DBDS **Database Design Specialists, Inc.**

"We are always 5NF."

Technical Project Report on

Bike and Electric Auto Rental (BEAR)

Name: Keyur Kirti Mehta
NN: 20

Contents

1) **Problem Statement** 3
2) **Logical Data Model (LDM)** 4
3) **Functional Dependency Analysis** 5
   1. **FDs of Universal Relation** 5
   2. **FDs from Memo # 1** 7
   3. **FDs of Personnel Data** 7
   4. **FDs of Certification Data** 7
   5. **FDs of Maintenance Data** 8
4) **Construction of relations** 9
5) **Physical Data Model (PDM)** 11
6) **List of Tables with Test Data** 12
7) **Query Results** 16
   1. **Queries from Memo # 3** 16
   2. **Queries from Memo # 4** 19
8) **Memo # 7 (Last minute Data effect on model)** 23
   1. **New Data from Bob** 23
   2. **Information from Sam** 23
   3. **Data items from Bob that look Different** 23
   4. Number of samples field 23
   5. Overlapping Dates 24

Appendix 25
   1. ***Grid mode analysis of the universal Relation*** 25
   2. ***Previous Logical Data Model*** 28
      a. ***Version 2*** 28
      b. ***Version 1*** 29
      c. ***Steps to revise LDM*** 29
   3. ***Prototype of Relational Database Design (RDD) generated by ErWin*** 30
   4. ***Test data loading queries*** 33

1) Problem Statement

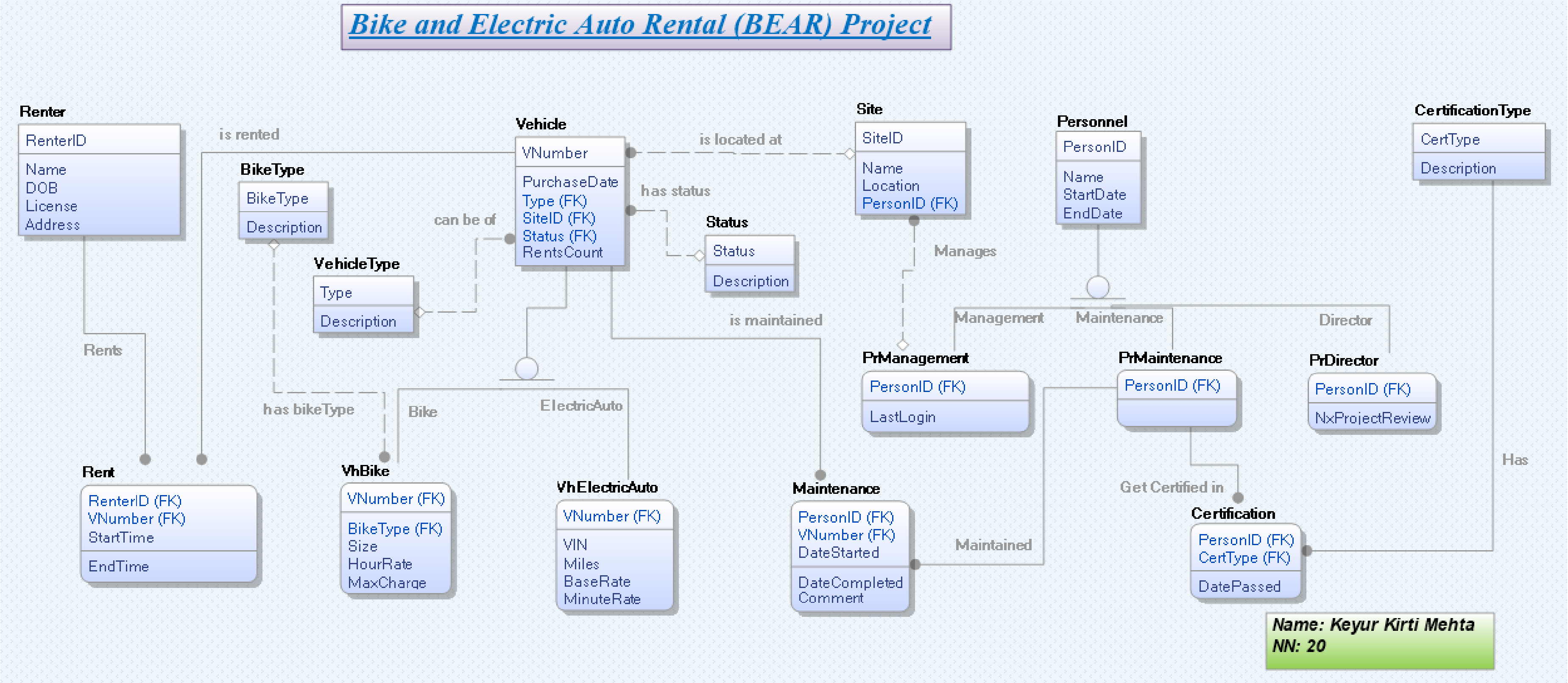
The problem statement is to design and implement a prototype of a database for company which has rental business of vehicles. The project is named as **Bike and Electric Auto Rental (BEAR)** project.

The company (customer) has various sites from which renters can rent the vehicle. There are 2 types of vehicles which can be rented. One Bike and other is electric Auto. Bick can also be of 2 types Road bike and Tandem bike. This model will keep track of all rentals made by the renters. It also keep records of personal details of renters.

The company wants to keep track of its personnel via this model database. It keeps track of managers, the maintenance personnel along with their certification details. This model also include records of vehicle maintenance detail along with detail of personal carried out the task.

The database prototype should be Relational Database Design in normalized form (5NF). The design should comply the data access rights (user can view and insert/update data as per the access assigned to him). The design should have integrity, referential constraint.

