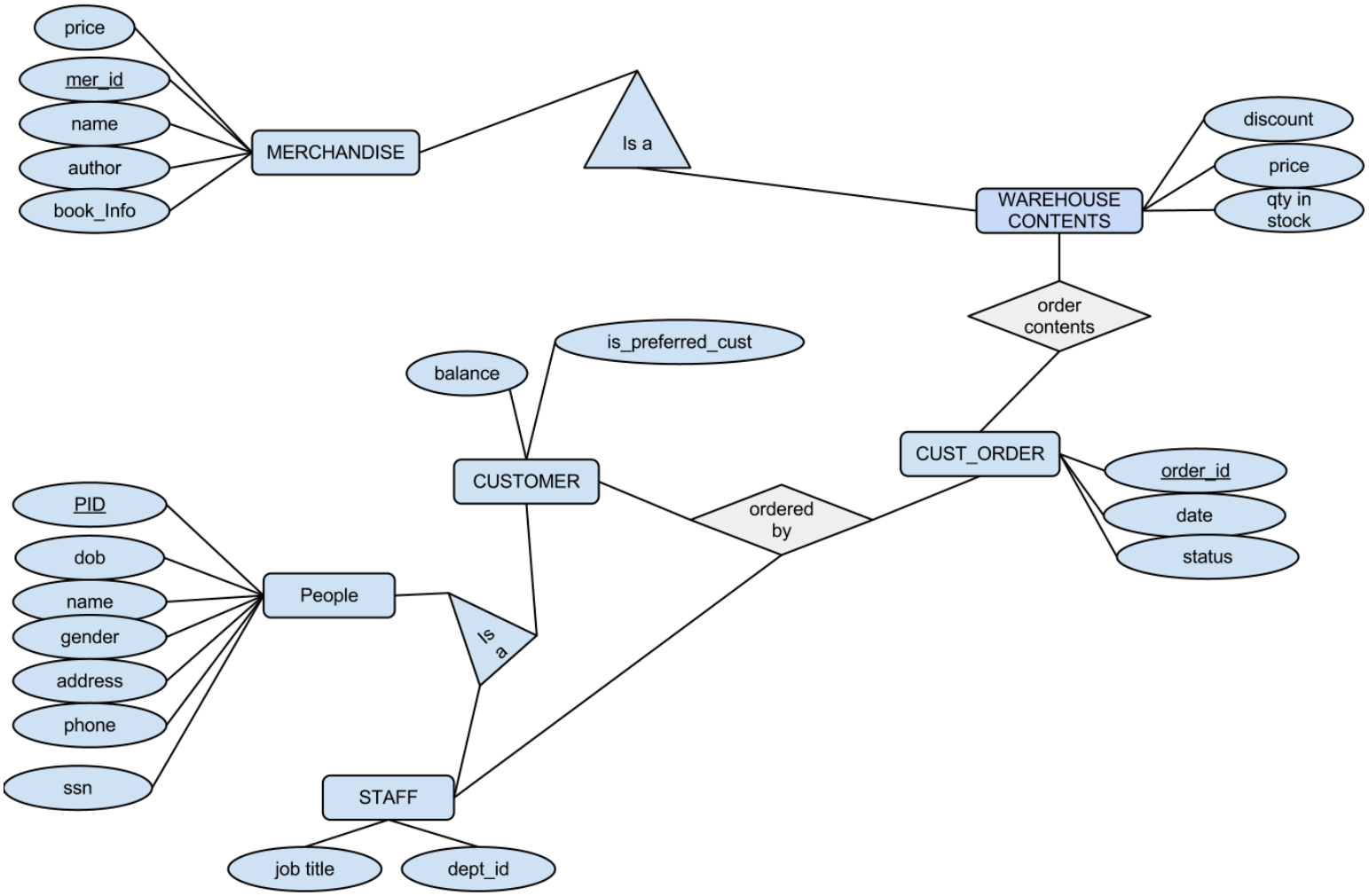
6. **Views data description that correspond to the intended classes of users**

View	**Data**
Salesperson View	The salesperson view contains the account information of customers and all the purchases made by customers helped by the particular salesperson.
Helpdesk Executive View	Customer information, order and billing history
Vendor Accountant View	Vendor account information, purchases made, Billing details and history
Customer Accountant View	Customer account information, books available, vendor information(in case book is not in stock), billing details
Store Manager View	Employee details (including salaries, contact information, contact details, roles), Customer details. He has access to entire database
Staff View	Employee basic information (for employees to see information about each other) like name, address, phone_number, job role and department id.

