**Technical Report on**

**Bike and Electric Auto Rental Database Design**

*Written Daniel Zinyemba*

**Functional Dependencies**

The first step of designing any relational database is performing a functional dependency analysis. The insights made during this analysis were instrumental in the design of the first iteration of the LDM.

**Table, Excel

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

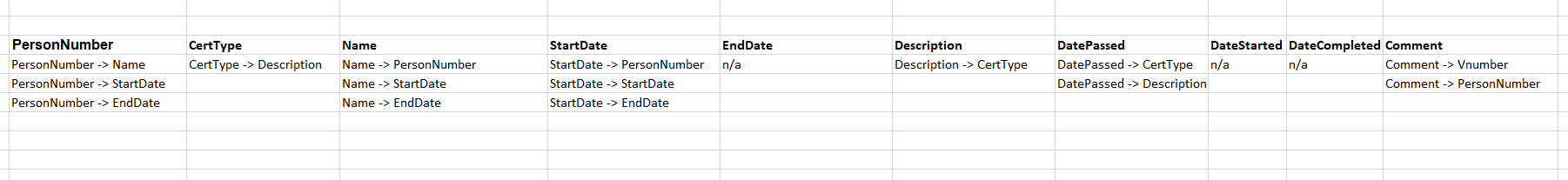
**Graphical user interface, application

Description automatically generated**

**Graphical user interface

Description automatically generated with medium confidence**

As the project progressed, more data and constraints were introduced by further memos. This forced me to update my functional dependency analysis in order to create an accurate impression of how this new data would fit. The data pictured in the data below had properties unique and unseen in the previous dataset. EndDate was null for all except one value in the Personnel universal relation. DateStarted and DateCompleted had different values for every entry listed. Because of this, despite the fact that they technically had functional dependencies, I discounted them because I could not use them to make relational inferences.

****

**Relational Database Design**

The Renters entity represents people who have rented any vehicle. They each have a unique identifier, a RenterID, as well as other personal information stored as attributes.

Vehicles, unsurprisingly, are by far the most complex entities in the BEAR database. They are divided into two subtypes, Bikes and Electric Vehicles, because there are many attributes that define those subgroups that don’t apply at all to their opposite.

Site is an entity that represents the site that each vehicle is located in.

The Rents entity stores information concerning the sale of individual licenses. In order to specifically identify a rent, we need to know the identity of the renter, the vehicle rented, and the time the rent occurred. This entity also tracks the number of times specific vehicles have been rented, as well as when they return.

Status and BikeType are represented solely by single character abbreviations in the sample data. They are organized in separate entities mostly so that they can have separate outputs.

Personnel are entities uniquely identified by their PersonNumber. They are also further stratified into Maintenance and Management personnel. It is very important to distinguish these groups in the database, because different staff members have different security clearances based on their responsibilities. A proper RDD allows referential integrity constraints to handle it.

Personnel can have Certifications from passing certification tests. These certifications have their own entity describing them.

Because it is possible for one member of staff Personnel to have multiple Certifications at once, the relation model needs a Capabilities relation, with a composite key made up of the PersonNumber and the CertType. With this entity, we can now determine who has what certification, and when they got it.

The Tasks entity stores the data for the jobs maintenance personnel perform to maintain the vehicles. Similar to Rents, it requires a triple key, including the id of the maintenance staff working, the vehicle number for the vehicle they are working on, and the date the task was started on.

Logical Data Model

A picture containing text, screenshot, indoor, several

Description automatically generated

Physical Data Model

Diagram, timeline

Description automatically generated

**DML INSERT STATEMENTS**

These insert statements will, if executed in SQLServer, load a generated database with the test data I was sent through the memos.