2) Logical Data Model (LDM)



3) Functional Dependency Analysis

Based on the analysis of universal relations, following functional dependencies may be identified:

1. FDs of Universal Relation

i. RenterID

a. RenterID → DOB
b. RenterID → License
c. RenterID → Address

ii. DOB

a. DOB → RenterID
b. DOB → License
c. DOB → Address

iii. Vnumber

a. Vnumber → PurchaseDate
b. Vnumber → Status
c. Vnumber → SiteID
d. Vnumber → Type
e. Vnumber → Size
f. Vnumber → BType
g. Vnumber → HourRate
h. Vnumber → MaxCharge
i. Vnumber → VIN
j. Vnumber → BaseRate
k. Vnumber → MinuteRate

iv. License

a. License → RenterID
b. License → DOB
c. License → Address

v. StartTime

a. StartTIme → Type
b. StartTIme → BType
c. StartTIme → VIN
d. StartTIme → Miles
e. StartTIme → BaseRate

f. StartTIme → MinuteRate

vi. EndTime

a. EndTime → Type
b. EndTime → BType
c. EndTime → VIN
d. EndTime → Miles
e. EndTime → BaseRate
f. EndTime → MinuteRate

vii. PurchaseDate

a. PurchaseDate → SiteID
b. PurchaseDate → Type
c. PurchaseDate → BaseRate
d. PurchaseDate → MinuteRate

viii. Address

a. Address → RenterID
b. Address → DOB
c. Address → License

ix. Size

a. Size → Type

x. BType

a. BType → Type

xi. HourRate

a. HourRate → Type
b. HourRate → Size
c. HourRate → BType
d. HourRate → MaxCharge

xii. MaxCharge

a. MaxCharge → Type
b. MaxCharge → Size
c. MaxCharge → BType
d. MaxCharge → HourRate

xiii. VIN

a. VIN → Type
b. VIN → BaseRate

c. VIN → MinuteRate

xiv. Miles

a. Miles → Type
b. Miles → VIN
c. Miles → BaseRate
d. Miles → MinuteRate

xv. BaseRate

a. BaseRate → Type
b. BaseRate → MinuteRate

xvi. MinuteRate

a. MinuteRate → Type

2. FDs from Memo # 1

Apart from this below FDs are also applicable based on the questions and answers received.

Name
RenterID →Name
Name →RenterID

Site Location
SiteID →Name
SiteID →Location
Name →SiteID
Name →Location
Location →SiteID
Location →Name

3. FDs of Personnel Data

PersonID →Name
PersonID →StartDate
PersonID →EndDate
Name → EndDate

4. FDs of Certification Data

CertType → Desc

Desc → CertType
DatePassed → CertType
DatePassed → Desc

5. FDs of Maintenance Data

Vnumber → PersonID
Vnumber → Comment

DateStarted → PersonID
DateStarted → Vnumber
DateStarted → DateCompleted
DateStarted → Comment

DateCompleted → PersonID
DateCompleted → Vnumber
DateCompleted → DateStarted
DateCompleted → Comment

Comment → PersonID
Comment → Vnumber

4) Construction of relations

Based on above FDs and enterprise statement following relations (table) of 3NF can be formed:

1. Renter

RenterID	Name	DOB	License	Address

Basic Renter information is stored in this relation.

2. Site

SiteID	Name	Location	PersonID

As per enterprise statement, each site is managed by site manager. So, PersonID is added.

3. Vehicle

VNumber	PurchaseDate	Type	SiteID	Status	RentCount

Common vehicle attributes are stored in the vehicle table (relation). And based on the type of the vehicle, there specific attributes are stored in sub type relations.
Sub type is used for the super entity vehicle with 2 sub types: VhBike for Bike and VhElectricAuto for Electric Auto.

4. VhBike

VNumber	BikeType	Size	HourRate	MaxCharge

5. VhElectricAuto

VNumber	VIN	Miles	BaseRate	MinuteRate

6. Rent

RenterID	VNumber	StartTime	End Time

Vehicle rent details are stored in this table which shows which renter has rented which vehicle.

7. BikeType

BikeType	Description

Type of Bike is described in this table.

8. Status

Status	Description

Vehicle status is described in this table.

9. VehicleType

Type	Description

Vehicle Type is described in this table.

10. Personnel

PersonID	Name	StartDate	EndDate

Personnel can be of management, Director and maintenance. So, sub type has been created for the personnel entity.

11. PrManagement

PersonID	LastLogin

Management personnel are listed in this table.

12. PrMaintenance

PersonID

Maintenance staff are listed in this table.

13. PrDirector

PersonID	NxProjReview

Project Director and its next project review date is listed in this table.

14. CertificationType

CertType	Description

Certification type is described in this table.

15. Certification

PersonID	CertType	DatePassed

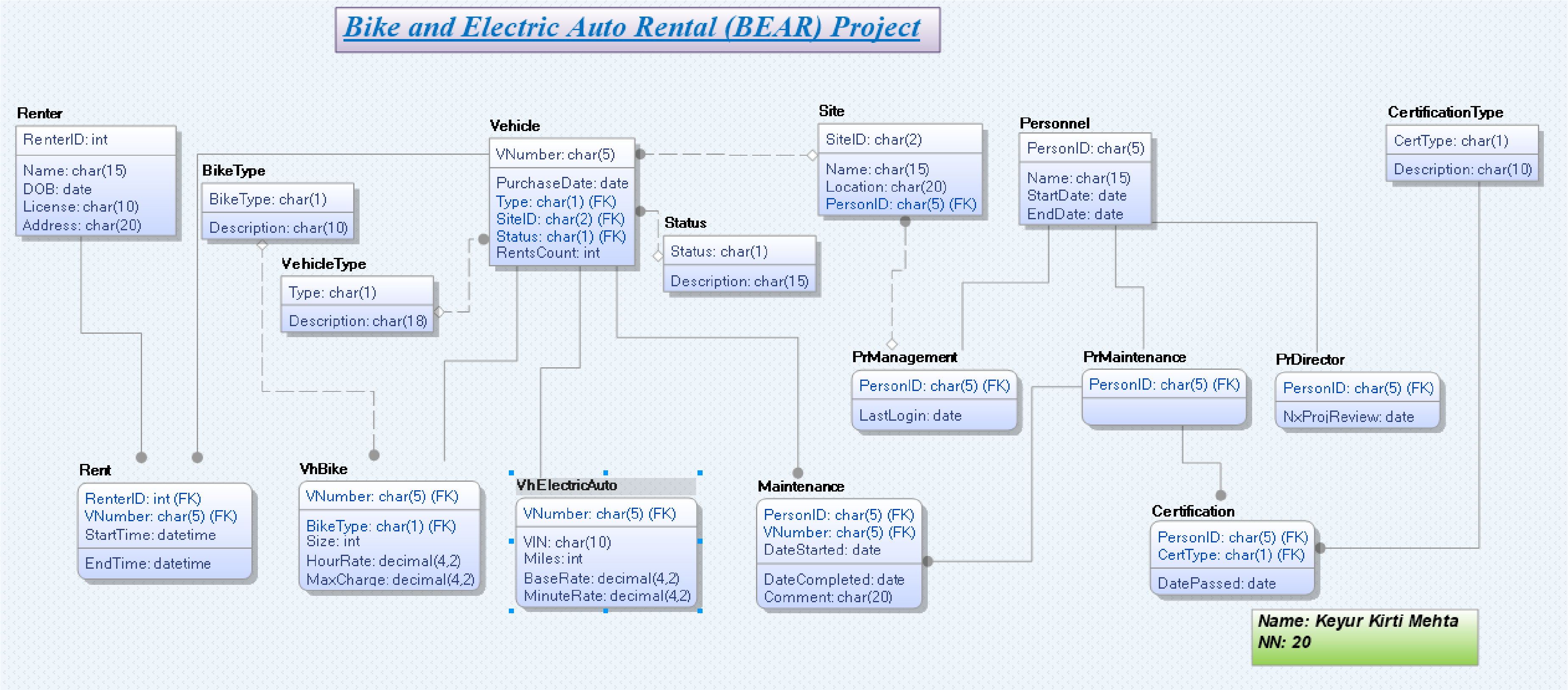
Maintenance Person’s certification details are stored in this table.

