**Business Name: Aurora Voyagers**

**Product: Luxury Spaceships**

**Introduction:**

The business we have chosen is a spaceship dealership called Aurora Voyagers. The business sells luxury class spaceships. The business also offers maintenance and repair services, along with replacement parts for the spaceships it sells. The customers of the business are those with a higher income, typically defined as upper-middle class or upper-class citizens, due to the high cost of the spaceships being sold. The business currently uses Microsoft Excel files to keep track of its customer information, along with the sales and inventory data. Each department of the business keeps their files in separate network drives, which are maintained by an internal IT department.

One challenge the business faces due to the current system is duplicated data since each department does not have access to records created by the other departments. Another challenge the current system creates is an inability to easily track sales and inventory data, causing delays or lack of inventory. A third challenge the business faces is reports and spreadsheets all need to be manually updated, leading to reduced efficiency during busy periods or when staffing is reduced.

The database we will be creating will have two parent entities, which will be Customers and Spaceships. The database will also have three child entities, which will be Orders, Payments, and Aftermarket (Parts/Maintenance).

**Questions for an Upper-Level Manager:**

1. What information do you need most often to make decisions about inventory and sales performance?
2. How do you currently track and forecast demand for different spaceship models?
3. What kind of customer or sales data would help you improve long-term business strategy?
4. What qualifies a spaceship for resale in terms of age, condition, and legal status?
5. How do we structure pricing, financing, and trade-in policies to remain competitive while maximizing profit margins?
6. What authentication, inspection, and documentation processes must be followed before a sale is finalized?

**Questions for a Low-Level Employee:**

1. What challenges do you face when trying to find details about available spaceship models or their specifications?
2. How do you currently record sales or maintenance requests from customers, and what problems do you encounter?
3. When a customer asks about their past orders or service history, how do you find that information?
4. What steps do I need to follow when checking in a newly delivered spaceship to make sure it meets company standards?
5. Who do I report to if I find damage, missing parts, or suspicious modifications on a ship?
6. Am I allowed to negotiate prices or offer discounts to customers, or does that need manager approval?

Answers:

1. We would need sales data by spaceship model.
2. Last Year’s monthly average compared to current year’s monthly average
3. Year-over-year customer and sales purchase history and financing requests.
4. Model/Manufacturing year is not less than 5 years old, must pass a full mechanical inspection and no legal restrictions on resale.
5. The pricing, financing, and trade-in policies are structured similarly to the automobile market, which is based on condition, age, and market demand, but on a larger, galactic scale.
6. The spaceship must pass a thorough mechanical inspection, all sales paperwork must be reviewed by a manager or auditor, and background checks/identity verification processes must be completed before the sale is finalized.
7. Specifications can go missing or be undocumented, for example if an owner modifies their own spaceship or the model is considered vintage. Having to check physical manuals or conduct research on every make/model can be very time consuming.
8. Currently we use Excel to track requests and customer information, which can lead to incorrect copies of customer data, duplicated requests causing extra or missing parts, and delays in corresponding with the customer due to inaccessibility.
9. To access the history, we have to access multiple Excel files depending on if the customer is looking for service or parts records. These are stored in separate file locations, so both must be checked to confirm each customer’s full history.
10. There is an inspection checklist and intake form to be completed for each ship that is tracked by the ship’s serial number.
11. Damage, missing parts, and suspicious modifications should be immediately reported to your supervisor. Do not complete any further work until the area is inspected by your supervisor, as this could lead to unsafe working conditions.
12. Discounts and/or price negotiations are authorized up to a certain percentage of the total sales price of the ship. For exact specifications, please speak with your management team.

AI generated Business Rules:

-Inventory & Sales Management

1. All spaceship sales data must be categorized and reported by model to assist in inventory and sales performance tracking.
2. Demand forecasting must be based on a comparison of monthly sales averages from the previous year to the current year, and reviewed quarterly.
3. Year-over-year customer purchase history and financing requests must be archived and accessible to support long-term business strategy decisions.

