Derivation of relations from the refined conceptual model:

Strong entity types:

RepairOrder(repairNo, originDate, completeDate, complaintNote**)**
Primary key: repairNo

Procedure(procedureNo, procedureDescription**)**
Primary key: procedureNo

LineItem(itemNo, itemName, description**)**
Primary key: itemNo
Alternate key: itemName

Invoice(invoiceNo, odometerStart, invoicePrintDate**)**
Primary key: invoiceNo

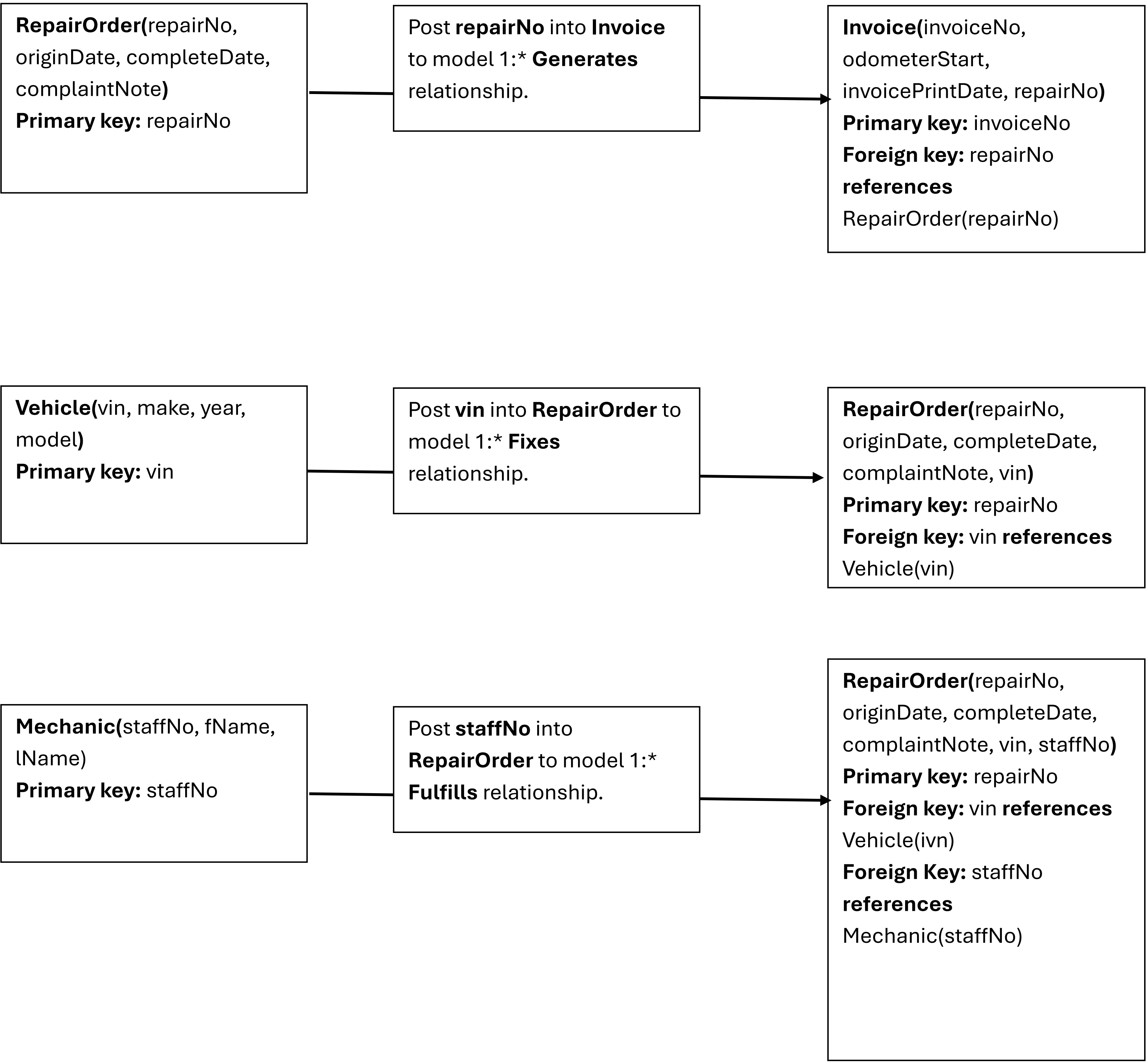
Vehicle(vin, make, year, model**) Changed from conceptual model, vin is unique and more easily accessed as primary key than vehicleNo.**
Primary key: vin

Mechanic(staffNo, fName, lName)
Primary key: staffNo

Weak Entity Types:

None

One-to-many (1:*) binary relationship types:



RepairOrder(repairNo, originDate, completeDate, complaintNote**)**
Primary key: repairNo

Post **repairNo** into **Invoice** to model 1:* **Generates** relationship.

