

GROUP 1: KIRK: Key Information Record Keepers

CERVALES, YUANN CZEDRIEHCK D.

DELA VEGA, ANGELO S.

RELOJ, REIGN JOHN N.

TALOSIG, JHUN FRANCIS M.

ACTIVITY

QUESTION 1: THE DATA DESIGN PROCESS

QUESTION 1: THE DATA DESIGN PROCESS (25 points).

Imagine that you have been assigned to a team that will be developing an inventory tracking system. As part of the project startup, your manager has asked each team leader to bring a basic work plan to the next meeting. At that meeting, these work plans will be analyzed to determine the overall project timeframe, costs, personnel requirements and software requirements.

For now, as the team leader for the data design team, you have been asked to bring a work plan that identifies the phases of data design and includes the following information for each phase:

- a). a description of the data design phase,
- b). the inputs of the phase,
- c). the outputs of the phase,
- d). a key issue addressed in the phase
- e). a challenge that you can anticipate would occur in the phase.

Please prepare the response you will bring to the meeting.

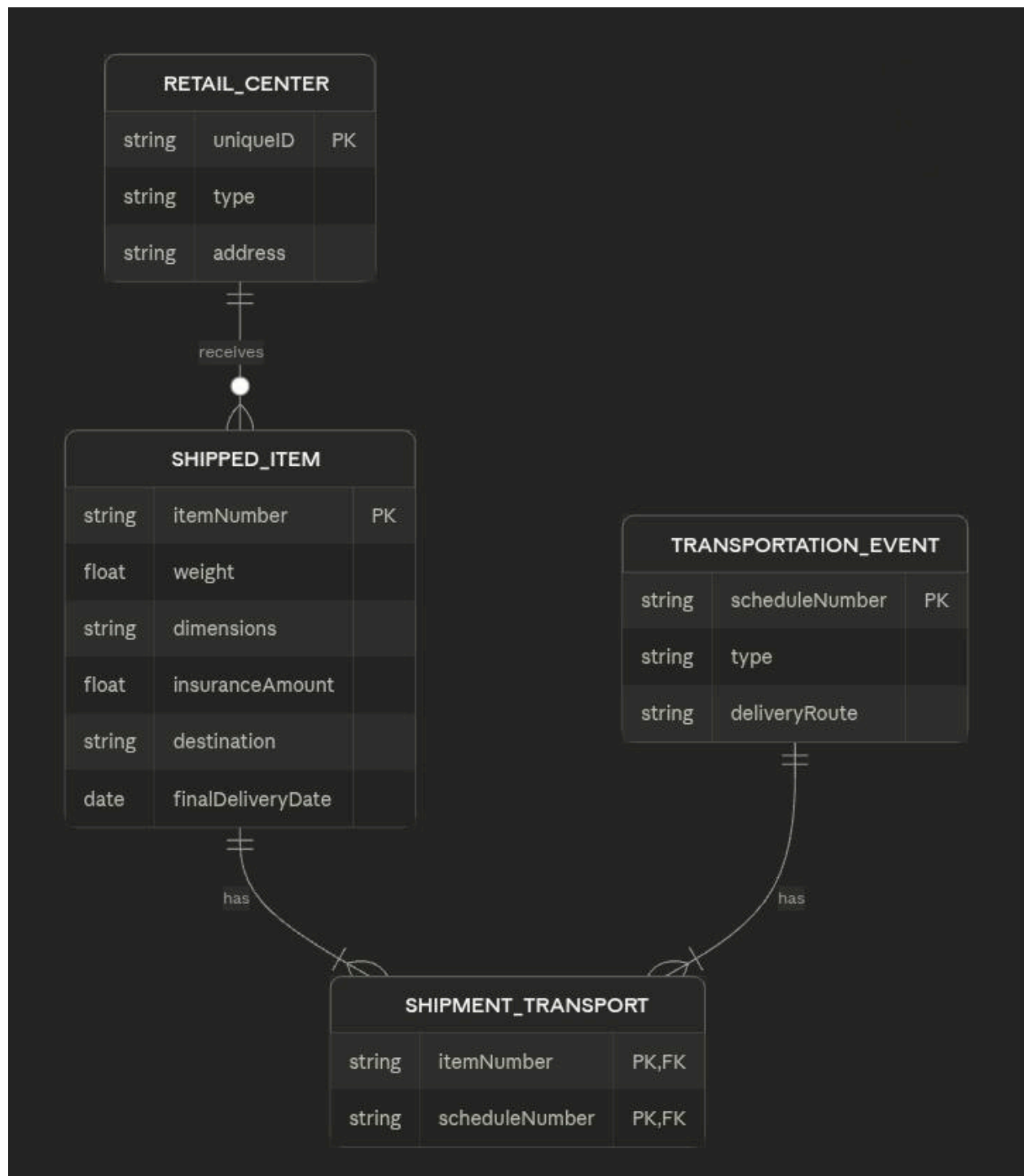
Description	Issue	Input	Output	Challenge
a. <u>Conceptual Design</u> Create model that captures major entities, relationships among entities, and attributes of entities required for a particular system.	- Capturing all data - Capturing relationships - Data integrity	- Functional specs - General understanding of problem	- ER diagram	None
b. <u>Logical Design</u> Transform the major entity/attribute /relationship requirements into high level specification for database	- Providing location for all data - Data integrity	- ER diagram	- Relational database schema	None
b2. <u>Improving Logical</u>	- Minimizing	- Relational	- Relational	None