-Spaceship Intake & Resale Eligibility

1. Spaceships eligible for resale must be no more than 5 years old, must pass a complete mechanical inspection, and have no legal restrictions on ownership or resale.
2. Newly delivered spaceships must undergo a standardized intake inspection using a checklist and intake form, which is logged by serial number into the inventory system.
3. Any spaceship with damage, missing parts, or suspicious modifications must be reported immediately to a supervisor, and no work should proceed until further instruction.

-Pricing, Financing & Trade-in Policies

1. Spaceship pricing, financing, and trade-in offers must be based on model, age, condition, and current galactic market demand, and reviewed bi-annually for competitiveness.
2. Sales representatives are allowed to offer discounts only up to a predefined percentage of the spaceship’s price; discounts beyond that require management approval.

-Pre-Sale Compliance & Verification

1. Before finalizing any sale, the spaceship must pass a full mechanical inspection, complete documentation review, and buyer identity verification, including background checks.
2. All sale paperwork must be reviewed and signed off by a manager or designated sales auditor before the transaction is finalized.

-Data Management & Customer Records

1. All spaceship specifications and modifications must be digitally documented and stored in a centralized database, accessible to sales and service staff.
2. Customer sales, maintenance, and service records must be consolidated into a single system, eliminating the need to search across multiple Excel files or file locations.
3. All customer service and maintenance requests must be logged into a centralized system, not spreadsheets, to prevent duplication, delays, and errors in service fulfillment.

-Employee Procedures & Escalations

1. Low-level employees must refer pricing negotiations beyond the authorized discount level to management, and cannot approve exceptions independently.
2. All employees are required to follow the formal escalation process when reporting ship condition issues, and must not bypass supervisor inspection under any circumstances.

Edited Business Rules:

1. All spaceship sales data must include the spaceship model and serial number for inventory and performance tracking.
2. Demand forecasting will be based on the comparison of average monthly sales from year to year. This forecasting will also be reviewed quarterly.
3. Each year’s customer purchase history and financing requests must be archived but remain accessible to allow for long-term business strategy decision making.
4. To be eligible for resale, a spaceship must be no more than 5 years old, must pass a complete mechanical inspection, and have no legal restrictions on ownership.
5. Every newly delivered spaceship must go through a standard intake inspection, documented with a checklist and intake form, and then recorded in the inventory system by its serial number.
6. Any spaceship with damage, missing parts, or suspicious modifications must be reported immediately to a supervisor.
7. Spaceship pricing, financing, and trade-in offers will be based on each ship’s individual make, model, age, condition, and the current galactic market demand for these combined factors.
8. Sales representatives are authorized to offer discounts only up to a predefined percentage of each individual spaceship’s total price.
9. Before any sale is finalized, each spaceship must pass a full mechanical inspection, complete documentation review, and the buyer must pass identity verification, including a background check.
10. Before any sales is finalized, all sales paperwork must be review and signed off by a manager or designated auditor.
11. All spaceship specifications, including any modifications, must be digitally documented in the centralized database, which must be accessible to all Sales, Parts, and service staff, decreasing the time required to research each ship’s information.
12. Customer sales records, service maintenance records, and parts sales records must be consolidated into a centralized database, which is accessible to all Sales. Parts, and Service staff, eliminating the need to search across multiple Excel files or file locations.
13. All customer service and maintenance requests must be consolidated into a centralized database, which is accessible to all Sales. Parts, and Service staff, to prevent duplication, delays, and errors in service fulfillment.
14. Lower-level employees must escalate pricing negotiations above their authorized discount levels to management and cannot approve any exceptions.
15. All employees are required to report hazardous or unsafe working conditions to their supervisor immediately, unless it is unsafe to do so., then they are expected to safely remove themselves from the area, and report the incident as soon as possible.