16. Maintenance

PersonID	Vnumber	DateStarted	DateCompleted	Comment

Vehicle maintenance data is stored in this table.

5) Physical Data Model (PDM)



6) List of Tables with Test Data

```
SELECT * FROM RENTER;
RenterID    Name            DOB        License    Address
----------- --------------- ---------- ---------- --------------------
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      1992-07-15 TX99923411 888 Tudor Dr

(6 row(s) affected)
```

```
SELECT * FROM BIKETYPE;
BikeType Description
-------- -----------
R        Road
T        Tandem

(2 row(s) affected)
```

```
SELECT * FROM STATUS;
Status Description
------ ---------------
A      Available
M      Maintenance
R      Rented

(3 row(s) affected)
```

```
SELECT * FROM VEHICLETYPE;
Type Description
---- ------------------
B    Bicycle
E    Electric Vehicle

(2 row(s) affected)
```

```
SELECT * FROM CERTIFICATIONTYPE;
CertType Description
-------- -----------
B        Bicycle
E        Electric
K        Kiosk

(3 row(s) affected)
```

```
SELECT * FROM PERSONNEL;
PersonID Name            StartDate  EndDate
-------- --------------- ---------- ----------
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 Grey      2017-01-10 NULL
P2400    Joe Brown       2017-01-10 NULL

(11 row(s) affected)


SELECT * FROM PRMANAGEMENT;
PersonID LastLogin
-------- ----------
P0101    NULL
P0102    NULL
P0103    NULL

(3 row(s) affected)


SELECT * FROM PRMAINTENANCE;
PersonID
--------
P2108
P2109
P2202
P2303

(4 row(s) affected)


SELECT * FROM PRDIRECTOR;
PersonID NxProjReview
-------- ------------
P0000    NULL

(1 row(s) affected)


SELECT PersonID
,CertType
,CONVERT(VARCHAR(10), DatePassed, 101) AS DatePassed
FROM CERTIFICATION;

PersonID CertType DatePassed
-------- -------- ----------
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
```

```
(7 row(s) affected)
```

*Note: Data is represented in the similar form as it was received from the customer, for better understanding of the customers.

```
SELECT * FROM SITE;
SiteID Name            Location             PersonID
------ --------------- -------------------- --------
AS     Airport Site    44 Airport Dr.       P0102
DS     Downtown Site   423 Main St.         P0101
SS     Suburban Site   22 Center Circle     P0103

(3 row(s) affected)
```

```
SELECT Vnumber
,SUBSTRING(CONVERT(varchar(11),PurchaseDate,106),4,8) AS PurchaseDate
,Type
,SiteID
,Status
,RentsCount
FROM VEHICLE;

Vnumber PurchaseDate Type SiteID Status RentsCount
------- ------------ ---- ------ ------ -----------
B0001   Aug 2016     B    SS     R      3
B0003   Aug 2016     B    SS     A      1
B0010   NULL         NULL NULL   NULL   NULL
B0011   Oct 2016     B    DS     A      3
E0012   Sep 2016     E    AS     R      4
E0014   Sep 2016     E    AS     R      3
E0444   Nov 2016     E    SS     A      3
E0523   Dec 2016     E    SS     A      0
E0524   Dec 2016     E    SS     M      0
T0002   Aug 2016     B    SS     A      4

(10 row(s) affected)
```

*Note: Data is represented in the similar form as it was received from the customer, for better understanding of the customers.

```
SELECT * FROM VHBIKE;
VNumber BikeType Size        HourRate                                MaxCharge
------- -------- ----------- --------------------------------------- -------------------------
--------------
B0001   R        26          8.00                                    56.00
B0003   R        24          6.00                                    43.00
B0011   R        26          8.00                                    56.00
T0002   T        26          10.00                                   70.00

(4 row(s) affected)
```

```
SELECT * FROM VHELECTRICAUTO;
VNumber VIN        Miles       BaseRate                                MinuteRate
------- ---------- ----------- --------------------------------------- -----------------------
----------------
E0012   EV10000234 2592        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   EV10000234 10          11.00                                   0.50

(5 row(s) affected)
```

*Note: Data is represented in the similar form as it was received from the customer, for better understanding of the customers.

```
SELECT * FROM MAINTENANCE;

PersonID VNumber DateStarted DateCompleted Comment
-------- ------- ----------- ------------- --------------------
P2108    B0001   2017-11-10  2017-11-10    Broken Chain
P2108    E0012   2017-01-10  2017-01-10    Broken wiper
P2109    B0003   2017-01-11  2017-01-11    Flat tire
P2109    B0003   2017-11-11  2017-11-11    Flat tire
P2202    B0010   2017-01-20  2017-01-20    New wheel
P2303    E0524   2017-12-20  2017-12-21    Initial checks

(6 row(s) affected)
```

```
SELECT RenterID
,Vnumber
,SUBSTRING(CONVERT(varchar(23),StartTime,121),1,16) AS StartTime
,SUBSTRING(CONVERT(varchar(23),EndTime,121),1,16) AS EndTime
FROM RENT;

RenterID    Vnumber StartTime        EndTime
----------- ------- ---------------- ----------------
1           B0001   2016-12-10 10:30 2016-12-10 12:30
1           B0011   2016-12-16 09:30 2016-12-16 12:30
1           B0011   2016-12-16 13:30 2016-12-16 16:30
1           E0012   2016-11-26 09:30 2016-11-26 18:30
2           E0012   2016-11-27 10:30 2016-11-27 18:30
2           E0012   2016-12-27 10:30 2016-12-27 18:30
2           E0014   2016-11-26 10:30 2016-11-26 14:30
3           B0001   2016-12-11 10:00 2016-12-11 16:00
4           B0001   2016-12-16 10:30 2016-12-16 18:30
4           E0444   2016-11-29 09:30 2016-11-30 09:30
8           B0003   2016-12-16 09:30 2016-12-16 12:30
8           E0012   2016-10-14 09:30 2016-10-15 13:30
8           E0523   2016-12-26 09:30 2017-01-02 18:30
8           T0002   2016-12-19 10:30 2016-12-19 12:30
9           B0011   2016-12-10 10:30 2016-12-10 12:30
9           E0524   2016-12-29 06:00 2016-12-31 13:00

(16 row(s) affected)
```

*Note: Data is represented in the similar form as it was received from the customer, for better understanding of the customers.