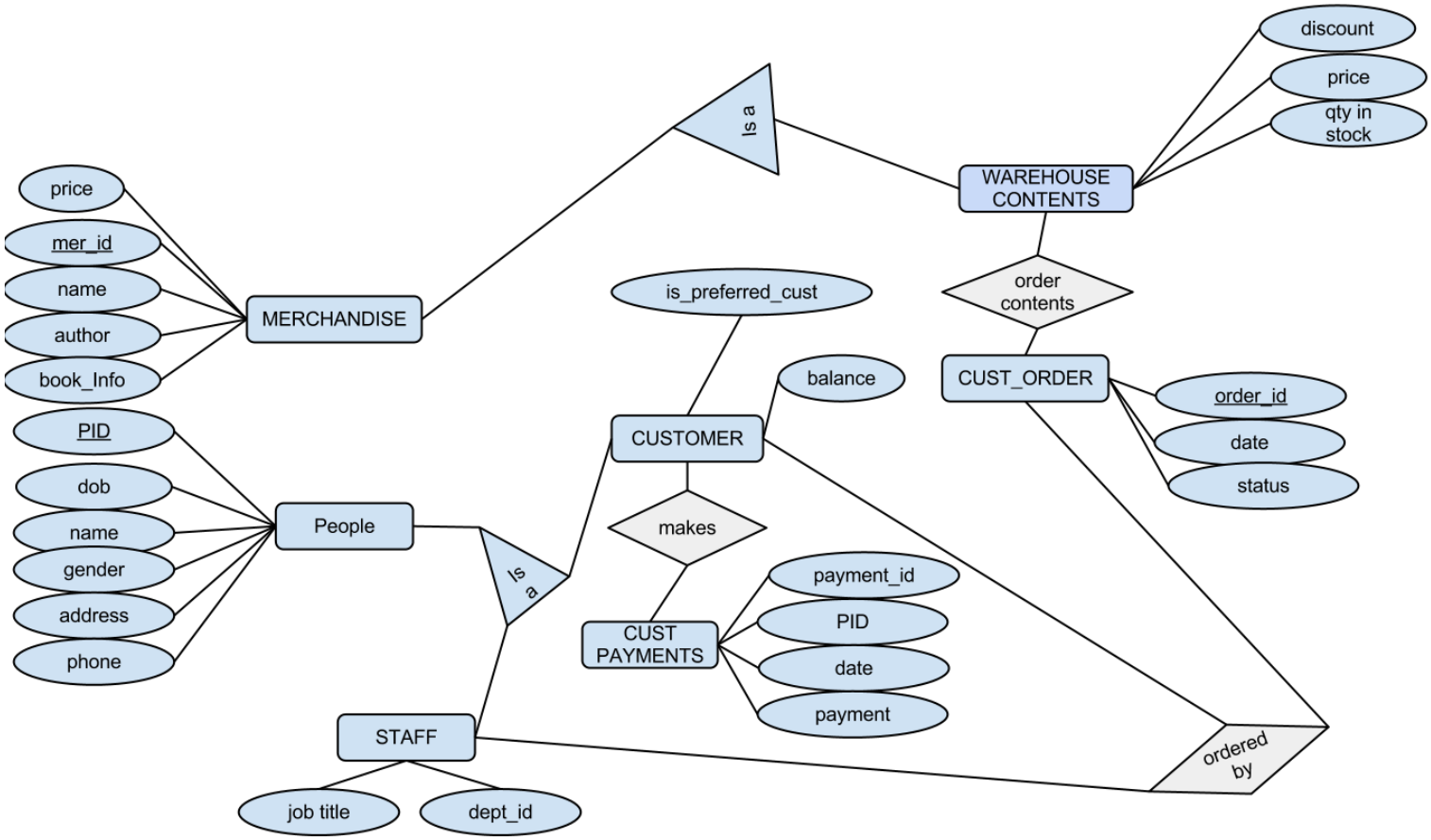
It might be possible that a person may have multiple views depending on its role like a customer accountant has both accountant view as well as staff view.