Revised business rules with entities/attributes for database in mind:

1. A single model can have many spaceships. Each sale record is tied to exactly one spaceship.
2. Sales data is aggregated by month/year for forecasting.
3. A customer can make many purchases and financing requests.
4. A spaceship must meet specific conditions to qualify for resale.
5. Every delivered spaceship must have exactly one intake inspection and one inventory entry.
6. If damaged or suspicious, a single report is filed and escalated.
7. Each spaceship has an individual price based on its attributes and market conditions.
8. Each rep may apply discounts to multiple sales but cannot exceed limit.
9. Every spaceship must pass inspection & every buyer must pass verification before sale is finalized.
10. Each sale must be approved by exactly one manager/auditor.
11. Each spaceship’s specs & modifications are stored as a single record accessible by multiple roles.
12. Sales, parts, and maintenance data are merged into one database.
13. Multiple service requests are logged per customer in a single database.
14. Employees must escalate exceptions to a single manager.
15. A hazard event leads to exactly one report to a supervisor.

Entities, attributes, and Relationships

Entity: Customer; Attributes listed below:

CustomerID (Primary key)

FirstName

LastName

CompanyName

Phone #

Email

Address

DateOfBirth

IdentityVerified

VendorID (foreign key)

Entity: Spaceship; Attributes listed below:

SpaceshipID (Primary key)

Serial #

VendorID (foreign key)

InvoiceID (foreign key)

Make

Model

Model Year

Condition

Modifications

SalePrice

LastMaintenanceDate

Entity: Order; Attributes listed below:

OrderID (Primary key)

InvoiceID (foreign key)

CustomerID (foreign key)

SpaceshipID (foreign key)

OrderDateTime

Destination

OrderStatus

DiscountApplied

TotalAmount

PaymentID (foreign key)

PartsID (foreign key)

Entity: Payment; Attributes listed below:

PaymentID (Primary key)

InvoiceID (foreign key)

VendorID (foreign key)

OrderID (foreign key)

PaymentMethod

TotalAmount

PaymentDateTime

TransactionID

PaymentStatus

Currency

Entity: Employee; Attributes listed below:

EmployeeID (Primary key)

EmployeeFullName

FirstName

LastName

Role

HireDate

Department

SupervisorID (lookup column for EmployeeID that displays EmployeeFullName)

EmployeeStatus

Salary

Entity: Parts; Attributes listed below:

PartID (Primary key)

QuantityInStock

UnitPrice

PartName

PartDescription

Manufacturer

WarrantyExpirationDate

PartNumber

SpaceshipID (foreign key)

DateInstalled

PartStatus

Entity: Maintenance Request; Attributes listed below:

MaintenanceRequestID (Primary key)

SpaceshipID (foreign key)

PartID (foreign key)

RequestedBy (lookup column for EmployeeID that displays EmployeeFullName)

RequestDate

RequestType (Job Code / Inspection)

RequestDescription

PriorityLevel

RequestStatus

CompletionDate

Entity: Finance Request; Attributes listed below:

FinanceRequestID (Primary key)

RequestedBy (lookup column for EmployeeID that displays EmployeeFullName)

RequestDate

FinancedAmount

ApprovedBy (lookup column for EmployeeID that displays EmployeeFullName)

ApprovalDate

RequestType

RequestNotes

ApprovalStatus

Entity: Invoice; Attributes listed below:

InvoiceID (Primary key)

InvoiceDateTime

InvoiceAmount

TaxAmount

TotalAmount

Currency

VendorID (foreign key)

CustomerID (foreign key)

OrderID (foreign key)

PaymentID (foreign key)

InvoicePaymentStatus

InvoicePaymentDate

DueDate

DateRecieved

ApprovedBy (lookup column for EmployeeID that displays EmployeeFullName)

VendorInvoiceNumber

Entity: Vendor; Attributes listed below:

VendorID (primary key)

VendorName

VendorContact

PhoneNumber

Address

VendorType

Status

Rating

Relationships:

