Database Project Proposal

Sean Kotrola, Ian Burns, Paul Cochran, Ben Napier

# Description:

This database will be used by a police department. The department needs a more effective way to manage the hardware components inside each of its vehicles: a task that is currently being accomplished by an overcrowded excel document. Thus, the database will keep track of the department’s vehicles, mobile computers, mobile computer docks, arbitrator systems, cradlepoint systems, and keyboards.

Vehicles are all of the police cars that are used for patrolling. As described above, they have a lot of equipment contained within them that must be monitored to ensure that each vehicle is patrol-ready at all times. Since all of the other items that the database will keep track of are housed within the police cars, vehicles will serve as the center of the database.

Mobile computers are the laptops that are used within the police cars. All of these computers are held in place by mobile computer docks. While there are several different types of computer docks, each one only holds one laptop at a time. In addition, each vehicle contains only one mobile computer and one mobile computer dock at a time.

Cradlepoint is the system that brings WiFi to police cars, which the mobile computers connect to. They use SIM cards (similar to phones) to accomplish this. Each vehicle should contain only one cradlepoint system.

Arbitrator systems are the systems responsible for capturing video footage from police cars. They are a combination of hardware and software that work together to accomplish this goal. The software is installed on the mobile computer, and the hardware includes some combination of a front and rear camera. Each vehicle will only contain one arbitrator system.

There are two types of keyboards that are used. One is membrane and one is mechanical. Depending on the temperature, one type of keyboard will have sticky keys, which makes typing difficult. Each police car should only have one keyboard at a time.

The users for the database will be the police department’s IT, dispatch, and property departments.

The IT users will use the database to keep track of inventory and ensure that the vehicles are always ready to be actively patrolling. Additionally, they will be responsible for maintaining the database’s integrity, and solving any issues associated with the database. They are the only ones who should have admin access.

Dispatch will use the database to see which cars have what equipment/items within them. This will allow them to see which vehicles are ready to be used, or which ones are not patrol ready at the moment.

Property would use the database in order to keep track of all of the items for the police department. This would allow them to easily see what items are where and which items the department needs to order/replace/fix.

### The entity sets for the database are:

* Vehicle
* Mobile Computer
* Mobile Computer Dock
* Arbitrator System
* Cradlepoint
* Keyboard
* Computer Software Status
* Front Camera
* Rear Camera

### The operations to be performed to maintain the database are:

* Enter the date when Arbitrator System last uploaded video footage to the server.
* Enter the date a technician last worked on the vehicle systems.
* Update how many vehicles are in the fleet.
* Update the type of computers in the fleet.
* Update the type of keyboards in the fleet.

### Some queries that could be asked of the database are (need 20):

1. Which vehicles have used arbitrator systems to upload footage recently?
2. Which vehicles are on COM Port 4?
3. What color Cradlepoint cable is in a car?
4. What is the PC Name for a certain car?
5. Which vehicles have a mobile computer which has the newest arbitrator version?
6. What is the model of a mobile computer?
7. Does a specific keyboard have sticky keys?
8. What is the mobile computer dock serial number being used by a computer?
9. When was the last time a specific car was modified?
10. Which vehicle has gone the longest without being modified\inspected?
11. What vehicle has a given PC Name in?
12. What is the serial number of a given PC Name?
13. What is the Cradlepoint card number for the Cradlepoint in a vehicle?
14. How many mobile computers are currently being used by IT?
15. What is the key for a computer encrypted with BitLocker?
16. Which cars have short dock stands?
17. How many vehicles have front and rear-view camera arbitrator systems?
18. How many bad bois have been caught by a specific vehicle?
19. Are there any vehicles that have a keyboard and do not have a computer in them?
20. Which Arbitrator Systems have cameras with a bad status code?

# ER Diagram

# 

# Ben - due by Tuesday

Mobile comp & Dock

SEANYBOI

Computer software & cradlepoint

IAN

Keyboard & arbitrator system

PUAL

Vehicle, front, & rear camera

# List of the attributes of each entity and relationship in the diagram, the domain of the attribute and the candidate keys

# Relational Model

Legend: PrimaryKey, *ForeignKey*

Format:

**Entity(**attribute1, attribute2, ...**)**

* Attribute1: domain
* Attribute2: domain
* ...