**Note: Values which were missing in universal relations, are represented as NULL in all the relations

7) Query Results

1. Queries from Memo # 3

1. What electric vehicle has the largest mileage?

Query:

```
SELECT TOP 1 VE.Vnumber
,MAX(VE.Miles) AS Max_Mileage
FROM VHELECTRICAUTO VE
GROUP BY VE.Vnumber
ORDER BY Max_Mileage DESC;
```

```
Output:
    Vnumber Max_Mileage
    ------- -----------
    E0014   3505

    (1 row(s) affected)
```

2. For each bicycle, list all the rentals in chronological (date) order

Query:

```
SELECT V.Vnumber
,R.RenterID
,R.Name
,SUBSTRING(CONVERT(varchar(23),RN.StartTime,121),1,16) AS StartTime
,SUBSTRING(CONVERT(varchar(23),RN.EndTime,121),1,16) AS EndTime
,vt.description AS VehicleType
FROM vehicle V, rent RN, Renter R, vehicleType vt
WHERE V.Vnumber = RN.Vnumber
AND RN.RenterID = R.RenterID
AND vt.type = v.type
AND V.type = 'B'
ORDER BY RN.StartTime;
```

Output:

```
Vnumber RenterID    Name            StartTime        EndTime          VehicleType
------- ----------- --------------- ---------------- ---------------- -----------------
-
B0001   1           Anne Green      2016-12-10 10:30 2016-12-10 12:30 Bicycle
B0011   9           Bill Brown      2016-12-10 10:30 2016-12-10 12:30 Bicycle
B0001   3           Bill Smith      2016-12-11 10:00 2016-12-11 16:00 Bicycle
B0011   1           Anne Green      2016-12-16 09:30 2016-12-16 12:30 Bicycle
B0003   8           Evan Black      2016-12-16 09:30 2016-12-16 12:30 Bicycle
B0001   4           Gail White      2016-12-16 10:30 2016-12-16 18:30 Bicycle
B0011   1           Anne Green      2016-12-16 13:30 2016-12-16 16:30 Bicycle
T0002   8           Evan Black      2016-12-19 10:30 2016-12-19 12:30 Bicycle

(8 row(s) affected)
```

3. What is the location of the downtown site?

Query:

```
SELECT Name
,Location
FROM Site
WHERE Name = 'Downtown Site';
```

```
 Output:
Name            Location
--------------- --------------------
Downtown Site   423 Main St.

(1 row(s) affected)
```

4. List top 2 bicycles based on the no of hours rented

Query:

```
SELECT TOP(2) A.Vnumber, A.VehicleType, A.TotalHoursRented
FROM (SELECT V.Vnumber, SUM(datediff(mi, R.startTime, endTime) / 60.0) As
TotalHoursRented
,vt.Description AS VehicleType
FROM Rent R, Vehicle V, vehicleType vt
WHERE R.Vnumber = V.Vnumber
AND vt.type = v.type
AND V.type = 'B'
       GROUP BY V.Vnumber, vt.Description) A
       ORDER BY A.TotalHoursRented DESC;
```

Output:

```
Vnumber VehicleType        TotalHoursRented
------- ------------------ ---------------------------------------
B0001   Bicycle            16.000000
B0011   Bicycle            8.000000

(2 row(s) affected)
```

5. List the electric automobile rentals made in October 2016 and in December 2016 at the airport site

Query:

```
SELECT V.Vnumber
,R.RenterID
,R.Name
,SUBSTRING(CONVERT(varchar(23),RN.StartTime,121),1,16) AS StartTime
,SUBSTRING(CONVERT(varchar(23),RN.EndTime,121),1,16) AS EndTime
,s.name AS SiteName
,Vt.description AS VehicleType
FROM vehicle V, rent RN, Renter R, vehicleType vt, site s
WHERE V.Vnumber = RN.Vnumber
AND RN.RenterID = R.RenterID
AND s.siteid = v.siteid
AND vt.Type = v.Type
AND V.type = 'E'
AND V.SiteID = 'AS'
AND (RN.StartTime BETWEEN '2016-10-01' AND '2016-10-31'
OR RN.StartTime BETWEEN '2016-12-01'AND'2016-12-31');
```

Output:

```
Vnumber RenterID    Name            StartTime        EndTime          SiteName
VehicleType
------- ----------- --------------- ---------------- ---------------- --------------- -
----------------
E0012   2           John Dough      2016-12-27 10:30 2016-12-27 18:30 Airport Site
Electric Vehicle
E0012   8           Evan Black      2016-10-14 09:30 2016-10-15 13:30 Airport Site
Electric Vehicle

(2 row(s) affected)
```

6. What vehicle are currently available for rent?

Query:

```
SELECT V.Vnumber
,S.Description AS VehicleStatus
FROM Vehicle V, Status S
WHERE S.Status = V.Status
AND S.Status = 'A';
```

Output:

```
Vnumber VehicleStatus
------- ---------------
B0003   Available
B0011   Available
E0444   Available
E0523   Available
T0002   Available

(5 row(s) affected)
```

7. List the renters and the types of bicycles they rented?

Query:

```
SELECT DISTINCT R.RenterID
,R.Name
,bt.Description AS BikeType
FROM vehicle V, rent RN, Renter R, VhBike VB, BikeType bt
WHERE V.Vnumber = RN.Vnumber
AND RN.RenterID = R.RenterID
AND V.Vnumber = VB.Vnumber
AND bt.BikeType = vb.BikeType
AND V.type = 'B';
```

Output:

```
RenterID    Name            BikeType
----------- --------------- ----------
1           Anne Green      Road
3           Bill Smith      Road
```

```
4           Gail White      Road
8           Evan Black      Road
8           Evan Black      Tandem
9           Bill Brown      Road

(6 row(s) affected)
```

2. Queries from Memo # 4

8. Who (Name and id) manages the Downtown Site?

Query:

```sql
SELECT S.personID
,p.name
,S.name AS SiteName
FROM Site S, PrManagement Pm, Personnel P
WHERE S.PersonID = pm.personID
AND pm.personID = p.personID
AND S.name = 'DownTown Site';
```