CUSTOMER one to many ORDER : places

CUSTOMER one to many INVOICE : receives

CUSTOMER many to one VENDOR : linked\_to

SPACESHIP one to many ORDER : sold\_in

SPACESHIP one to many PARTS : contains

SPACESHIP one to many MAINTENANCEREQUEST : has

SPACESHIP many to one VENDOR : supplied\_by

ORDER one to one INVOICE : billed\_on

ORDER one to one PAYMENT : paid\_by

ORDER one to many PARTS : may\_use

PAYMENT many to one VENDOR : sent\_to

PAYMENT many to one INVOICE : paid\_for

INVOICE many to one VENDOR : issued\_by

INVOICE many to one EMPLOYEE : approved\_by

MAINTENANCEREQUEST many to one EMPLOYEE : requested\_by

FINANCEREQUEST many to one EMPLOYEE : requested\_by

FINANCEREQUEST many to one EMPLOYEE : approved\_by

EMPLOYEE many to one EMPLOYEE : supervised\_by

ER Diagram example code:

erDiagram

CUSTOMER {

int CustomerID PK

string FirstName

string LastName

string CompanyName

string Phone

string Email

string Address

date DateOfBirth

bool IdentityVerified

int VendorID FK

}

SPACESHIP {

int SpaceshipID PK

string SerialNumber

int VendorID FK

int InvoiceID FK

string Make

string Model

int ModelYear

string Condition

string Modifications

decimal SalePrice

date LastMaintenanceDate

}

"ORDER" {

int OrderID PK

int InvoiceID FK

int CustomerID FK

int SpaceshipID FK

int PaymentID FK

int PartsID FK

datetime OrderDateTime

string Destination

string OrderStatus

decimal DiscountApplied

decimal TotalAmount

}

PAYMENT {

int PaymentID PK

int InvoiceID FK

int VendorID FK

int OrderID FK

string PaymentMethod

decimal TotalAmount

datetime PaymentDateTime

string TransactionID

string PaymentStatus

string Currency

}

EMPLOYEE {

int EmployeeID PK

string EmployeeFullName

string FirstName

string LastName

string Role

date HireDate

string Department

int SupervisorID FK

string EmployeeStatus

decimal Salary

}

PARTS {

int PartID PK

int SpaceshipID FK

int QuantityInStock

decimal UnitPrice

string PartName

string PartDescription

string Manufacturer

date WarrantyExpirationDate

string PartNumber

date DateInstalled

string PartStatus

}

MAINTENANCEREQUEST {

int MaintenanceRequestID PK

int SpaceshipID FK

int PartID FK

int RequestedBy FK

date RequestDate

string RequestType

string RequestDescription

string PriorityLevel

string RequestStatus

date CompletionDate

}

FINANCEREQUEST {

int FinanceRequestID PK

int RequestedBy FK

date RequestDate

decimal FinancedAmount

int ApprovedBy FK

date ApprovalDate

string RequestType

string RequestNotes

string ApprovalStatus

}

INVOICE {

int InvoiceID PK

datetime InvoiceDateTime

decimal InvoiceAmount

decimal TaxAmount

decimal TotalAmount

string Currency

int VendorID FK

int CustomerID FK

int OrderID FK

int PaymentID FK

string InvoicePaymentStatus

date InvoicePaymentDate

date DueDate

date DateReceived

int ApprovedBy FK

string VendorInvoiceNumber

}

VENDOR {

int VendorID PK

string VendorName

string VendorContact

string PhoneNumber

string Address

string VendorType

string Status

string Rating

}

CUSTOMER }o--|| VENDOR : "belongs\_to"

SPACESHIP }o--|| VENDOR : "supplied\_ by"

SPACESHIP }o--|| INVOICE : "linked\_to"

"ORDER" }o--|| CUSTOMER : "placed\_by"

"ORDER" }o--|| SPACESHIP : "includes"