Invoice(invoiceNo, odometerStart, invoicePrintDate, repairNo**)**
Primary key: invoiceNo
Foreign key: repairNo **references** RepairOrder(repairNo)

Vehicle(vin, make, year, model**)**
Primary key: vin

Post **vin** into **RepairOrder** to model 1:* **Fixes** relationship.

RepairOrder(repairNo, originDate, completeDate, complaintNote, vin**)**
Primary key: repairNo
Foreign key: vin **references** Vehicle(vin)

Mechanic(staffNo, fName, lName)
Primary key: staffNo

Post **staffNo** into **RepairOrder** to model 1:* **Fulfills** relationship.

RepairOrder(repairNo, originDate, completeDate, complaintNote, vin, staffNo**)**
Primary key: repairNo
Foreign key: vin **references** Vehicle(ivn)
Foreign Key: staffNo **references** Mechanic(staffNo)

<u>One-to-one (1:1) binary relationship types:</u>

a) **Mandatory participation on both sides:**

None

b) **Mandatory Participation on one side:**

None

c) **Optional participation on both sides:**

None

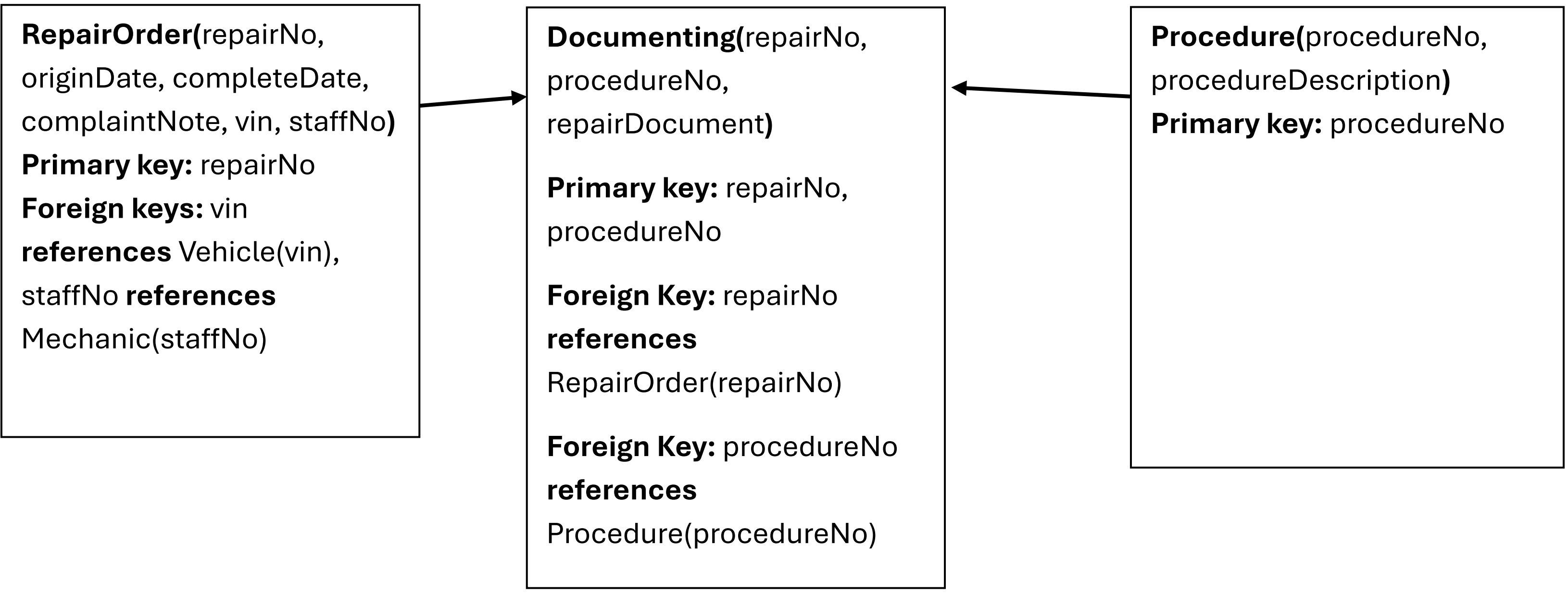
<u>One-to-one recursive relationships:</u>
None