Query:

```
personID name            SiteName
-------- --------------- ---------------
P0101    Mary Manager    Downtown Site

(1 row(s) affected)
```

9. Who (Name and ID) has access to suburban Site vehicle Data?

Query:

```sql
SELECT S.personID
,p.name
,S.name AS AccessToVehicleData
FROM Site S, PrManagement Pm, Personnel P
WHERE (S.PersonID = pm.personID
AND pm.personID = p.personID)
AND S.name = 'Suburban Site'
UNION
SELECT DISTINCT p.personid
,p.name
,S.name AS AccessTo
FROM Vehicle V, maintenance m, PrMaintenance Prm, Site S, Personnel P
WHERE s.siteid = v.siteid
AND v.vnumber = m.VNumber
AND m.PersonID = prm.PersonID
AND prm.PersonID = p.PersonID
AND S.name = 'Suburban Site';
```

Output:

```
personID name            AccessToVehicleData
-------- --------------- -------------------
P0103    Sam Supervisor  Suburban Site
```

```
P2108    Joe Brown       Suburban Site
P2109    Jane Smith      Suburban Site
P2303    Mike Green      Suburban Site

(4 row(s) affected)
```

10. How many times has each vehicle actually been rented?

Query:

```
SELECT DISTINCT Vnumber
,RentsCount
FROM Vehicle;
```

```
Output:
Vnumber RentsCount
------- -----------
B0001   3
B0003   1
B0010   NULL
B0011   3
E0012   4
E0014   3
E0444   3
E0523   0
E0524   0
T0002   4

(10 row(s) affected)
```

11. List the vehicles maintained by P2108

Query:

```
SELECT DISTINCT PersonID
,Vnumber
FROM Maintenance
WHERE PersonID = 'P2108';
```

Output:

```
PersonID Vnumber
-------- -------
P2108    B0001
P2108    E0012

(2 row(s) affected)
```

12. At the suburban site, how many income resulted from the

a. Bicycle rentals?

Query:

```
SELECT SUM(RentAmount) AS TotalBikeRentalIncomeAtSuburbanSite
FROM (SELECT A.Vnumber
,CASE WHEN (A.Hours_Difference * A.HourRate) <= A.MaxCharge THEN
(A.Hours_Difference * A.HourRate)
WHEN(A.Hours_Difference * A.HourRate) > A.MaxCharge THEN A.MaxCharge
END AS RentAmount
,A.MaxCharge
FROM (SELECT DATEDIFF(hh, R.starttime, R.endtime) as Hours_Difference
,R.Vnumber
,Vb.HourRate
,Vb.MaxCharge
FROM Vehicle V, Rent R, vhbike vb
WHERE R.Vnumber = V.Vnumber
AND V.VNumber = vb.VNumber
AND V.SiteID = 'SS'
AND V.Type = 'B') A) B;
```

Output:

```
TotalBikeRentalIncomeAtSuburbanSite
---------------------------------------
158.00

(1 row(s) affected)
```

b. Electric automobiles rentals (Optional: 1 point bonus)

Query:

```
SELECT SUM(RentAmount) AS TotalBikeRentalIncomeAtSuburbanSite
FROM (SELECT A.Vnumber
,(A.BaseRate + (A.minute_Difference * A.minuteRate)) AS RentAmount
FROM (SELECT R.Vnumber
,DATEDIFF(mi, R.starttime, R.endtime) as minute_Difference
,Ve.BaseRate
,Ve.MinuteRate
FROM Vehicle V, Rent R, vhElectricAuto ve
WHERE R.Vnumber = V.Vnumber
AND V.VNumber = ve.VNumber
AND V.SiteID = 'SS'
AND V.Type = 'E') A ) B;
```

Output:

```
TotalBikeRentalIncomeAtSuburbanSite
---------------------------------------
7568.00

(1 row(s) affected)
```

13. List the vehicle, type and date rented for all vehicles rented by Anne Green

Query:

```
SELECT R.RenterID
,R.Name
,Re.Vnumber
,Vt.Description
,SUBSTRING(CONVERT(varchar(23),Re.StartTime,121),1,16) AS StartTime
```

```
,SUBSTRING(CONVERT(varchar(23),Re.EndTime,121),1,16) AS EndTime
FROM Renter R, Rent Re, Vehicle V, VehicleType vt
WHERE R.RenterID = Re.RenterID
AND Re.Vnumber = V.Vnumber
AND vt.type = v.type
AND R.Name = 'Anne Green';
```

Output:

```
RenterID    Name            Vnumber Description        StartTime        EndTime
----------- --------------- ------- ------------------ ---------------- ---------------
-
1           Anne Green      B0001   Bicycle            2016-12-10 10:30 2016-12-10
12:30
1           Anne Green      B0011   Bicycle            2016-12-16 09:30 2016-12-16
12:30
1           Anne Green      B0011   Bicycle            2016-12-16 13:30 2016-12-16
16:30
1           Anne Green      E0012   Electric Vehicle   2016-11-26 09:30 2016-11-26
18:30

(4 row(s) affected)
```

8) Memo # 7 (Last minute Data effect on model)

1. New Data from Bob
   - **Tuple # 1** cannot be inserted as RenterID 5 is not present in the parent table Renter. Referential integrity constraint will fail.
     Also StartTime field is null for the tuple which is part of composite primary key for relation (table) Rent. And primary key field cannot have null value.
     New Renter tuple needs to be added in Renter table in order to insert this tuple.
   - **Tuple # 2** can be inserted as it is satisfying both the conditions, RenterID 8 is present in the master table and also satisfying primary key constraint of rent relation (table).

2. Information from Sam
   - **New Site**
     Adding this data will require change in design in Site table where datatype for Name needs to be increased from char(15) to char(20) as new name consist of more than 15 characters.
     Once it is altered, new site can be added to the relation (table) Site. It will require 1 INSERT statement. It will just require additional information as Site ID and Location. With this, data can be inserted.
   - **Hourly Rate of Tandem**
     This data can be updated with just 1 UPDATE query. It will require Vnumber of tandem for which hourly rate needs to be updated. Update query needs to fire on table VhBike.

3. Data items from Bob that look Different
   - Vnumber E0012 has different status, #Rents and siteID. As Vnumber is primary key in Vehicle table new entry cannot be made for this vehicle data. It will be integrity constraint violation. Field value for Vnumber E0012 needs to be updated as mentioned in memo.
   - Once updated, then new entry can be made in Rent table as all the primary key fields are present and they are not violating the constraint. In start time field time is not mentioned, SO, SQL server will take 00:00:00 as default time.

