Erik Poole

**Part 1 – English to Schema**

- A grocery store needs to track an inventory of products for sale. It has zero or more of each type of product for sale, and needs to track the quantity and price for each product. A product has a name and a "stock keeping unit" (SKU) (hint: this is a real thing, google it). Remember that a valid table instance can't have duplicate rows - the store does not care about differentiating between individual items of the same product type, but it does want to be able to count them.

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

- Consider the grocery store database from the previous problem, but with a few differences: Now we don't care about tracking quantity, but we do want to track which aisle(s) the product is to be displayed on. Sometimes a product is displayed on more than one aisle in special display racks, but the product can not have multiple display cases per aisle. You may copy the relevant parts from your previous answer, but they will need modifications/additions.

Products [SKU (integer), name (string), price (real)]
Displays [SKU (integer), aisle (integer)]

- A car has a make, model, year, color, and VIN (vehicle identification number). A salesperson has a name and a social security number, and is responsible for trying to sell zero or more cars. A car dealership has an inventory of cars and a set of salespeople. It needs to keep track of which car(s) each salesperson is trying to sell. More than one salesperson can be assigned to any given car, but a car does not necessarily have any salespeople assigned to it.

Cars [VIN (integer), make (string), model (string), year (integer), color (string)]
Salespeople [Social Security Number (integer), name (string)]
Assigned Sales [Social Security Number (integer), VIN (integer)]

**Part 2 – SQL Table Declarations**

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

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

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

CREATE TABLE Patrons (
	CardNum int,
	Name string,
	PRIMARY KEY (CardNum)
)

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

**Part 3 – Fill in Tables**

Table Cars –

| VIN | MAKE | MODEL | YEAR | COLOR |
|---|---|---|---|---|
| JTHCK262185023070 | Toyota | Tacoma | 2008 | Red |
| 2G2FV22G012142399 | Toyota | Tacoma | 1999 | Green |
| 5XYKT3A66CG246053 | Tesla | Model 3 | 2018 | White |
| 1G2FS23T3ML208078 | Subaru | WRX | 2016 | Blue |
| 1N4AL2AP7BN493562 | Ford | F150 | 2004 | Red |

Table Salespeople –

| SOCIAL SECURITY NUMBER | NAME |
|---|---|
| 690-05-0427 | Arnold |
| 235-05-2043 | Hannah |
| 466-50-4759 | Steve |

Assigned Sales –

| SOCIAL SECURITY NUMBER | VIN |
|---|---|
| 690-05-0427 | JTHCK262185023070 |
| 690-05-0427 | 2G2FV22G012142399 |
| 235-05-2043 | JTHCK262185023070 |
| 235-05-2043 | 1N4AL2AP7BN493562 |
| 466-50-4759 | 5XYKT3A66CG246053 |

**Part 4 – Keys and Superkeys**

Consider the following instance:

| A1 | A2 | A3 |
|---|---|---|
| x | 4.0 | q |
| y | 4.0 | p |
| z | 3.1 | p |
| z | 4.0 | p |

Fill in the missing entries in the table below. Some cells have been given as an example.

| 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},<br>{A1, A2},<br>{A2, A3},<br>{A1, A3} | No |

**Part 5 – Abstract Reasoning**

1. If {x} is a superkey, then any set containing x is also a superkey.

True! If {x} on its own is a superkey then adding additional attributes won't change the uniqueness of {x}. Adding attributes might make a set into a superkey, but it can never invalidate one.

2. If {x} is a key, then any set containing x is also a key.

False! One of the requirements for being a key is being a superkey and we saw from the last question that any set that is a superkey shares that status with larger sets that include it. The other requirement of being a key is that no proper subset of our set is a superkey. Adding {x} to any set would make it a superkey, but if {x} is a key then the new set cannot be a key by definition.

3. If {x} is a key, then {x} is also a superkey.

True! Being a superkey is one of the requirements for being a key.

4. If {x, y, z} is a superkey, then one of {x}, {y}, or {z} must also be a superkey.

False! {x, y, z}'s row uniqueness may be a result of a combination of the smaller subsets. It's entirely possible that none of {x}, {y}, or {z} are superkeys.

5. 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 schema must have at least one key and if no proper subsets are keys than {x, y, z} must be a key for the schema to be valid.