<u>Superclass/Subclass Relationship Types:</u>
Mandatory, Disjoint:

LineItemLabor(itemNo, rate, hours, itemName, description**)**
Primary key: itemNo

LineItemPart(itemNo, price, itemName, description**)**
Primary key: itemNo

<u>Many-to-many (*:*) binary relationship types:</u>



RepairOrder(repairNo, originDate, completeDate, complaintNote, vin, staffNo**)**
Primary key: repairNo
Foreign keys: vin **references** Vehicle(vin), staffNo **references** Mechanic(staffNo)

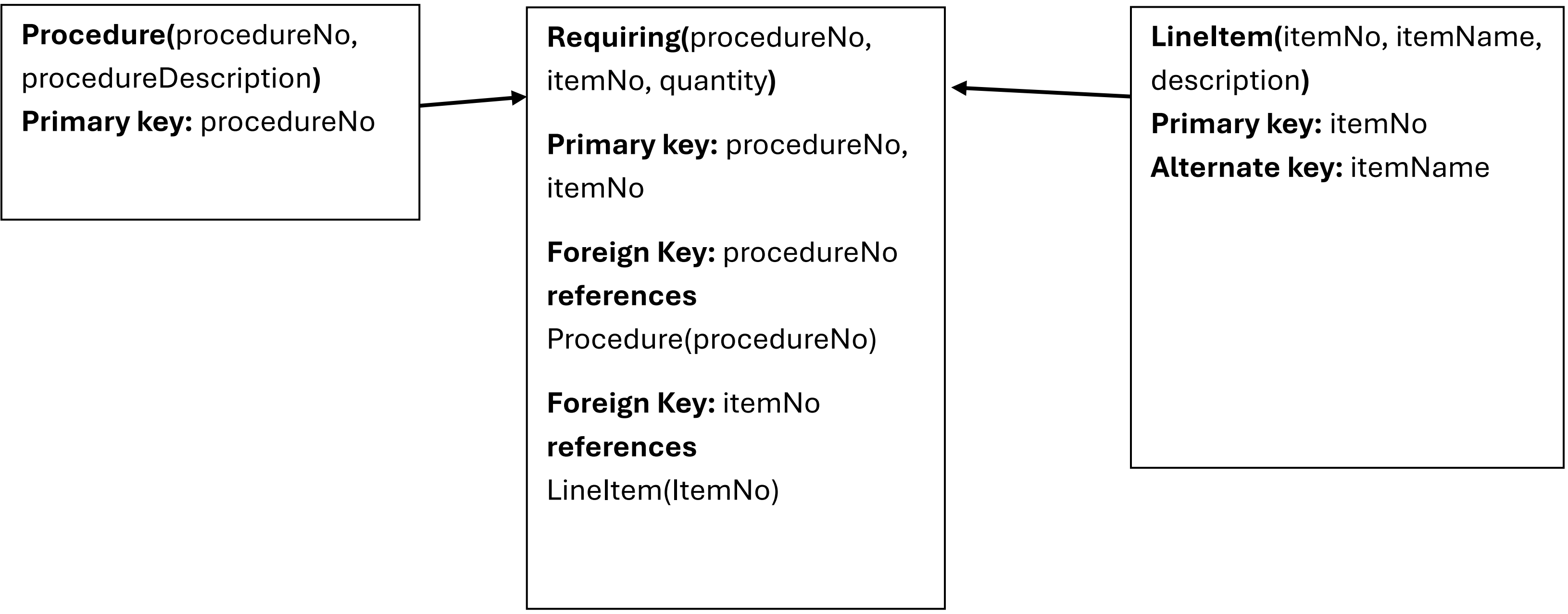
Documenting(repairNo, procedureNo, repairDocument**)**

Primary key: repairNo, procedureNo

Foreign Key: repairNo **references** RepairOrder(repairNo)

Foreign Key: procedureNo **references** Procedure(procedureNo)