**Entities**

NOTE: To avoid redundancy, unless otherwise stated, all attributes are assumed to be NOT NULL

**Vehicle**(unitNumber, make, model, lastModifyDate, badBoysCaught)

* unitNumber: Primary Key, String of digits
* make: String from a list of makes
* model: String from a list of models
* lastModifyDate: Datetime (Can be null)
* badBoysCaught: Integer >= 0

**MobileComputer**(serialNumber, model, PCName, gpsComPort, type, inUse, bitlockerKey,

serviceTag)

* serialNumber: Primary Key, String
* model: String which must match a given list of model names
* PCName: String
* gpsComPort: Integer
* type: String from a list of given types
* inUse: Boolean
* bitLockerKey: String
* serviceTag: Integer >=0

**Cradlepoint**(cardNumber, cableColor)

* cardNumber: Primary Key, String of digits
* cableColor: String, Color Name

**Keyboard**(type, serialNumber, model, isSticky)

* type: String - either ‘Mechanical’ or ‘Membrane’
* serialNumber: Primary Key, string of digits
* model: String from a list of model names
* isSticky:Boolean

**ArbitratorSystem**(ID*,* hasFrontCamera, hasRearCamera, lastVideoUploadDate)

* ID: Primary Key, integer >= 0
* hasFrontCamera:Boolean
* hasRearCamera: Boolean
* lastVideoUploadDate: Datetime

**ComputerSoftwareStatus**(*serialNumber*, WindowsVersion, GPSUpdateLastRun,

ArbitratorVersion)

* *serialNumber*: Foreign Key to MobileComputer
* WindowsVersion: String as a version number (Windows 10, etc.)
* GPSUpdateLastRun: Datetime
* ArbitratorVersion: String as version number

**Dock**(serialNumber, standType)

* serialNumber: Primary Key, string
* standType: string from a list of standTypes

**FrontCamera**(Status, CameraID)

* status: string (“Broken”, “Working”, etc.)
* CameraID: Primary Key, integer

**RearCamera**(Status, CameraID)

* status: string (“Broken”, “Working”, etc.)
* CameraID: Primary Key, integer

**Relationships**

Has()

* No Attributes

Uses()

* No Attributes

IsMountedOn()

* No Attributes

**Functionality of all the relationships in the diagram**

Has(From vehicle to X) - One to One

* One Vehicle can only have one of each of these elements and each element can only exist in one vehicle at a time

Has(From Mobile Computer to Mobile Computer Status) - One to One

* One Mobile computer can only have one status, and each status belongs to only one computer

IsMountedOn - One to One

* A Mobile Computer can only be mounted on one dock at a time, and each dock can only be used with one computer at a time

Uses(x2) - One to One

* An arbitrator system can only use one front and back camera at a time, and a front or back camera can only be used in one arbitrator system at a time

**Integrity or consistency constraints**

1. All dates must be less than or equal to the current date
2. No Negative numbers are allowed for any numeric field
3. The maximum length of any string field is 256 characters

**Logical Design**

**Vehicle**(unitNumber, make, model, lastModifyDate, badBoysCaught)

* Candidate Keys: unitNumber
* Primary Key:unitNumber
* Functional dependencies satisfied:
  + unitNumber -> make, model, lastModifyDate, badBoysCaught
* Normal form: 5nf

**VehicleComputer**(*unitNumber, serialNumber*)

* Candidate keys: unitNumber, serialNumber
* Primary Key: unitNumber
* Functional dependencies satisfied:
  + unitNumber -> serialNumber
  + serialNumber -> unitNumber
* Normal Form: 5nf

**MobileComputer**(serialNumber, model, PCName, gpsComPort, type, inUse, bitlockerKey,

serviceTag)

* Candidate Key: serialNumber
* Primary Key: serialNumber
* Functional dependencies satisfied:
  + serialNumber -> model, PCName, gpsComPort, type, inUse, bitlockerKey, serviceTag
* Normal Form: 5nf

**Cradlepoint**(cardNumber, cableColor)

* Candidate keys: cardNumber
* Primary key: cardNumber
* Functional dependencies satisfied:
  + cardNumber -> cableColor
* Normal form: 5nf