Design Improve the high-level database specification.	redundancy - Minimizing ambiguity	database schema	database schema	
c. <u>Physical Design</u> Transform the high-level specifications for database into detailed specifications for how to construct actual database in a specific relational database software.	- Performance - Data integrity	- Relational database schema - Meaning of data	- Technical specifications for construction of the database	None

QUESTION 2: CREATING AN ENTITY-RELATIONSHIP DIAGRAM (25 points).

UPS prides itself on having up-to-date information on the processing and current location of each shipped item. To do this, UPS relies on a company-wide information system. Shipped items are the heart of the UPS product tracking information system. Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the UPS system at a single retail center. Retail centers are characterized by their type, uniqueID, and address. Shipped items make their way to their destination via one or more standard UPS transportation events (i.e., flights, truck deliveries). These transportation events are characterized by a unique scheduleNumber, a type (e.g, flight, truck), and a deliveryRoute.

Please create an Entity Relationship diagram that captures this information about the UPS system. Be certain to indicate identifiers and cardinality constraints.



Entities: `RETAIL\_CENTER`, `SHIPPED\_ITEM`, `TRANSPORTATION\_EVENT`, and the bridge table `SHIPMENT\_TRANSPORT`.

Relationships:

- `RETAIL\_CENTER` → `SHIPPED\_ITEM` — one-to-zero-or-many (`||--o{`), meaning a center may receive any number of items, but each item goes to exactly one center.

- `SHIPPED\_ITEM` → `SHIPMENT\_TRANSPORT` — one-to-one-or-many (`||--|{`), resolving the M:N on this side.
- `TRANSPORTATION\_EVENT` → `SHIPMENT\_TRANSPORT` — one-to-one-or-many (`||--|{`), resolving the M:N on the other side.
- The composite PK on `SHIPMENT\_TRANSPORT` (`itemNumber` + `scheduleNumber`) is marked with both `PK` and `FK` to show it serves dual roles as the bridge key.