Procedure(procedureNo, procedureDescription**)**
Primary key: procedureNo



Procedure(procedureNo, procedureDescription**)**
Primary key: procedureNo

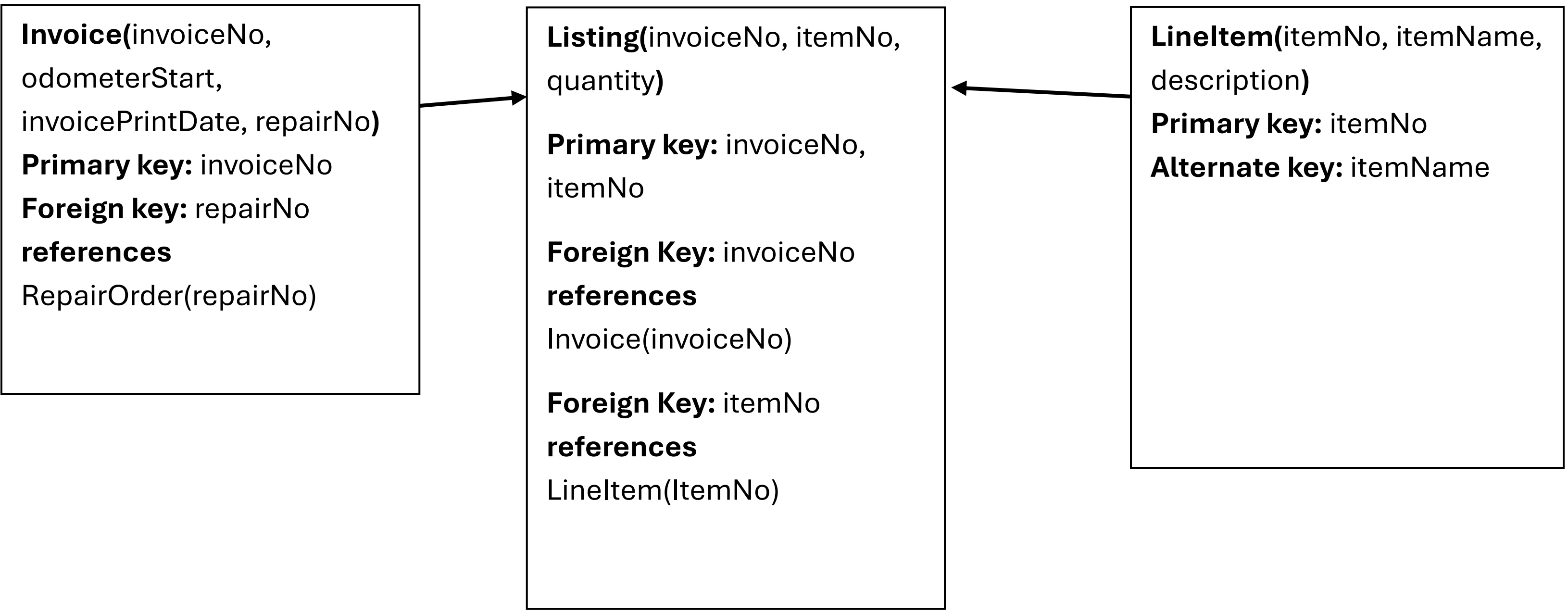
Requiring(procedureNo, itemNo, quantity**)**

Primary key: procedureNo, itemNo

Foreign Key: procedureNo **references** Procedure(procedureNo)

Foreign Key: itemNo **references** LineItem(ItemNo)

LineItem(itemNo, itemName, description**)**
Primary key: itemNo
Alternate key: itemName



Invoice(invoiceNo, odometerStart, invoicePrintDate, repairNo**)**
Primary key: invoiceNo
Foreign key: repairNo **references** RepairOrder(repairNo)