7. **Local E/R diagrams for each view listed in item 6 with relevant database information**

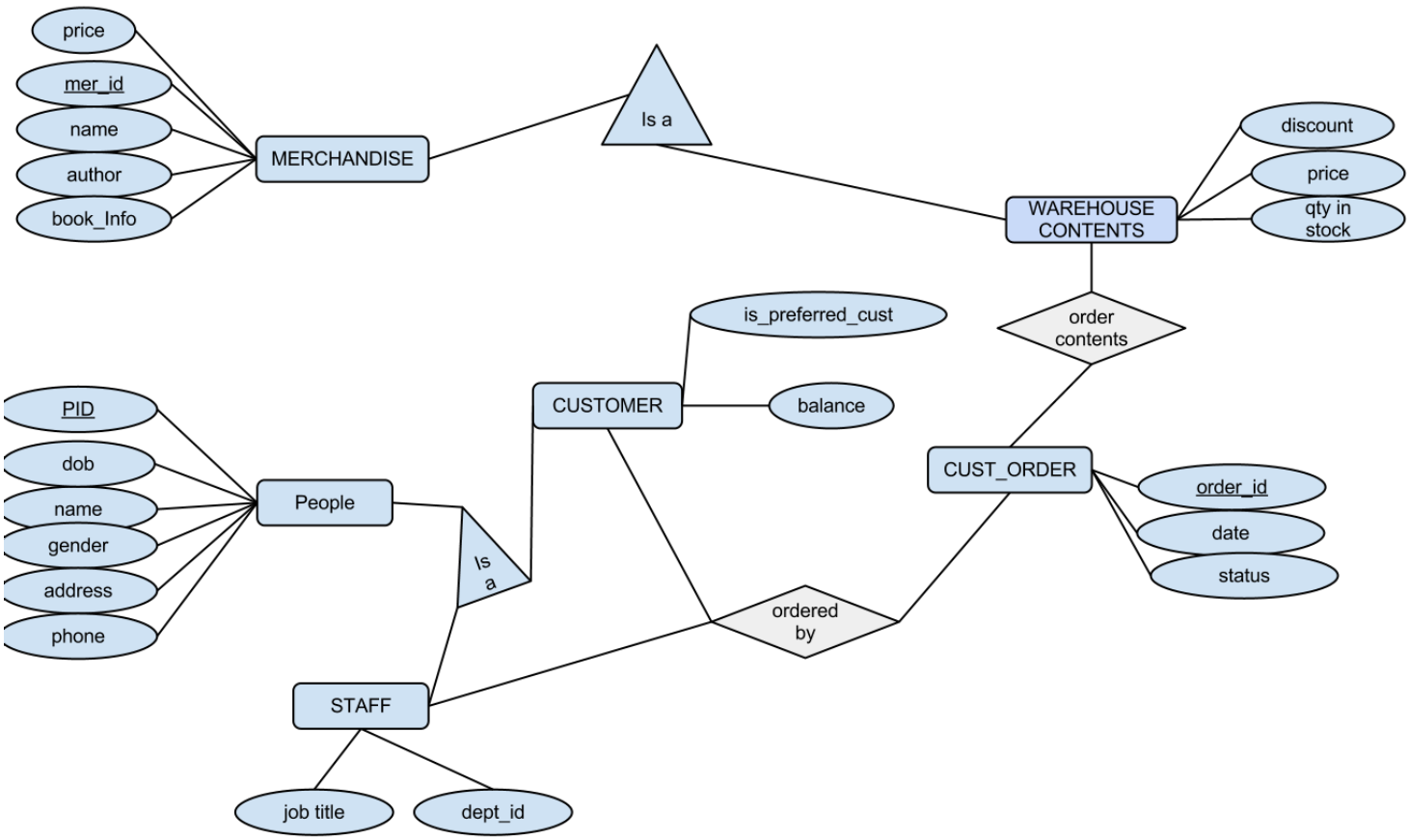
Customer Accountant View



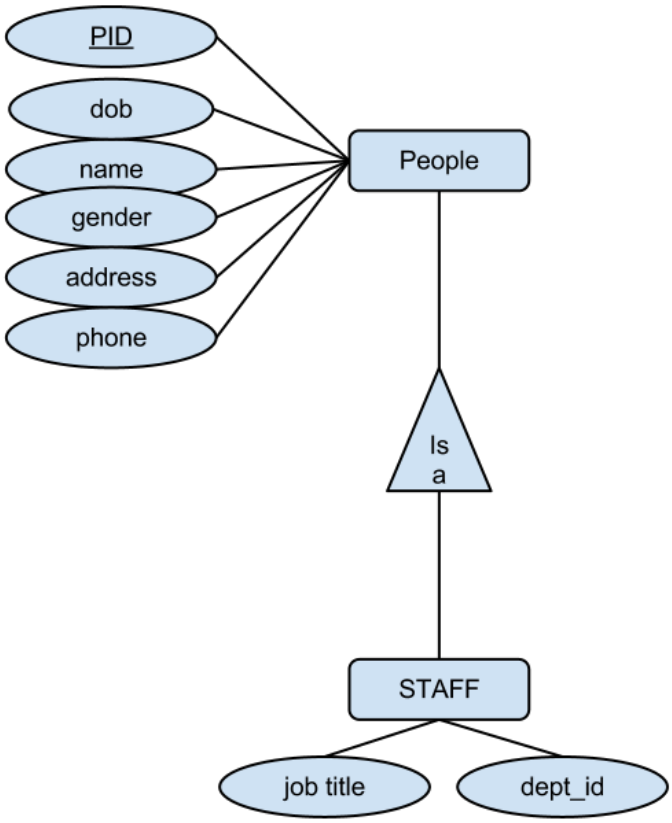
HelpDesk Executive View