4. Number of samples field
   - Number of rentals fields (RentsCount) can be removed as it neither have primary key nor foreign key constraint on it.
   - It just there will be problem while generating report, which vehicle is rented how many times? But this can also be solved by querying it on rent table to get count of renterid and group by Vnumber. This way we can get no of times each vehicle is rented.

5. Overlapping Dates

- Vehicle table has 1 status field. Once bike is rented this field can be change to ‘R’ (Rented). And when vehicle is returned back its status should change to ‘Available’
- Now before insert trigger can be written on Rent table. Which will check this field for particular vehicle. If status is ‘R’ then cannot insert the new tuple.

Appendix

1. *Grid mode analysis of the universal Relation*

Based on the given universal relation grid mode is constructed to identify the non-trivial dependencies.

	RenterID	DOB	Vnumber	License	StartTime	EndTime	PurchaseDate	Status	#Rents	SiteID	Address	Type	Size	Type (Btype)	HourRate	MaxCharge	VIN	Miles	BaseRate	MinuteRate
RenterID	--	Y	1, 2	Y	1, 2	1, 2	1, 2	1, 2	1, 2	1, 2	Y	1, 8	5, 6	5, 6	5, 6	5, 6	9, 10	9, 10	12, 13	12, 13
DOB	Y	--	1, 2	Y	1, 2	1, 2	1, 2	1, 2	1, 2	1, 2	Y	1, 8	5, 6	5, 6	5, 6	5, 6	9, 10	9, 10	12, 13	12, 13
Vnumber	1, 3	1, 3	--	1, 3	1, 3	1, 3	Y	Y	7, 7a	Y	1, 3	Y	Y	Y	Y	Y	Y	9, 15	Y	Y
License	Y	Y	1, 2	--	1, 2	1, 2	1, 2	1, 2	1, 2	1, 2	Y	1, 8	5, 6	5, 6	5, 6	5, 6	9, 10	9, 10	12, 13	12, 13
StartTime	1,7	1, 7	1,7	1,7	--	NA	1, 7	1, 7	1, 7	1, 7	1, 7	Y	2, 5	Y	2, 5	2, 5	Y	Y	Y	Y
EndTime	2, 5	2, 5	2, 5	2, 5	NA	--	2, 5	1, 7	1, 7	1, 7	1, 7	Y	2, 5	Y	2, 5	2, 5	Y	Y	Y	Y
PurchaseDate	1, 3	1, 3	1, 5	1, 5	1, 5	1, 5	--	1, 5	1, 5	Y	1, 5	Y	1, 5	5, 6	5, 6	5, 6	9, 10	9, 10	Y	Y
Status	1, 3	1, 3	5, 6	1, 3	1, 3	1, 3	2, 5	--	2, 5	2, 5	2, 5	1, 8	2, 5	5, 6	5, 6	5, 6	9, 10	9, 10	11, 13	11, 13
#Rents	1, 3	1, 3	1, 8	1, 3	1, 3	1, 3	1, 8	1, 11	--	1, 8	1, 3	1, 8	1, 8	1, 8	5, 6	5, 6	10, 11	10, 11	10, 11	1, 8
SiteID	1, 3	1, 3	4, 5	4, 5	4, 5	4, 5	1, 11	4, 5	4, 5	--	4, 5	1, 11	5, 6	5, 6	5, 6	5, 6	9, 10	9, 10	11, 13	11, 13
Address	Y	Y	1, 2	Y	1, 2	1, 2	1, 2	1, 2	1, 2	1, 2	--	1, 8	5, 6	5, 6	5, 6	5, 6	9, 10	9, 10	12, 13	12, 13

	RenterID	DOB	Vnumber	License	StartTime	EndTime	PurchaseDate	Status	#Rents	SiteID	Address	Type	Size	Type (Btype)	HourRate	MaxCharge	VIN	Miles	BaseRate	MinuteRate
Type	1, 3	1, 3	1, 2	1, 3	1, 2	1, 2	1, 2	1, 2	1, 2	1, 2	1, 3	--	5, 6	5, 6	5, 6	5, 6	9, 10	9, 10	12, 13	12, 13
Size	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	**Y**	--	1, 6	1, 6	1, 6	9, 10	9, 10	12, 13	12, 13
Type (Btype)	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	**Y**	4, 5	--	4, 5	4, 5	9, 10	9, 10	12, 13	12, 13
HourRate	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	**Y**	**Y**	**Y**	--	**Y**	9, 10	9, 10	12, 13	12, 13
MaxCharge	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	**Y**	**Y**	**Y**	**Y**	--	9, 10	9, 10	12, 13	12, 13
VIN	8, 9	8, 9	1, 2	8, 9	8, 9	8, 9	1, 2	1, 2	1, 2	1, 2	8, 9	**Y**	5, 6	5, 6	5, 6	5, 6	--	9, 15	**Y**	**Y**
Miles	8, 9	8, 9	1, 2	8, 9	8, 9	8, 9	1, 2	1, 2	1, 2	1, 2	8, 9	**Y**	5, 6	5, 6	5, 6	5, 6	**Y**	--	**Y**	**Y**
BaseRate	8, 9	8, 9	9, 10	8, 9	8, 9	8, 9	1, 2	13, 14	1, 2	1, 2	8, 9	**Y**	5, 6	5, 6	5, 6	5, 6	9, 10	9, 10	--	**Y**
Minute Rate	8, 9	8, 9	9, 10	8, 9	8, 9	8, 9	10, 11	10, 11	1, 2	11, 12	8, 9	**Y**	5, 6	5, 6	5, 6	5, 6	9, 10	9, 10	10, 11	--

Note: Cells highlighted in blue are not satisfying the FDs condition based on the additional data provided in memo # 5.

