Part 1. - English to Schema

Example: **Student [__sID (integer)__, name (string), GPA (real)]**

Inventory

Quantity (not unique)

Price (not unique)

Name (not unique)

SKU (unique)

Inventory [SKU(string), name(string), price(real), quantity(integer)]

Aisles (not unique, can be on multiple, cannot have multiple on the same aisle though)

Product Aisles [SKU(string), Aisle(integer), name(string), price(real)]

car:

Make (not unique)(string)
Model (not unique)(string)
Year (not unique)(integer)
Color (not unique)(string)
VIN (unique) (string)

Salesperson:
Name (not unique) (string)
SSN (unique)(string)

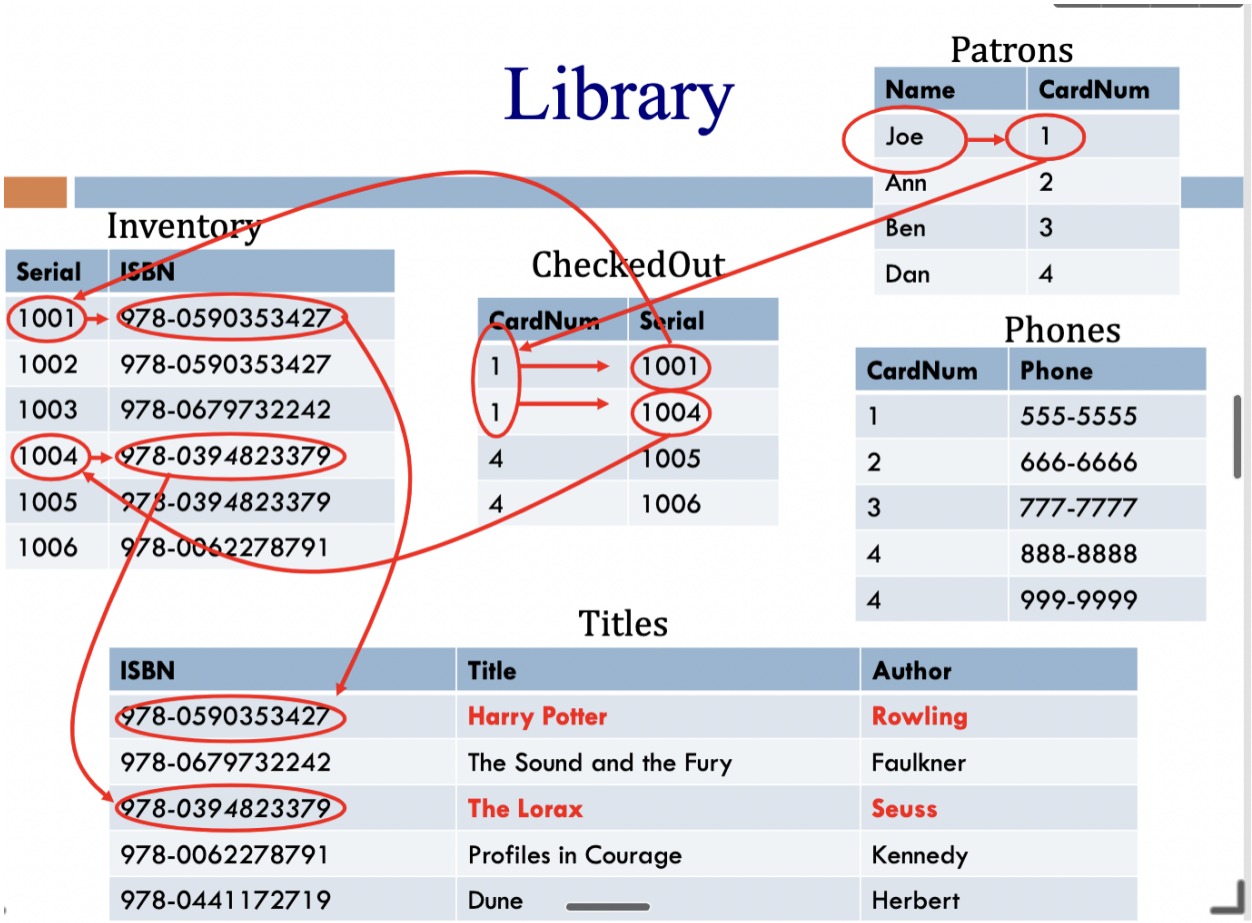
Dealership:
Inventory of cars (not unique)(integer)
Set of salespeople (more than 1 can be assigned to any given car, but a car does not necessarily have any