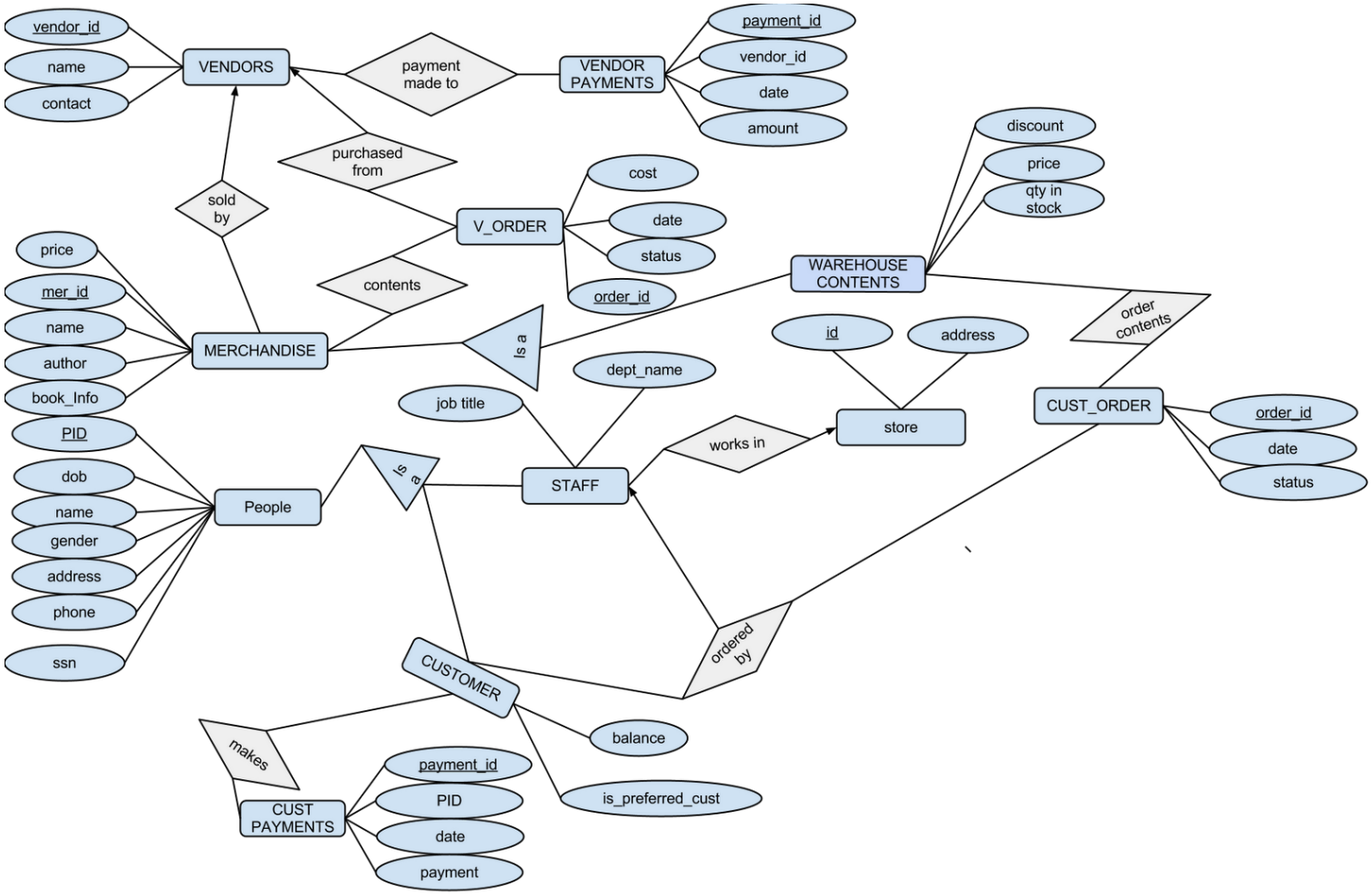
SalesPerson View



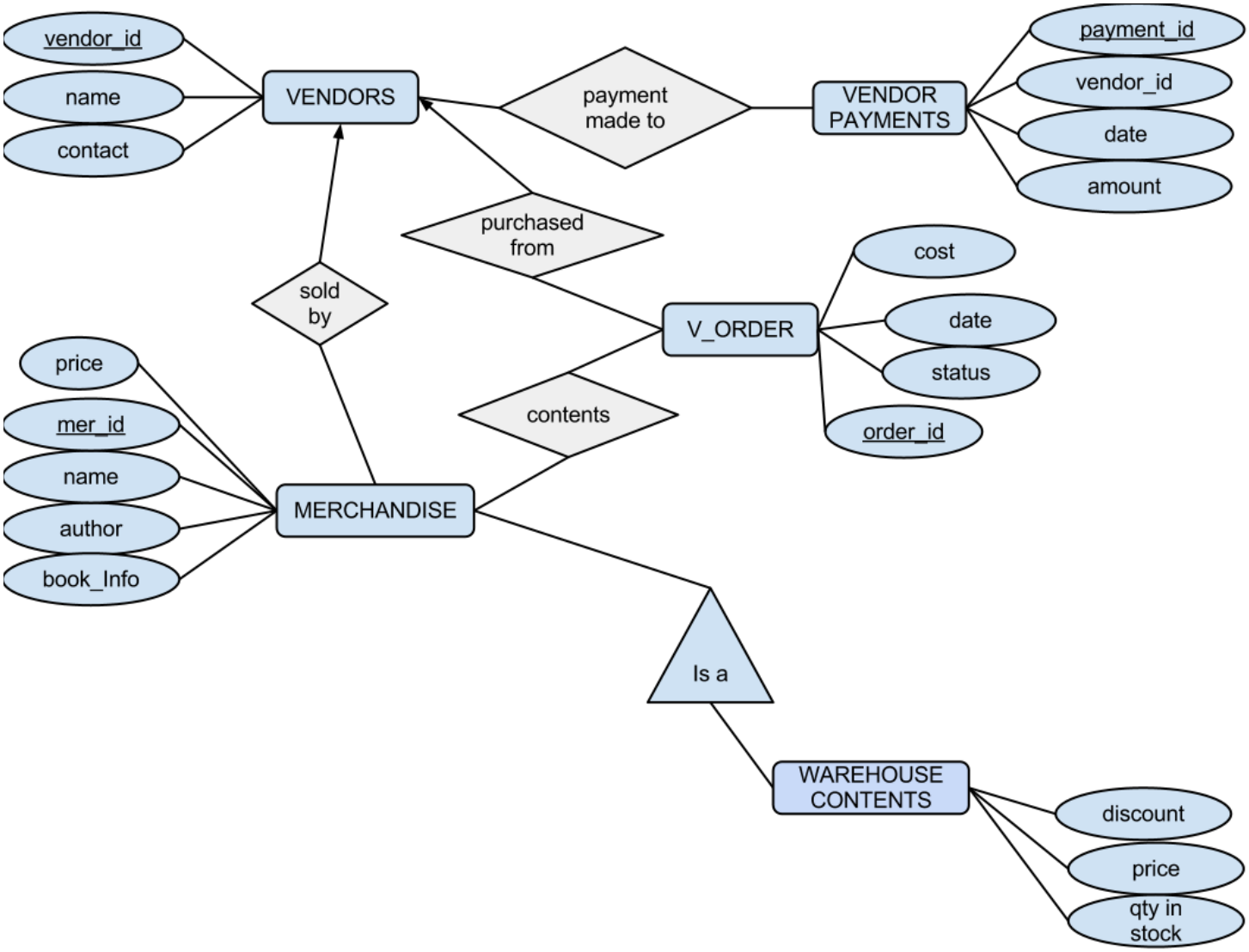
Staff View


PID
dob
name
gender
address
phone
People
Is
a
STAFF
job title
dept_id

StoreManager View



Vendor Accountant View



8. **E/R diagram description with design decisions**

The people involved with the bookstore are manager, customer accountant, staff, vendors and vendor accountant and helpdesk persons.