Question #3

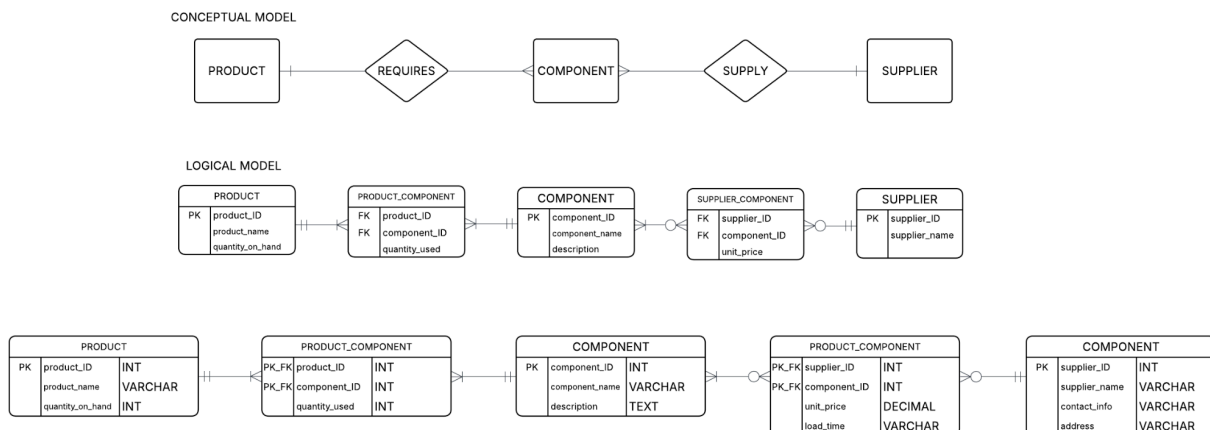
Manufacturer

A manufacturing company produces products. The following product information is stored: product name, product ID and quantity on hand. These products are made up of many components. Each component can be supplied by one or more suppliers. The following component information is kept: component ID, name, description, suppliers who supply them, and products in which they are used.

Assumptions

- A supplier can exist without providing components.
- A component does not have to be associated with a supplier.
- A component does not have to be associated with a product. Not all components are used in products.
- A product cannot exist without components.

List all the business rules, business objects, characteristics and relationships. Convert it to Conceptual then Logical and the Physical using an ERD to show, how you would track this information. Show entity names, primary keys, attributes for each entity, relationships between the entities and cardinality.



Business Objects (Entities)

The system contains three main business objects: Product, Component, and Supplier.

Business Rules

1. A product must be made up of one or more components — a product cannot exist without components.
2. A component can be used in many products, but it does not have to be used in any product.
3. A component can be supplied by one or more suppliers, but it does not have to be associated with any supplier.
4. A supplier can supply one or more components, but a supplier can exist without providing any components.
5. A component is not required to have a supplier.
6. A product is not required to be associated with a supplier directly — the relationship between products and suppliers is indirect, through components.

Characteristics (Attributes)

Product: product_id (PK), product_name, quantity_on_hand

Component: component_id (PK), component_name, description

Supplier: supplier_id (PK), supplier_name, contact_info

Product_Component (associative): product_id (FK), component_id (FK), quantity_used

Supplier_Component (associative): supplier_id (FK), component_id (FK), unit_price

Relationships and Cardinality

Relationship	Type	Description
Product — Component	Many-to-Many (mandatory on Product side)	A product requires one or more components; a component may appear in zero or more products
Component — Supplier	Many-to-Many (optional on both sides)	A component may be supplied by zero or more suppliers; a supplier may supply zero or more components