- Personnel Data

	PersonNo	Name	StartDate	EndDate
PersonNo	--	Y	Y	Y
Name	1, 7	--	1, 7	Y
StartDate	1, 2	1, 2	--	4, 5
EndDate	1, 2	1, 2	2, 3	--

- Certification Data

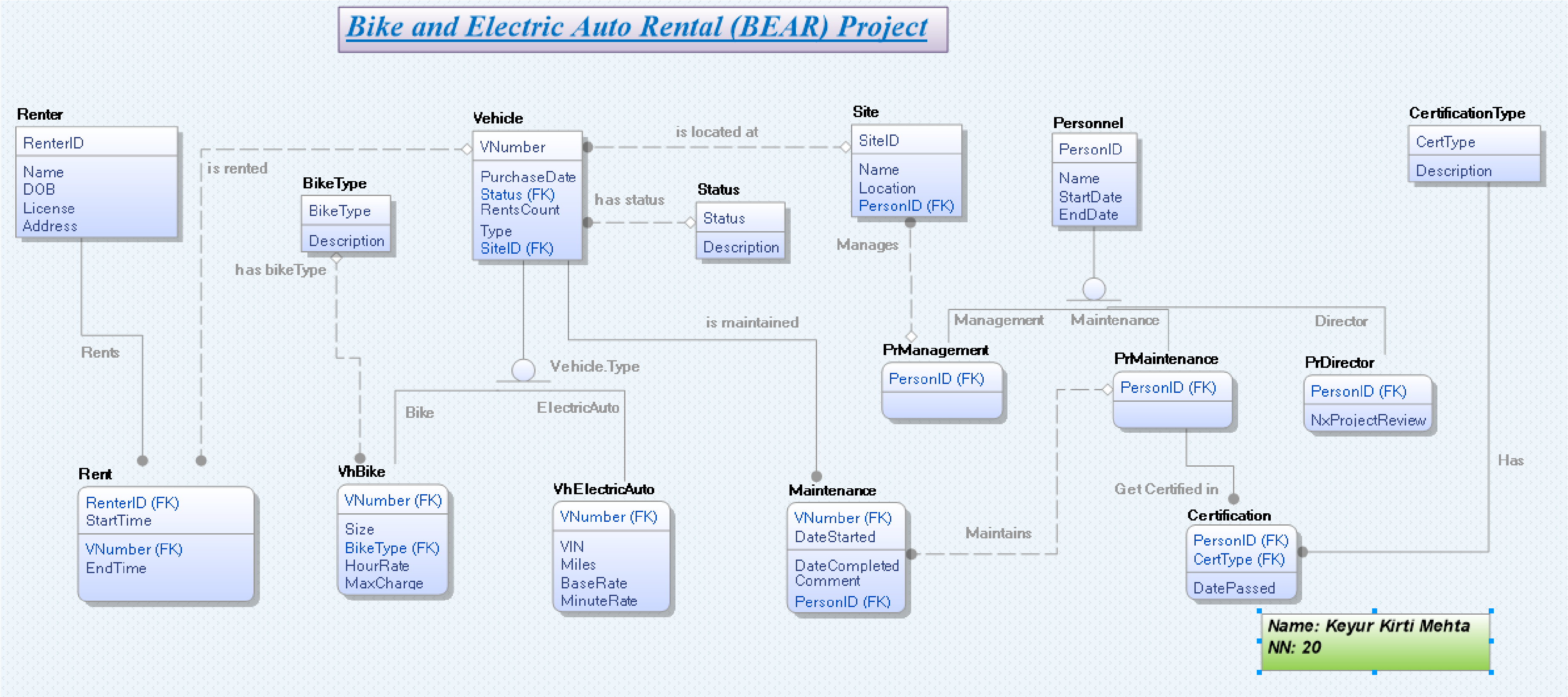
	PersonNo	CertType	Desc	DatePassed
PersonNo	--	1, 2	1, 2	1, 2
CertType	1, 4	--	Y	5, 6
Desc	1, 4	Y	--	5, 6
DatePassed	1, 4	Y	Y	--

- Maintenance Task Data

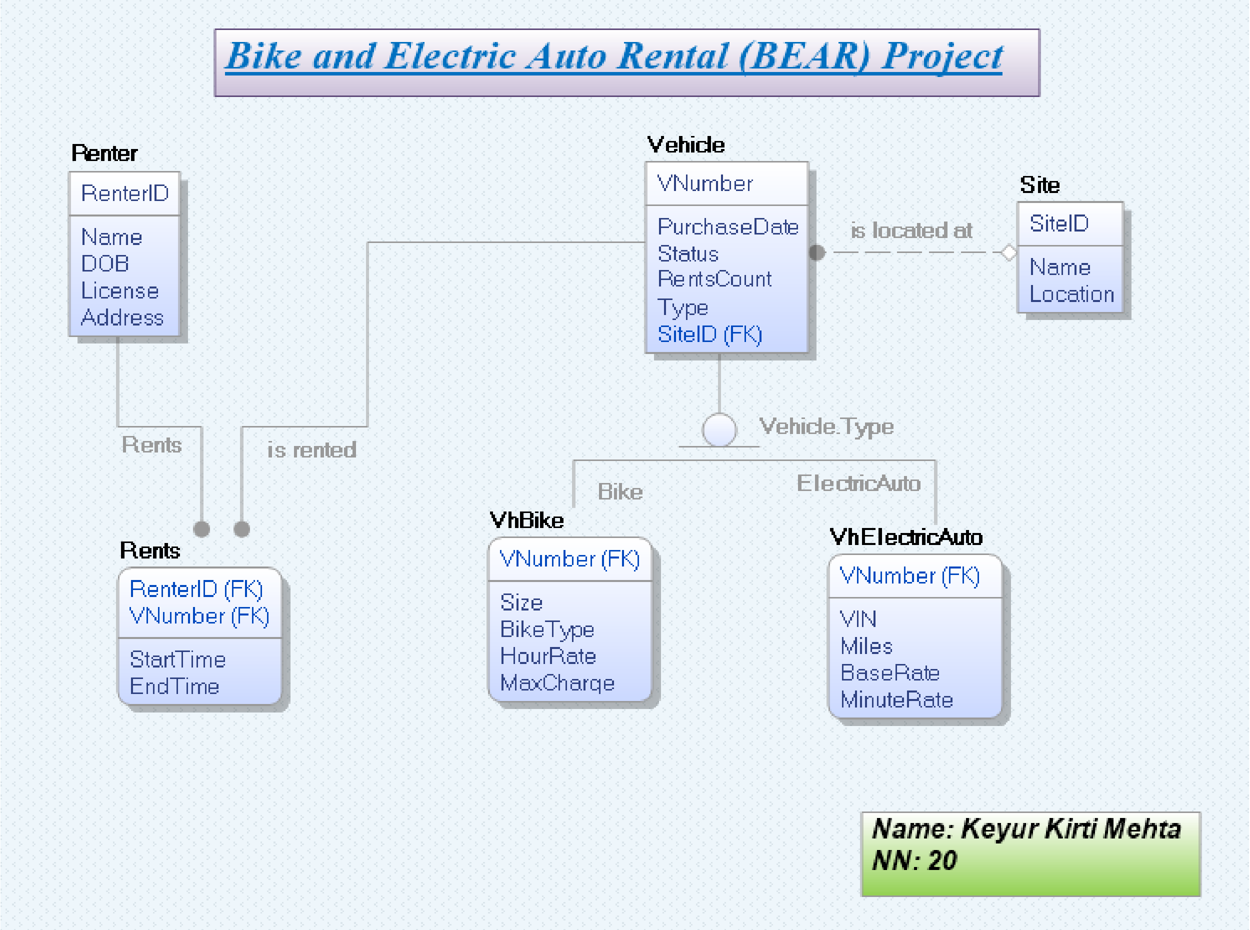
	PersonID	Vnumber	DateStarted	DateCompleted	Comment
PersonID	--	1, 3	1, 3	1, 3	1, 3
Vnumber	Y	--	2,4	2,4	Y
DateStarted	Y	Y	--	Y	Y
DateCompleted	Y	Y	Y	--	Y
Comment	Y	Y	2, 4	2, 4	--