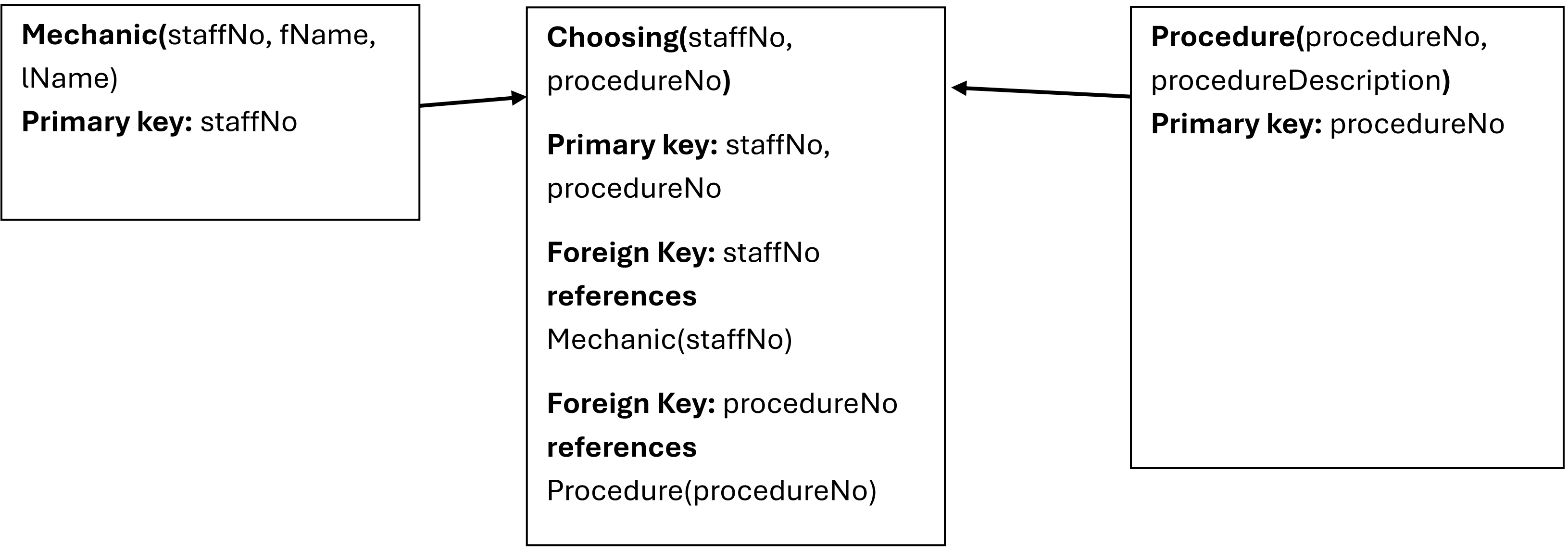
Listing(invoiceNo, itemNo, quantity**)**

Primary key: invoiceNo, itemNo

Foreign Key: invoiceNo **references** Invoice(invoiceNo)

Foreign Key: itemNo **references** LineItem(ItemNo)

LineItem(itemNo, itemName, description**)**
Primary key: itemNo
Alternate key: itemName



Mechanic(staffNo, fName, lName)
Primary key: staffNo

Choosing(staffNo, procedureNo**)**

Primary key: staffNo, procedureNo

Foreign Key: staffNo **references** Mechanic(staffNo)

Foreign Key: procedureNo **references** Procedure(procedureNo)

Procedure(procedureNo, procedureDescription**)**
Primary key: procedureNo

Complex relationship types:

None

Relations:

RepairOrder(repairNo, originDate, completeDate, complaintNote, vin, staffNo**)** **Primary key:** repairNo **Foreign key:** vin **references** Vehicle(vin) **Foreign Key:** staffNo **references** Mechanic(staffNo)	**Documentation(**repairNo, procedureNo, repairDocument**)** **Primary key:** repairNo, procedureNo **Foreign Key:** repairNo **references** RepairOrder(repairNo) **Foreign Key:** procedureNo **references** Procedure(procedureNo)
Procedure(procedureNo, procedureDescription**)** **Primary key:** procedureNo	**Requirement(**procedureNo, itemNo, quantity**)** **Primary key:** procedureNo, itemNo **Foreign Key:** procedureNo **references** Procedure(procedureNo) **Foreign Key:** itemNo **references** LineItem(ItemNo)
LineItem(itemNo, itemName, description**)** **Primary key:** itemNo **Alternate key:** itemName	**Listing(**invoiceNo, itemNo, quantity**)** **Primary key:** invoiceNo, itemNo **Foreign Key:** invoiceNo **references** Invoice(invoiceNo) **Foreign Key:** itemNo **references** LineItem(ItemNo)
Invoice(invoiceNo, odometerStart, invoicePrintDate, repairNo**)** **Primary key:** invoiceNo **Foreign key:** repairNo **references** RepairOrder(repairNo)	**Choice(**staffNo, procedureNo**)** **Primary key:** staffNo, procedureNo **Foreign Key:** staffNo **references** Mechanic(staffNo) **Foreign Key:** procedureNo **references** Procedure(procedureNo)
Vehicle(vin, make, year, model**)** **Primary key:** vin	
Mechanic(staffNo, fName, lName) **Primary key:** staffNo	

Multi-valued attributes:

None

Validate relations using normalization:

Repair Order:

repairNo -> originDate, completeDate, complaintNote, vin, staffNo

1NF: No repeating groups

2NF: Every non-primary-key attribute is fully functionally dependent on primary key **repairNo** (no partial functional dependency).

3NF: No non-primary-key attribute is transitively dependent on the primary key **repairNo**.