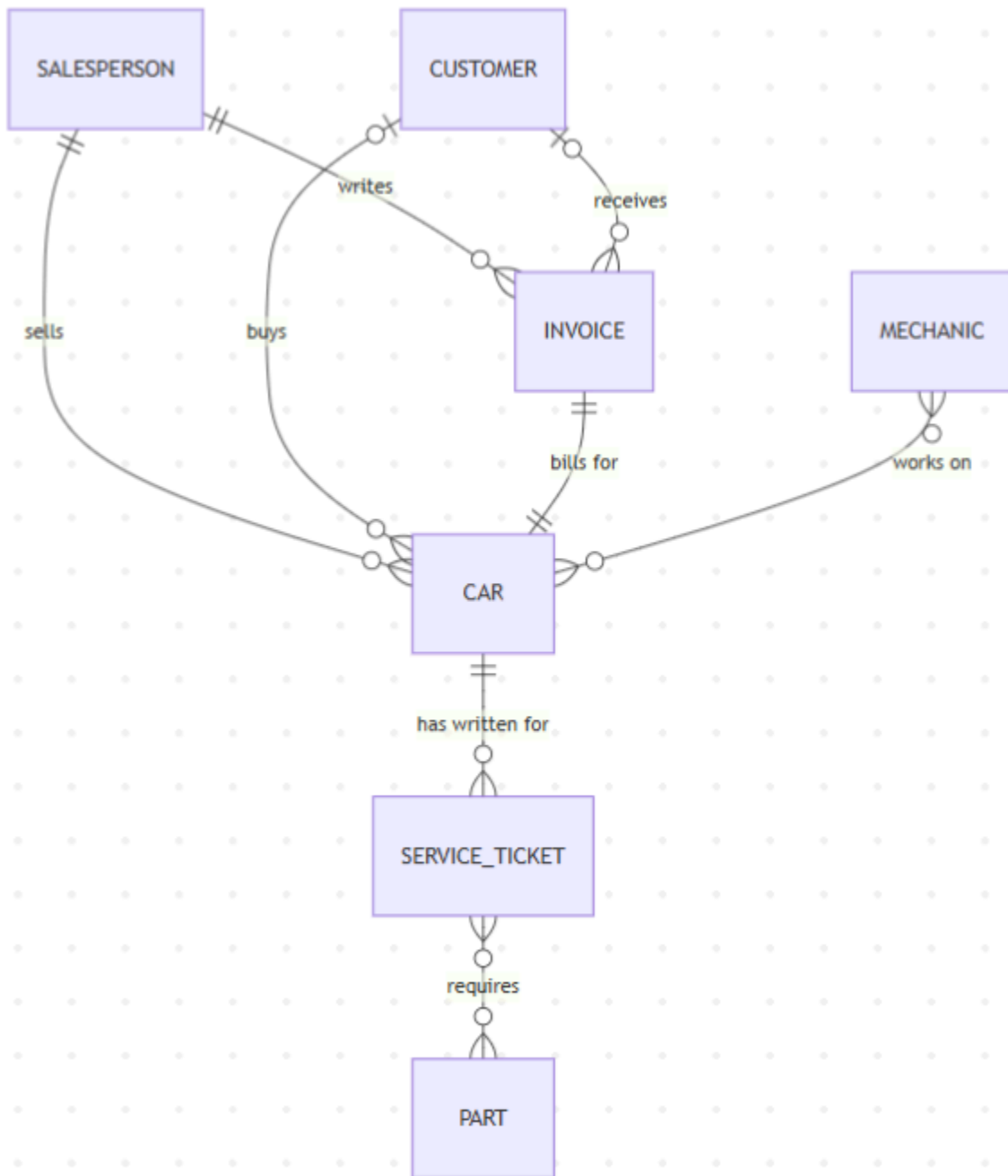
Question # 4: Car Dealership

Create an ERD for a car dealership. The dealership sells both new and used cars, and it operates a service facility. Base your design on the following business rules:

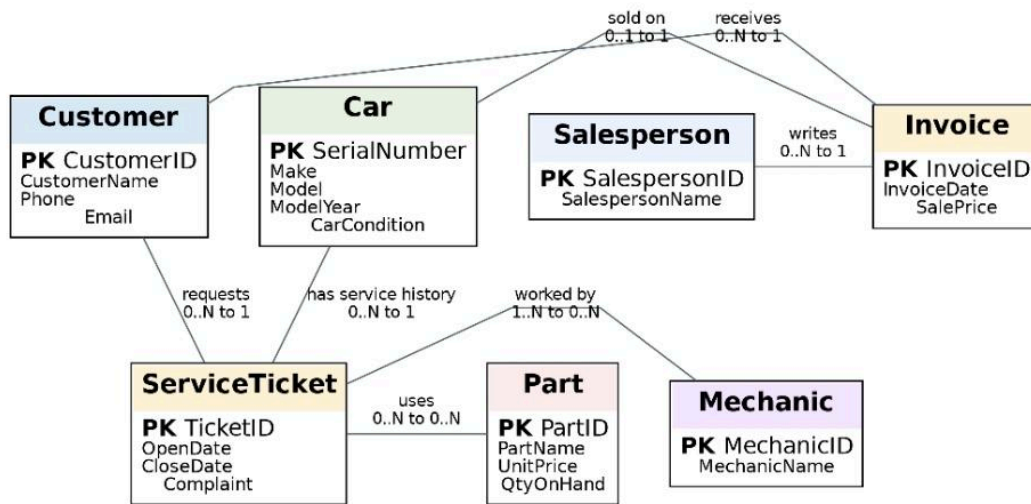
- A salesperson may sell many cars, but each car is sold by only one salesperson.
 - A customer may buy many cars, but each car is bought by only one customer.
 - A salesperson writes a single invoice for each car he or she sells.
 - A customer gets an invoice for each car he or she buys.
 - A customer may come in just to have his or her car serviced; that is, a customer need not buy a car to be classified as a customer.
 - When a customer takes one or more cars in for repair or service, one service ticket is written for each car.
 - The car dealership maintains a service history for each of the cars serviced. The service records are referenced by the car's serial number.
 - A car brought in for service can be worked on by many mechanics, and each mechanic may work on many cars.
 - A car that is serviced may or may not need parts (e.g., adjusting a carburetor or cleaning a fuel injector nozzle does not require providing new parts).
-
- List all business objects, characteristics and relationships. Convert it to Conceptual then Logical and the Physical using an ERD. Show entity names, primary keys, attributes for each entity, relationships between the entities and cardinality.

Business Objects Entities	Business Objects Characteristics	Relationship Between Business Objects	Relationship Constraints
Customer	CustomerID, CustomerName, Phone, Email	Requests ServiceTicket (1:M); Receives Invoice (1:M)	Customers can exist without any car purchases or service history.
Salesperson	SalespersonID, SalespersonName	Writes Invoice (1:M)	Every Invoice must reference exactly 1 Salesperson.
Car	SerialNumber, Make, Model, ModelYear, CarCondition	Appears on Invoice (1:1); Has ServiceTicket (1:M)	CarCondition must be 'NEW' or 'USED'. SerialNumber is the Primary Key.
Invoice	InvoiceID, InvoiceDate, SalePrice, SerialNumber, CustomerID, SalespersonID	(Referenced by Salesperson, Customer, and Car)	UNIQUE constraint on SerialNumber (one invoice per sold car). All FKs are NOT NULL.
ServiceTicket	TicketID, OpenDate, CloseDate, Complaint, SerialNumber, CustomerID	Assigned to Mechanic (M:N); Uses Part (M:N)	Must reference 1 Customer and 1 Car. TicketID is the Primary Key.
Mechanic	MechanicID, MechanicName	Assigned to ServiceTicket (M:N)	Resolved via SERVICE_ASSIGNMENT associative table.
Part	PartID, PartName, UnitPrice, QtyOnHand	Used by ServiceTicket (M:N)	Resolved via SERVICE_PART associative table.
Service_Assignment	TicketID, MechanicID, LaborHours	Links ServiceTicket to Mechanic	Composite PK (TicketID, MechanicID). Tracks specific labor hours.
Service_Part	TicketID, PartID, QuantityUsed	Links ServiceTicket to Part	Composite PK (TicketID, PartID). QuantityUsed must be > 0.

Conceptual Model



Logical Model



Physical Model