INSERT INTO Site VALUES

('AS','Airport Site','44 Airport Dr.'),

('DS','Downtown Site','423 Main St.'),

('SS','Suburban Site','22 Center Circle');

INSERT INTO Status VALUES

('R','Rented'),

('A','Available'),

('M','Maintenance');

INSERT INTO Personnel VALUES

('P0000', 'Bob Bureaucrat',NULL, NULL),

('P0101','Mary Manager' ,NULL, NULL),

('P0102', 'Fred Foreman', NULL, NULL),

('P0103','Sam Supervisor', NULL, NULL),

('P2108','Joe Brown','2016-11-03', NULL),

('P2109','Jane Smith','2016-11-03', NULL),

('P2202','Frank Martin','2016-11-05', NULL),

('P2213','Anne Dough','2016-11-10','2016-11-20'),

('P2303','Mike Green','2016-11-10', NULL),

('P2309','Alice Green','2017-01-10', NULL),

('P2400','Joe Brown','2017-01-10', NULL);

INSERT INTO Renters VALUES

('1','Anne Green','1995-01-02','IN24340020','233 Main St.'),

('2','John Dough','1990-11-03','IN24388033','123 1st St.'),

('3','Bill Smith','2000-10-04','IN67800000','444 W State.'),

('4','Gail White','1997-06-06','KY44230100','PO Box 2388.'),

('8','Evan Black','1995-04-12','IN24500001','42 Blvd Pl'),

('9','Bill Brown','1997-06-06','TX99923411','888 Tudor Dr');

INSERT INTO Capability VALUES

('P2108','B','11/06/2016'),

('P2108','E','11/10/2016'),

('P2108','K','11/04/2016'),

('P2109','B','11/06/2016'),

('P2202','B','11/06/2016'),

('P2303','B','11/25/2016'),

('P2303','E','11/20/2016');

INSERT INTO Bikes VALUES

('B0001','R','26',8.00,56.00),

('B0010', 'R','26',8.00,56.00),

('B0011','R','26',8.00,56.00),

('B0003','R','24',6.00,43.00),

('T0002','T','26',10.00,70.00);

INSERT INTO ElectricVehicles VALUES

('E0012','EV10000234','2505',9.00,0.40),

('E0014','EV10000235','3505',9.00,0.40),

('E0444','EV10000500','1200',10.00,0.40),

('E0523','EV10000600','500',11.00,0.50),

('E0524','EV10000601','10',11.00,0.50);

INSERT INTO Vehicles VALUES

('B0001','R','SS','B','Aug 2020'),

('B0010','R','DS','B', NULL),

('B0011','A','DS','B','Oct 2020'),

('B0003','A','SS','B','Aug 2020'),

('T0002','A','SS','B','Aug 2020'),

('E0012','R','AS','E','Sep 2020'),

('E0014','R','AS','E','Sep 2020'),

('E0444','A','SS','E','Nov 2020'),

('E0523','A','SS','E','Dec 2020'),

('E0524','M','SS','E','Dec 2020');

INSERT INTO Maintenance VALUES

('P2108'),

('P2109'),

('P2202'),

('P2213'),

('P2303'),

('P2309'),

('P2400');

INSERT INTO Tasks VALUES

('P2108','E0012','2017-10-10','2017-01-10','Broken Wiper'),

('P2109','B0003','2017-01-11','2017-01-11','Flat Tire'),

('P2202','B0001','2017-01-10','2017-11-10','Broken Chain'),

('P2213','B0003','2017-01-11','2017-11-11','Flat Tire'),

('P2303','B0010','2017-01-20','2017-01-20','New Wheel'),

('P2400','E0524','2017-12-20','2017-12-21','Initial Checks');

INSERT INTO Management VALUES

('P0000', NULL, NULL),

('P0101', NULL, 'DS'),

('P0102', NULL, 'AS'),

('P0103', NULL, 'SS');