BCNF: Determinant is candidate key (primary key)

Procedure

procedureNo -> procedureDescription

1NF: No repeating groups

2NF: The non-primary-key attribute is functionally dependent on **procedureNo** (no partial functional dependency).

3NF: The non-primary-key attribute is transitively dependent on the primary key **procedureNo**.

BCNF: Determinant is candidate key (primary key)

LineItem

itemNo -> itemName, description

1NF: No repeating groups

2NF: Every non-primary-key attribute is fully functionally dependent on primary key **itemNo** (no partial functional dependency)

3NF: No non-primary-key attribute is transitively dependent on the primary key **itemNo**.

BCNF: Determinant is candidate key (primary key)

Invoice

invoiceNo -> odometerStart, invoicePrintDate, repairNo

1NF: No repeating groups

2NF: Every non-primary-key attribute is fully functionally dependent on primary key **invoiceNo** (no partial functional dependency). Primary key is not composite.

3NF: No non-primary-key attribute is transitively dependent on the primary key **invoiceNo**.

BCNF: Determinant is candidate key (primary key)

Vehicle

vin -> make, year, model

1NF: No repeating groups

2NF: Every non-primary-key attribute is fully functionally dependent on primary key **vin** (no partial functional dependency). Primary key is not composite.

3NF: No non-primary-key attribute is transitively dependent on the primary key **vin**.

BCNF: Determinant is candidate key (primary key).

Mechanic

staffNo -> fName, lName

1NF: No repeating groups

2NF: Every non-primary-key attribute is fully functionally dependent on primary key **staffNo** (no partial functional dependency). Primary key is not composite

3NF: No non-primary-key attribute is transitively dependent on the primary key **staffNo**.

BCNF: Determinant is candidate key (primary key)

Documenting

repairNo, procedureNo -> repairDocument

1NF: No repeating groups

2NF: Every non-primary-key attribute is fully functionally dependent on primary key **repairNo, procedureNo** (no partial functional dependency). **repairDocument** is specific to the combination of **repairNo** and **procedureNo.**

3NF: The non-primary-key attribute is not transitively dependent on the primary key **repairNo, procedureNo**.

BCNF: Determinant is candidate key (primary key)

Requiring

procedureNo, itemNo -> quantity

1NF: No repeating groups

2NF: Every non-primary-key attribute is fully functionally dependent on primary key **procedureNo, itemNo** (no partial functional dependency). **quantity** is specific to the combination of **procedureNo** and **itemNo.**

3NF: The non-primary-key attribute is not transitively dependent on the primary key **procedureNo, itemNo**.

BCNF: Determinant is candidate key (primary key)

Listing

invoiceNo, itemNo -> quantity

1NF: No repeating groups

2NF: Every non-primary-key attribute is fully functionally dependent on primary key **invoiceNo, itemNo** (no partial functional dependency). **quantity** is specific to the combination of **invoiceNo** and **itemNo.**

3NF: The non-primary-key attribute is not transitively dependent on the primary key **invoiceNo, itemNo**.

BCNF: Determinant is candidate key (primary key)

Choosing
staffNo, procedureNo
1NF: No repeating groups
2NF: There is nothing to depend on. No non-primary attributes. No partial functional dependency.
3NF: No non-primary-key attributes, no transitive dependency.
BCNF: No determinant, just primary key.

Validation of logical model against corresponding user transactions:

Repair Order:
A **Repair Order** is used to track a repair. Each repair order has a number that is generated by the system, an origination date, and a completion date. A repair order has any number of notes that are used to describe the complaints, as well as document the course of a repair. A repair order is for a **Vehicle**.

Mechanic:
After the car is diagnosed, the **Mechanic** will decide what **Procedures** are needed to repair it.

Procedure:
The available procedures are stored in the information system as procedure definitions. Some procedure definitions include: Replacing starter motor; Replacing accelerator cable; State inspection; Lube, oil, and oil filter; Replacing tail light bulb; Replacing head light bucket; Repairing and replacing Pitman arm; and Servicing and repairing transmission.

Invoice:
In some cases, a repair estimate is prepared. In other cases, a repair is done directly. In any case, an **Invoice** of **Line Items** required for the repair is prepared with a description, quantity, and price for each line item. The invoice is printed either as an **estimate** or as a **final bill** when the repair is complete and payable. There can be multiple invoices per repair order, including partial billing or pre-payment, as well as multiple estimates. The invoice includes the following information: Invoice number, which is listed on the document handed to the customer. Odometer mileage when the vehicle comes in and when it is finished. Date when the repair originated (from the repair order). Date when the repair is complete (from the repair order). Date when the invoice is printed. Date paid.