Entities such as vendors , vendor payments, merchandise, warehouse contents ,customers, staff are created which would allow better searching and lower the redundancies.

Staff and Customers are inherited entities of people record. This was done because it is more meaningful and reduces data redundancy.

Customers entities have attributes of preferred customers and balance which keeps track of the most valuable customers of the bookstore and credit balance that the customer has.

We made several decisions regarding relationship types and constraints:

a. sold_by: a merchandise can be sold by one and exactly one vendor.
b. purchased_from: an order is purchased from an exactly one vendor
c. order_contents: Keeps track of the customer orders and its present status, also used in getting usage records of customers
d. contents: Keeps track of vendor order, its present status and, also used in getting usage records of vendors.
e. makes: It keeps track of the payments made by the customer till date and could be used in case of cancellation of orders
f. ordered_by: It keeps track of the staff who had helped customers in selecting a merchandise and is also used by the bookstore to obtain information about staff performance.

9. **Local relational schema using the local E/R diagrams listed in item 7**

Manager View

a. Vendors(vendor_id,name, contact)
b. Merchandise(mer_id,c_price,name,author,book_info)
c. sold_by(vendor_id,mer_id)
d. vendor_payments(payment_id,vendor_id,date,amount)
e. payment_made_to(vendor_id,payment_id)
f. v_order(order_id,status,date,cost)
g. purchased_from(vendor_id,order_id)
h. contents(order_id,mer_id)
i. warehouse_contents(mer_id,discount,s_price,qty_in_stock,c_price,name,author, book_info)
j. order_contents(mer_id,order_id)
k. cust_order(order_id,date,status)
l. ordered_by(order_id,pid_staff,pid_cust)
m. people(pid, date_of_birth, name, gender, address, phone, ssn)
n. staff(pid, job_title, dept_name)
o. customer(pid, is_preferred_cust, balance)
p. cust_payments(payment_id, people_id, date, payment_amount)
q. makes(pid,payment_id)
r. store (id,address)
s. works_in(pid,id)

HelpDeskExecutive View

a. Merchandise(mer_id,c_price,name,author,book_info)
b. warehouse_contents(mer_id,discount,s_price,qty_in_stock,c_price,name,author, book_info)
c. order_contents(mer_id,order_id)
d. cust_order(order_id,date,status)
e. ordered_by(order_id,pid_staff,pid_cust)
f. people(pid, date_of_birth, name, gender, address, phone, ssn)
g. staff(pid, job_title, dept_name)
h. customer(pid, is_preferred_cust, balance)
i. cust_payments(payment_id, people_id, date, payment_amount)
j. makes(pid,payment_id)

Vendor_Accountant View

a. Vendors(vendor_id,name, contact)
b. Merchandise(mer_id,c_price,name,author,book_info)
c. sold_by(vendor_id,mer_id)
d. vendor_payments(payment_id,vendor_id,date,amount)
e. payment_made_to(vendor_id,payment_id)
f. v_order(order_id,status,date,cost)
g. purchased_from(vendor_id,order_id)
h. contents(order_id,mer_id)
i. warehouse_contents(mer_id,discount,s_price,qty_in_stock,c_price,name,author, book_info)

Customer_Accountant View

a. Merchandise(mer_id,c_price,name,author,book_info)
b. warehouse_contents(mer_id,discount,s_price,qty_in_stock,c_price,name,author, book_info)
c. order_contents(mer_id,order_id)
d. cust_order(order_id, date, status)
e. ordered_by(order_id,pid_staff,pid_cust)
f. people(pid, date_of_birth, name, gender, address, phone, ssn)
g. staff(pid, job_title, dept_name)
h. customer(pid, is_preferred_cust, balance)

Staff View

a. people(pid, date_of_birth, name, gender, address, phone, ssn)
b. staff(pid, job_title, dept_name)
c. customer(pid, is_preferred_cust, balance)

10. **Local schema documentation and design decisions.**

A mechanical approach was used to convert the E-R diagram into relations with a possible room for future adjustments.

a. Entity sets become relations with the same set of attributes.
b. Relationship was converted into relation with attributes that of the relationship and with the keys of the participating entity sets.

E-R approach was used to convert subclass of into relations. A relation attribute ‘balance’ was added to customer to maintain the credit information of the customer. A customer can pay an amount greater than his due and the remaining amount would be used as his credit for his next transactions. For the fact that there is a chain of bookstore available, an entity store has been created. Warehouse has been created as a subclass of merchandise, as it is a superset of warehouse.