INSERT INTO Rents VALUES

('1','B0001','2020-12-10 10:30','2020-12-10 12:30',3),

('1','B0011','2020-12-16 09:30','2020-12-16 12:30',2),

('3','B0001','2020-12-11 10:00','2020-12-11 16:00',3),

('4','B0001','2020-12-16 10:30','2020-12-16 18:30',3),

('8','B0003','2020-12-16 09:30','2020-12-16 12:30',1),

('8','T0002','2020-12-19 10:30','2020-12-19 12:30',4),

('9','B0011','2020-12-10 10:30','2020-12-10 12:30',2),

('1','B0011','2020-12-16 13:30','2020-12-16 16:30',3),

('1','E0012','2020-11-26 09:30','2020-11-26 18:30',3),

('2','E0012','2020-11-27 10:30','2020-11-27 18:30',3),

('2','E0014','2020-11-26 10:30','2020-11-26 14:30',3),

('4','E0444','2020-11-29 09:30','2020-11-30 09:30',3),

('8','E0012','2020-10-14 09:30','2020-10-15 13:30',3),

('8','E0523','2020-12-26 09:30','2021-01-02 18:30',0),

('9','E0524','2020-12-29 06:00','2020-12-31 13:00',0),

('2','E0012','2020-12-27 10:30','2016-12-27 18:30',4);

(NULL,'B0010',NULL,NULL,0);

INSERT INTO BikeType VALUES

('R','Road'),

('T','Tandem');

INSERT INTO Certifications VALUES

('B','Bicycle'),

('E','Electric'),

('K','Kiosk');

**QUERIES**

In order to test the robustness of this database design, I conducted a series of test queries that should encompass the breadth of information that the database should be able to provide. Above each query, there is a bold, numbered question. The query immediately below this question should answer this question, with said answer displayed immediately below the query in plaintext. The syntax and formatting of these queries were preserved for the sake of reproducible testing.

**1) What Electric Vehicle has the most mileage.**

SELECT MAX(Miles) FROM ElectricVehicles;

-----------

3505

(1 row affected)

**2) For each Bicycle, list all of the rentals in chronological order.**

SELECT Rents.VNumber, StartTime, EndTime FROM Bikes, Rents

WHERE Bikes.VNumber = Rents.VNumber

ORDER BY Rents.StartTime ASC;

VNumber StartTime EndTime

------- ----------------------- -----------------------

B0001 2020-12-10 10:30:00.000 2020-12-10 12:30:00.000

B0011 2020-12-10 10:30:00.000 2020-12-10 12:30:00.000

B0001 2020-12-11 10:00:00.000 2020-12-11 16:00:00.000

B0011 2020-12-16 09:30:00.000 2020-12-16 12:30:00.000

B0003 2020-12-16 09:30:00.000 2020-12-16 12:30:00.000

B0001 2020-12-16 10:30:00.000 2020-12-16 18:30:00.000

B0011 2020-12-16 13:30:00.000 2020-12-16 16:30:00.000

T0002 2020-12-19 10:30:00.000 2020-12-19 12:30:00.000

(8 rows affected)

**3) What is the location of the Downtown Site?**

SELECT Location FROM Site

WHERE SiteID = 'DS';

Location

--------------------

423 Main St.

(1 row affected)

**4) List the top 2 bicycles based on the number of hours rented.**

SELECT TOP 2 Rents.VNumber FROM Vehicles, Rents

WHERE VehicleType = 'B'

AND Vehicles.VNumber = Rents.VNumber

ORDER BY Rents.RentNumber ;

VNumber

-------

B0003

B0011

(2 rows affected)

**5) List the electric automobile rentals based on the number of hours rented.**

SELECT Rents.VNumber, Rents.RentNumber FROM ElectricVehicles, Rents

WHERE ElectricVehicles.VNumber = Rents.VNumber

ORDER BY Rents.RentNumber DESC;

VNumber RentNumber

------- -----------

E0012 4

E0014 3

E0444 3

E0012 3

E0012 3

E0012 3

E0523 0

E0524 0

(8 rows affected)