Line Item:
A **Procedure** may have many line items and a **Line Item** pertains to a specific procedure. Examples of line items include: Starter; Labor for starter replacement; Towing; Accelerator cable; Labor for replacing the accelerator cable; Bulb; Labor for lube, oil, and oil filter; Oil filter; Headlight adjusting screw; and State safety inspection.

<u>Definition of integrity constraints:</u>

Primary Key Constraints/ Entity Integrity:

RepairOrder
Primary Key Constraints/ Entity Integrity:
Primary key: repairNo must be unique and not null.
Foreign Key Constraints/Referential Integrity
Foreign key: vin **references** Vehicle(vin) ON UPDATE CASCADE ON DELETE CASCADE
Foreign Key: staffNo **references** Mechanic(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION
To Insert a staffNo into the **RepairOrder** relation, it must already exist in the **Mechanic** relation.

Procedure

Primary Key Constraints/ Entity Integrity:

Primary key: procedureNo must be unique and not null.

LineItem

Primary Key Constraints/ Entity Integrity:

Primary key: itemNo must be unique and not null.

Invoice

Primary Key Constraints/ Entity Integrity:

Primary key: invoiceNo must be unique and not null.

Foreign Key Constraints/Referential Integrity

Foreign key: repairNo **references** RepairOrder(repairNo) ON UPDATE CASCADE ON DELETE CASCADE

Vehicle

Primary Key Constraints/ Entity Integrity:

Primary key: vin must be unique and not null.

Mechanic

Primary Key Constraints/ Entity Integrity:

Primary key: staffNo must be unique and not null.

Documenting

Primary Key Constraints/ Entity Integrity:

Primary key: repairNo, procedureNo must be unique and not null.

Foreign Key Constraints/Referential Integrity

Foreign Key: repairNo **references** RepairOrder(repairNo) ON UPDATE CASCADE ON DELETE CASCADE

Foreign Key: procedureNo **references** Procedure(procedureNo) ON UPDATE CASCADE ON DELETE NO ACTION

Requiring

Primary Key Constraints/ Entity Integrity:

Primary key: procedureNo, itemNo must be unique and not null.

Foreign Key Constraints/Referential Integrity

Foreign Key: procedureNo **references** Procedure(procedureNo) ON UPDATE CASCADE ON DELETE CASCADE

Foreign Key: itemNo **references** LineItem(ItemNo) ON UPDATE CASCADE ON DELETE NO ACTION

Listing

Primary Key Constraints/ Entity Integrity:

Primary key: invoiceNo, itemNo must be unique and not null.

Foreign Key Constraints/Referential Integrity

Foreign Key: invoiceNo **references** Invoice(invoiceNo) ON UPDATE CASCADE ON DELETE CASCADE

Foreign Key: itemNo **references** LineItem(ItemNo) ON UPDATE CASCADE ON DELETE NO ACTION

Choosing

Primary Key Constraints/ Entity Integrity:

Primary key: staffNo, procedureNo must be unique and not null.