**VehicleCradlepoint**(*unitNumber, cardNumber*)

* Candidate keys: unitNumber, cardNumber
* Primary Key:unitNumber
* Functional dependencies satisfied:
  + unitNumber -> cardNumber
  + cardNumber -> unitNumber
* Normal Form: 5nf

**Keyboard**(type, serialNumber, model, isSticky)

* Candidate keys: serialNumber
* Primary Key: serialNumber
* Functional dependencies satisfied:
  + serialNumber -> type, model, isSticky
* Normal form: 5nf

**VehicleKeyboard**(*unitNumber, serialNumber*)

* Candidate Keys: unitNumber, serialNumber
* Primary Key:unitNumber
* Functional dependencies satisfied:
  + unitNumber -> serialNumber
  + serialNumber -> unitNumber
* Normal Form: 5nf

**ArbitratorSystem**(ID*,* hasFrontCamera, hasRearCamera, lastVideoUploadDate)

* Candidate keys: ID
* Primary key: ID
* Functional dependencies satisfied:
  + ID -> hasFrontCamera, hasRearCamera, lastVideoUploadDate
* Normal Form -> 5nf

**VehicleArbitratorSystem**(*unitNumber*, *ID*)

* Candidate Keys: unitNumber, ID
* Primary Key: unitNumber
* Functional dependencies satisfied:
  + unitNumber -> ID
  + ID-> unitNumber
* Normal Form: 5nf

**ComputerSoftwareStatus**(*serialNumber*, WindowsVersion, GPSUpdateLastRun,

ArbitratorVersion)

* Candidate Keys: serialNumber
* Primary key: serialNumber
* Functional dependencies satisfied:
  + serialNumber -> WindowsVersion, GPSUpdateLastRun, ArbitratorVersion
* Normal Form: 5nf

**Dock**(serialNumber, standType)

* Candidate Keys: serialNumber
* Primary Key: serialNumber
* Functional dependencies satisfied:
  + serialNumber -> standType
* Normal Form: 5nf

**MobileComputerDock**(*computer\_serialNumber*, *dock\_serialNumber*)

* Candidate Keys: computer\_serialNumber, dock\_serialNumber
* Primary Key: computer\_serialNumber
* Functional dependencies satisfied:
  + computer\_serialNumber -> dock\_serialNumber
  + dock\_serialNumber -> computer\_serialNumber
* Normal Form: 5nf

**FrontCamera**(Status, CameraID)

* Candidate keys: CameraID
* Primary Key: CameraID
* Functional dependencies satisfied:
  + CameraID -> Status
* Normal Form: 5nf

**RearCamera**(Status, CameraID)

* Candidate keys: CameraID
* Primary Key: CameraID
* Functional dependencies satisfied:
  + CameraID -> Status
* Normal Form: 5nf

**ArbitratorSystemFrontCamera**(*ID*, *CameraID*)

* Candidate keys: ID, CameraID
* Primary key: ID
* Functional dependencies satisfied:
  + ID -> CameraID
  + CameraID -> ID
* Normal form: 5nf

**ArbitratorSystemRearCamera**(*ID*, *CameraID*)

* Candidate keys: ID, CameraID
* Primary key: ID
* Functional dependencies satisfied:
  + ID -> CameraID
  + CameraID -> ID
* Normal form: 5nf

**Q U E R I E S:**

**1. Which vehicles have used arbitrator systems to upload footage recently?**

SELECT V.unitNumber, Arb.lastVideoUpdate FROM Vehicle V

JOIN VehicleArbitratorSystem VAS ON V.unitNumber = VAS.unitNumber

JOIN ArbitratorSystem Arb ON VAS.ID = Arb.ID

ORDER BY lastVideoUpdate DESC;

**2. Which vehicles are on COM Port 4?**

SELECT V.unitNumber

FROM Vehicle V

JOIN VehicleComputer VC ON V.unitNumber = VC.unitNumber

JOIN MobileComputer C ON C.serialNumber = VC.serialNumber

WHERE gpsComPort = <gpsComPortHere>;

**3. What color Cradlepoint cable is in a car?**

SELECT cableColor FROM VehicleCradlepoint

NATURAL JOIN CradlePoint

WHERE VehicleCradlepoint.unitNumber = <unitNumberHere>;