**6) What vehicles are currently available for rent?**

SELECT VNumber FROM Vehicles

WHERE Status = 'A';

VNumber

-------

B0003

B0011

E0444

E0523

T0002

(5 rows affected)

**7) List the renters of bicycles and the types of bicycles they rented.**

SELECT Renters.Name, Renters.RenterID FROM Renters, Rents, Vehicles

WHERE Vehicles.VehicleType = 'B'

AND Vehicles.VNumber = Rents.VNumber

AND Rents.RenterID = Renters.RenterID;

Name RenterID

------------------ -----------

Anne Green 1

Anne Green 1

Anne Green 1

Bill Smith 3

Gail White 4

Evan Black 8

Evan Black 8

Bill Brown 9

(8 rows affected)

**8) Who (Name and ID) manages the Downtown Site?**

SELECT Personnel.Name, Personnel.PersonNumber FROM Management, Personnel

WHERE Personnel.PersonNumber = Management.PersonNumber

AND Management.SiteID = 'DS';

Name PersonNumber

-------------------- ------------

Mary Manager P0101

(1 row affected)

**9) Who (Name and ID) has access to Suburban Site vehicle data?**

SELECT Personnel.Name, Personnel.PersonNumber FROM Management, Personnel

WHERE Personnel.PersonNumber = Management.PersonNumber

AND Management.SiteID = 'SS';

Name PersonNumber

-------------------- ------------

Sam Supervisor P0103

(1 row affected)

**10) Who How many times has each vehicle actually been rented?**

SELECT Vehicles.VNumber, Rents.RentNumber FROM Vehicles, Rents

WHERE Vehicles.VNumber = Rents.VNumber;

VNumber RentNumber

------- -----------

B0001 3

B0011 2

B0011 3

E0012 3

E0012 3

E0012 4

E0014 3

B0001 3

B0001 3

E0444 3

B0003 1

E0012 3

E0523 0

T0002 4

B0011 2

E0524 0

(16 rows affected)

**11) List the vehicles maintained by P2108.**

SELECT Vehicles.VNumber FROM Vehicles, Tasks, Maintenance

WHERE Maintenance.PersonNumber = 'P2108'

AND Maintenance.PersonNumber = Tasks.PersonNumber

AND Tasks.VNumber = Vehicles.VNumber;

VNumber

-------

E0012

(1 row affected)

**12) List the vehicle, type, and date rented for all vehicles rented by Anne Green.**

SELECT Vehicles.VNumber, Vehicles.VehicleType, Rents.StartTime FROM Vehicles, Rents, Renters

WHERE Renters.Name = 'Anne Green'

AND Renters.RenterID = Rents.RenterID

AND Rents.VNumber = Vehicles.VNumber;

VNumber VehicleType StartTime

------- ----------- -----------------------

B0001 B 2020-12-10 10:30:00.000

B0011 B 2020-12-16 09:30:00.000

B0011 B 2020-12-16 13:30:00.000

E0012 E 2020-11-26 09:30:00.000

(4 rows affected)

**Appendix**

Previous Iterations:

Memo 3

Diagram

Description automatically generatedA picture containing text, screenshot, sign, white

Description automatically generated

Memo 2

Diagram

Description automatically generated

Memo 1

Diagram

Description automatically generated

Generate Database Script:

ndDate datetime NULL ,

PRIMARY KEY CLUSTERED (PersonNumber ASC)

)

go

CREATE TABLE Maintenance

(

PersonNumber char(5) NOT NULL ,

PRIMARY KEY CLUSTERED (PersonNumber ASC),

FOREIGN KEY (PersonNumber) REFERENCES Personnel(PersonNumber)

)

go

CREATE TABLE Site

(

SiteID char(2) NOT NULL ,

Name varchar(20) NULL ,

Location varchar(20) NULL ,

PRIMARY KEY CLUSTERED (SiteID ASC)