Inventory [VIN(string), make(string), model(string), year(integer), color(string)]

Employees [SSN(string), name(string)]

Car assignments [VIN(string), SSN(integer)]

Part 2 - SQL Table Declarations




```
CREATE TABLE Patrons (
  Name (string),
  CardNum (integer),
  PRIMARY KEY (CardNum),
)

CREATE TABLE CheckedOut (
  CardNum (integer),
  Serial (integer),
  PRIMARY KEY (Serial),
  FOREIGN KEY (CardNum) REFERENCES Patrons (CardNum)
  FOREIGN KEY (Serial) REFERENCES Inventory (Serial)
)

CREATE TABLE Phones (
  CardNum (integer),
  Phone (string),
  PRIMARY KEY (CardNum, Phone)
  FOREIGN KEY (CardNum) REFERENCES Patrons (CardNum)
)
```

```
CREATE TABLE Inventory (
  Serial (integer),
  ISBN (string),
  PRIMARY KEY (Serial),
  FOREIGN KEY (ISBN) REFERENCES Titles (ISBN)
)

CREATE TABLE Titles (
  ISBN (string),
  Title (string),
  Author (string),
  PRIMARY KEY (ISBN),
  UNIQUE (Title, Author)
)
```

Part 3 - Fill in Tables

Inventory				
VIN (string)	**Make (string)**	**Model (string)**	**Year (integer)**	**Color (string)**
4Y1SL65848Z411439	Toyota	Tacoma	2008	red
L87WEC5499P0J3Q22	Toyota	Tacoma	1999	Green
FG6925TQ88M011L4J	Tesla	Model 3	2018	White
902R5HV7844N1L7N2	Subaru	WRX	2016	Blue
FG899EZ205C3UNE61	Ford	F150	2004	Red

Employees	
SSN (string)	**Name (string)**
848-22-3209	Arnold
222-65-5114	Hannah

Employees	
405-96-4376	Steve

Salespeople:

Arnold, trying to sell all Toyotas
Hannah, trying to sell all red cars
Steve, selling the Tesla

Inventory [VIN(string), make(string), model(string), year(integer), color(string)]

Employees [SSN(integer), name(string), sales(integer)]

Car assignments [VIN(string), SSN(string)]

Car Assignments	
VIN (string)	**SSN (string)**
4Y1SL65848Z411439	848-22-3209
L87WEC5499P0J3Q22	848-22-3209
4Y1SL65848Z411439	222-65-5114
FG899EZ205C3UNE61	222-65-5114
FG6925TQ88M011L4J	405-96-4376

Part 4 - Keys and Superkeys

Attribute Sets	Superkey?	Proper Subsets	Key?
{A1}	no	{}	no
{A2}	no	{}	no
{A3}	no	{}	no
{A1, A2}	yes	{A1}, {A2}	yes
{A1, A3}	no	{A1}, {A3}	no
{A2, A3}	no	{A2}, {A3}	no
{A1, A2, A3}	yes	{A1}, {A2}, {A3}, {A1, A2}, {A2, A3}, {A1, A3}	no

Part 5 - Abstract Reasoning

- If {x} is a superkey, then any set containing x is also a superkey:
  True- because {x} is always a super key, any larger set combination will still be unique because it contains {x} which will make everything within unique
- If {x} is a key, then any set containing x is also a key.
  False- for {x} to be a key it must not contain a super key within it's subsets. For something to be a key it must also be a superkey and contain no superkey within it's subsets, so a set containing {x} may be a superkey but it cannot be a key.
- If {x} is a key, then {x} is also a superkey.
  True- for {x} to be a key, it must also be a super key, that is one of the requirements for something to be a key
- If {x, y, z} is a superkey, then one of {x}, {y}, or {z} must also be a superkey.
  False- {x, y, z} may be a key, which would mean that no smaller proper subset contains a super key

- If an entire schema consists of the set {x, y, z}, and if none of the proper subsets of {x, y, z} are keys, then {x, y, z} must be a key.
  True- a key cannot contain a superkey within it's subset, so for this schema to be legitimate, {x, y, z} needs to be a key