**4. What is the PC Name for a certain car?**

SELECT PCName

FROM MobileComputer MC

JOIN VehicleComputer VC ON MC.serialNumber = VC.serialNumber

JOIN Vehicle V ON VC.unitNumber = V.unitNumber

WHERE V.unitNumber = <unitNumberHere>;

**5. Which vehicles have a mobile computer which has the newest arbitrator version?**

SELECT V.unitNumber FROM Vehicle V

JOIN VehicleComputer VC ON V.unitNumber = VC.unitNumber

JOIN MobileComputer MC ON VC.serialNumber = MC.serialNumber

JOIN ComputerSoftwareStatus CSS ON MC.serialNumber = CSS.serialNumber

WHERE CSS.ArbitratorVersion LIKE "NEW";

**6. What is the model of a mobile computer?**

SELECT model FROM MobileComputer

WHERE serialNumber = "<serialNumberHere>";

**7. Does a specific keyboard have sticky keys?**

SELECT hasStickyKeys FROM Keyboard

WHERE serialNumber = "<serialNumberHere>";

**8. What is the mobile computer dock serial number being used by a computer?**

SELECT D.serialNumber

FROM Dock D

JOIN MobileComputerDock MCD ON D.serialNumber = MCD.dock\_serialNumber

JOIN MobileComputer MC ON MCD.computer\_serialNumber = MC.serialNumber

WHERE MC.PCName = "<PCNameHere>";

**9. When was the last time a specific car was modified?**

SELECT lastModifyDate FROM Vehicle WHERE unitNumber = <unitNumberHere>;

**10. Which vehicle has gone the longest without being modified\inspected?**

SELECT unitNumber, lastModifyDate FROM Vehicle

ORDER BY lastModifyDate ASC

LIMIT 1;

**11. What vehicle has a given PC Name in?**

select v.unitNumber from Vehicle v join VehicleComputer c on v.unitNumber = c.unitNumber join MobileComputer MC on c.serialNumber = MC.serialNumber

where PCName = "enter pc name here"

**12. What is the serial number of a given PC Name?**

Select serialNumber from MobileComputer where PCName = "enter pc name here"

**13. What is the Cradlepoint card number for the Cradlepoint in a vehicle?**

select VC.cardNumber from CradlePoint join VehicleCradlepoint VC on CradlePoint.cardNumber = VC.cardNumber join Vehicle V on VC.unitNumber = V.unitNumber

where V.unitNumber =

**14. How many mobile computers are currently being used by IT?**

select count(serialNumber) as "Number of Computers" from MobileComputer where PCName = "IT"

**15. What is the key for a computer encrypted with BitLocker?**

# Not sure what to filter this by

Select bitLockerKey from MobileComputer where PCName = "";

**16. Which cars have short dock stands?**

select V.\* from Dock join MobileComputerDock MCD on Dock.SerialNumber = MCD.dock\_serialNumber join MobileComputer MC on MCD.computer\_serialNumber = MC.serialNumber join VehicleComputer VC on MC.serialNumber = VC.serialNumber join Vehicle V on VC.unitNumber = V.unitNumber

where Dock.standType = "short"

**17. How many vehicles have front and rear-view camera arbitrator systems?**

select count(VAS.unitNumber) from ArbitratorSystem join VehicleArbitratorSystem VAS on ArbitratorSystem.ID = VAS.ID where hasFrontCamera = TRUE and hasRearCamera = TRUE

**18. How many bad bois have been caught by a specific vehicle?**

select badBoysCaught from Vehicle where unitNumber = something

**19. Are there any vehicles that have a keyboard and do not have a computer in them?**

select \* from Vehicle join VehicleKeyboard on Vehicle.unitNumber = VehicleKeyboard.unitNumber where Vehicle.unitNumber not in(select Vehicle.unitNumber from Vehicle join VehicleComputer on Vehicle.unitNumber = VehicleComputer.unitNumber)

**20. Which Arbitrator Systems have cameras with a bad status code?**

select \* from ArbitratorSystem join ArbitratorSystemFrontCamera ASFC on ArbitratorSystem.ID = ASFC.ID join ArbitratorSystemRearCamera ASRC on ArbitratorSystem.ID = ASRC.ID join FrontCamera FC on ASFC.CameraID = FC.cameraID join RearCamera RC on ASRC.CameraID = RC.cameraID

where FC.status = "Broken" or RC.status = "Broken";