"ORDER" }o--|| INVOICE : "billed\_on"

"ORDER" }o--|| PAYMENT : "paid\_by"

"ORDER" }o--|| PARTS : "uses"

PAYMENT }o--|| VENDOR : "sent\_to"

PAYMENT }o--|| INVOICE : "paid\_for"

PAYMENT }o--|| "ORDER" : "settles"

PARTS }o--|| SPACESHIP : "installed\_in"

MAINTENANCEREQUEST }o--|| SPACESHIP : "for"

MAINTENANCEREQUEST }o--|| PARTS : "involves"

MAINTENANCEREQUEST }o--|| EMPLOYEE : "requested\_by"

FINANCEREQUEST }o--|| EMPLOYEE : "requested\_by/approved\_by"

INVOICE }o--|| VENDOR : "issued\_by"

INVOICE }o--|| CUSTOMER : "issued\_to"

INVOICE }o--|| "ORDER" : "documents"

INVOICE }o--|| PAYMENT : "paid\_by"

INVOICE }o--|| EMPLOYEE : "approved\_by"

EMPLOYEE }o--|| EMPLOYEE : "supervised\_by"

Key Justification:

Entity: Vendor

Candidate keys: (VendorName,VendorContact,VendorAddress;VendorStatus); VendorID

Primary key: VendorID

Foreign keys: n/a

Super key: VendorID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: Customer

Candidate keys: CustomerID; (FirstName, LastName, DateOfBirth, Phone#, Email); (CompanyName, Address, Phone#, Email)

Primary key: CustomerID

Foreign keys: VendorID

Super key: CustomerID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: Payment

Candidate keys: PaymentID; TransactionID; (InvoiceID,OrderID,PaymentDateTime, PaymentStatus, TotalAmount)

Primary key: PaymentID

Foreign keys: InvoiceID; VendorID; OrderID; CustomerID

Super key: PaymentID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: Invoice

Candidate keys: InvoiceID; (InvoiceDateTime, InvoiceAmount, CustomerID, VendorID)

Primary key: InvoiceID

Foreign keys: VendorID; CustomerID; OrderID; PaymentID; ApprovedBy

Super key: InvoiceID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: FinanceRequest

Candidate keys: FinanceRequestID; (RequestDate,SpaceshipID, FinancedAmount,RequestedBy)

Primary key: FinanceRequestID

Foreign keys: SpaceshipID; RequestedBy; ApprovedBy

Super key: FinanceRequestID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: Parts

Candidate keys: PartsID; (PartName, Manufacturer, PartNumber, PartStatus)

Primary key: PartsID

Foreign keys: SpaceshipID

Super key: PartsID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: Spaceship

Candidate keys: SpaceshipID; (Make, Model, ModelYear, VendorID); (Make, Serial#)

Primary key: SpaceshipID

Foreign keys: VendorID

Super key: SpaceshipID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: MaintenanceRequest

Candidate keys: MaintenanceRequestID; (SpaceshipID, RequestedBy, ReqeuestedDate, RequestType, RequestStatus)

Primary key: MaintenanceRequestID

Foreign keys: SpaceshipID, PartsID, RequestedBy

Super key: MaintenanceRequestID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: Order

Candidate keys: OrderID; (CustomerID, InvoiceID, OrderDateTime, TotalAmount, OrderStatus)

Primary key: OrderID

Foreign keys: InvoiceID; CustomerID; SpaceshipID; PaymentID; PartsID

Super key: OrderID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.

Entity: Employee

Candidate keys: EmployeeID; (EmployeeFirstName,EmployeeLastName, EmployeeRole, EmployeeStatus, EmployeeHireDate, EmployeeSupervisorID)

Primary key: EmployeeID

Foreign keys: SupervisorID

Super key: EmployeeID

Primary key reasoning: We used a unique, integer identifier that is not an intelligent characteristic of the entity as the primary key to aid in auto-incrementing and easy identification.