Foreign Key Constraints/Referential Integrity

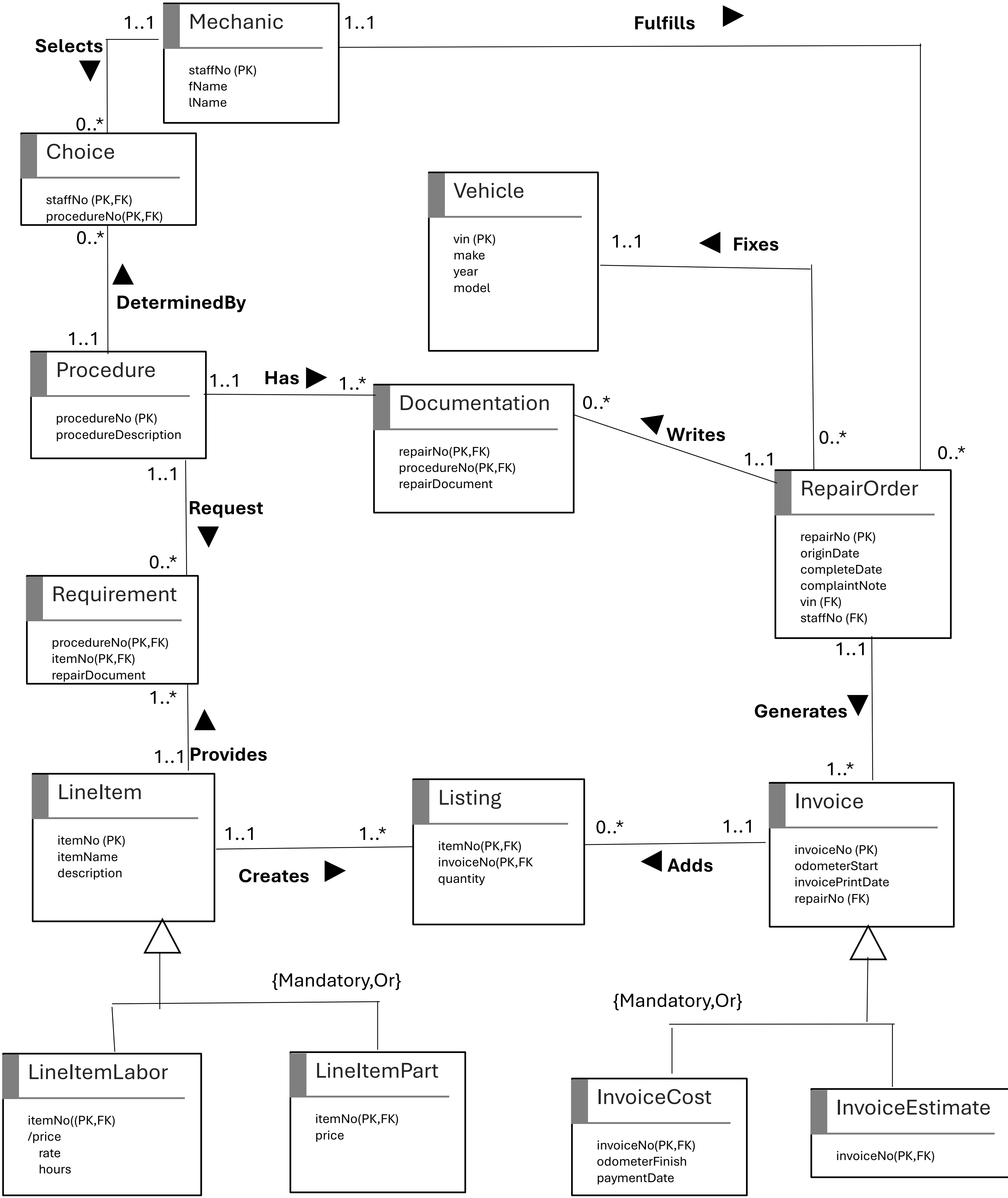
Foreign Key: staffNo **references** Mechanic(staffNo) ON UPDATE CASCADE ON DELETE CASCADE

Foreign Key: procedureNo **references** Procedure(procedureNo) ON UPDATE CASCADE ON DELETE NO ACTION

General Constraints:

General constraints could be added, but there are none specified in the instructions. For example, there could be constraints implemented limiting the number of Repair Orders a single mechanic could have active at one time.

UPDATED EER DIAGRAM:


Mechanic
staffNo (PK)
fName
lName
1..1
Selects
0..*
1..1
Fulfills
Choice
staffNo (PK,FK)
procedureNo(PK,FK)
0..*
DeterminedBy
1..1
Vehicle
vin (PK)
make
year
model
1..1
Fixes
0..*
Procedure
procedureNo (PK)
procedureDescription
1..1
Has
1..*
Documentation
repairNo(PK,FK)
procedureNo(PK,FK)
repairDocument
0..*
Writes
1..1
0..*
RepairOrder
repairNo (PK)
originDate
completeDate
complaintNote
vin (FK)
staffNo (FK)
1..1
Request
0..*
Requirement
procedureNo(PK,FK)
itemNo(PK,FK)
repairDocument
1..*
Provides
1..1
1..1
Generates
1..*
LineItem
itemNo (PK)
itemName
description
1..1
Creates
1..*
Listing
itemNo(PK,FK)
invoiceNo(PK,FK
quantity
0..*
Adds
1..1
Invoice
invoiceNo (PK)
odometerStart
invoicePrintDate
repairNo (FK)
{Mandatory,Or}
{Mandatory,Or}
LineItemLabor
itemNo((PK,FK)
/price
rate
hours
LineItemPart
itemNo(PK,FK)
price
InvoiceCost
invoiceNo(PK,FK)
odometerFinish
paymentDate
InvoiceEstimate
invoiceNo(PK,FK)