)

go

CREATE TABLE Status

(

Status char(1) NOT NULL ,

Description varchar(20) NULL ,

PRIMARY KEY CLUSTERED (Status ASC)

)

go

CREATE TABLE Vehicles

(

VNumber char(5) NOT NULL ,

SiteID char(2) NULL ,

VehicleType char(1) NULL ,

PurchaseDate char(8) NULL ,

Status char(1) NULL ,

PRIMARY KEY CLUSTERED (VNumber ASC),

FOREIGN KEY (SiteID) REFERENCES Site(SiteID),

FOREIGN KEY (Status) REFERENCES Status(Status)

)

go

CREATE TABLE Tasks

(

PersonNumber char(5) NOT NULL ,

VNumber char(5) NOT NULL ,

DateStarted date NOT NULL ,

DateCompleted date NULL ,

Comment varchar(20) NULL ,

PRIMARY KEY CLUSTERED (PersonNumber ASC,VNumber ASC,DateStarted ASC),

FOREIGN KEY (PersonNumber) REFERENCES Maintenance(PersonNumber),

FOREIGN KEY (VNumber) REFERENCES Vehicles(VNumber)

)

go

CREATE TABLE Renters

(

RenterID integer NOT NULL ,

Name char(18) NULL ,

DOB date NULL ,

License char(10) NULL ,

Address varchar(12) NULL ,

PRIMARY KEY CLUSTERED (RenterID ASC)

)

go

CREATE TABLE Rents

(

RenterID integer NOT NULL ,

VNumber char(5) NOT NULL ,

StartTime datetime NOT NULL ,

EndTime datetime NULL ,

RentNumber integer NULL ,

PRIMARY KEY CLUSTERED (RenterID ASC,VNumber ASC,StartTime ASC),

FOREIGN KEY (RenterID) REFERENCES Renters(RenterID),

FOREIGN KEY (VNumber) REFERENCES Vehicles(VNumber)

)

go

CREATE TABLE Management

(

PersonNumber char(5) NOT NULL ,

LastAccessDate datetime NULL ,

SiteID char(2) NULL ,

PRIMARY KEY CLUSTERED (PersonNumber ASC),

FOREIGN KEY (PersonNumber) REFERENCES Personnel(PersonNumber),

FOREIGN KEY (SiteID) REFERENCES Site(SiteID)

)

go

CREATE TABLE ElectricVehicles

(

VNumber char(5) NOT NULL ,

VIN char(10) NULL ,

Miles integer NULL ,

BaseRate float NULL ,

MinuteRate float NULL ,

PRIMARY KEY CLUSTERED (VNumber ASC),

FOREIGN KEY (VNumber) REFERENCES Vehicles(VNumber)

)

go

CREATE TABLE Certification

(

CertType char(8) NOT NULL ,

Description char(18) NULL ,

PRIMARY KEY CLUSTERED (CertType ASC)

)

go

CREATE TABLE Capability

(

PersonNumber char(5) NOT NULL ,

CertType char(8) NOT NULL ,

DatePassed datetime NULL ,

PRIMARY KEY CLUSTERED (PersonNumber ASC,CertType ASC),

FOREIGN KEY (CertType) REFERENCES Certification(CertType),

FOREIGN KEY (PersonNumber) REFERENCES Personnel(PersonNumber)

)

go

CREATE TABLE BikeType

(

BikeType char(1) NOT NULL ,

Description varchar(20) NULL ,

PRIMARY KEY CLUSTERED (BikeType ASC)

)

go

CREATE TABLE Bikes

(

VNumber char(5) NOT NULL ,

BikeType char(1) NULL ,

Size integer NULL ,

HourRate float NULL ,

MaxCharge float NULL ,

PRIMARY KEY CLUSTERED (VNumber ASC),

FOREIGN KEY (VNumber) REFERENCES Vehicles(VNumber),

FOREIGN KEY (BikeType) REFERENCES BikeType(BikeType)

)

go