2. Previous Logical Data Model

a. Version 2



b. Version 1



c. Steps to revise LDM

1. As per updated universal relation provided in memo # 4, functional dependency grid is revised. Based on the data all non-trivial FDs are checked again if it still true.
2. Also newly introduced entity is examined for the FDs analysis and grid is created to find the functional dependencies.
3. Based on the above steps new set of FDs are identified.
4. New relations are created and old relations are revised based on the current FDs.
5. Logical Data Model is created using erWin.

3. ***Prototype of Relational Database Design (RDD) generated by ErWin***

```
CREATE TABLE Personnel
(
   PersonID             char(5)  NOT NULL ,
   Name                 char(15)  NULL ,
   StartDate            date  NULL ,
   EndDate              date  NULL ,
    PRIMARY KEY  CLUSTERED (PersonID ASC)
)
go


CREATE TABLE PrManagement
(
   PersonID             char(5)  NOT NULL ,
   LastLogin            date  NULL ,
    PRIMARY KEY  CLUSTERED (PersonID ASC),
    FOREIGN KEY (PersonID) REFERENCES Personnel(PersonID)
)
go

CREATE TABLE Site
(
   SiteID               char(2)  NOT NULL ,
   Name                 char(15)  NULL ,
   Location             char(20)  NULL ,
   PersonID             char(5)  NULL ,
    PRIMARY KEY  CLUSTERED (SiteID ASC),
    FOREIGN KEY (PersonID) REFERENCES PrManagement(PersonID)
)
go

CREATE TABLE Status
(
   Status               char(1)  NOT NULL ,
   Description          char(15)  NULL ,
    PRIMARY KEY  CLUSTERED (Status ASC)
)
go

CREATE TABLE VehicleType
(
   Type                 char(1)  NOT NULL ,
   Description          char(18)  NULL ,
    PRIMARY KEY  CLUSTERED (Type ASC)
)
go

CREATE TABLE Vehicle
(
   VNumber              char(5)  NOT NULL ,
   PurchaseDate         date  NULL ,
   Type                 char(1)  NULL ,
   SiteID               char(2)  NULL ,
   Status               char(1)  NULL ,
   RentsCount           int  NULL ,
    PRIMARY KEY  CLUSTERED (VNumber ASC),
    FOREIGN KEY (SiteID) REFERENCES Site(SiteID),
 FOREIGN KEY (Status) REFERENCES Status(Status),
 FOREIGN KEY (Type) REFERENCES VehicleType(Type)
)
go

CREATE TABLE VhElectricAuto
(
   VNumber              char(5)  NOT NULL ,
   VIN                  char(10)  NULL ,
   Miles                int  NULL ,
   BaseRate             decimal(4,2)  NULL ,
   MinuteRate           decimal(4,2)  NULL ,
    PRIMARY KEY  CLUSTERED (VNumber ASC),
    FOREIGN KEY (VNumber) REFERENCES Vehicle(VNumber)
)
go

CREATE TABLE BikeType
(
   BikeType             char(1)  NOT NULL ,
   Description          char(10)  NULL ,
    PRIMARY KEY  CLUSTERED (BikeType ASC)
)
go

CREATE TABLE VhBike
(
   VNumber              char(5)  NOT NULL ,
   BikeType             char(1)  NULL ,
   Size                 int  NULL ,
   HourRate             decimal(4,2)  NULL ,
   MaxCharge            decimal(4,2)  NULL ,
    PRIMARY KEY  CLUSTERED (VNumber ASC),
    FOREIGN KEY (VNumber) REFERENCES Vehicle(VNumber),
 FOREIGN KEY (BikeType) REFERENCES BikeType(BikeType)
)
go

CREATE TABLE Renter
(
   RenterID             int  NOT NULL ,
   Name                 char(15)  NULL ,
   DOB                  date  NULL ,
   License              char(10)  NULL ,
   Address              char(20)  NULL ,
    PRIMARY KEY  CLUSTERED (RenterID ASC)
)
go

CREATE TABLE Rent
(
   RenterID             int  NOT NULL ,
   VNumber              char(5)  NOT NULL ,
   StartTime            datetime  NOT NULL ,
   EndTime              datetime  NULL ,
    PRIMARY KEY  CLUSTERED (RenterID ASC,VNumber ASC,StartTime ASC),
    FOREIGN KEY (RenterID) REFERENCES Renter(RenterID),
 FOREIGN KEY (VNumber) REFERENCES Vehicle(VNumber)
)
```

```
go

CREATE TABLE PrDirector
(
   PersonID             char(5)  NOT NULL ,
   NxProjReview         date  NULL ,
    PRIMARY KEY  CLUSTERED (PersonID ASC),
    FOREIGN KEY (PersonID) REFERENCES Personnel(PersonID)
)
go

CREATE TABLE PrMaintenance
(
   PersonID             char(5)  NOT NULL ,
    PRIMARY KEY  CLUSTERED (PersonID ASC),
    FOREIGN KEY (PersonID) REFERENCES Personnel(PersonID)
)
go

CREATE TABLE Maintenance
(
   PersonID             char(5)  NOT NULL ,
   VNumber              char(5)  NOT NULL ,
   DateStarted          date  NOT NULL ,
   DateCompleted        date  NULL ,
   Comment              char(20)  NULL ,
    PRIMARY KEY  CLUSTERED (PersonID ASC,VNumber ASC,DateStarted ASC),
    FOREIGN KEY (VNumber) REFERENCES Vehicle(VNumber),
 FOREIGN KEY (PersonID) REFERENCES PrMaintenance(PersonID)
)
go

CREATE TABLE CertificationType
(
   CertType             char(1)  NOT NULL ,
   Description          char(10)  NULL ,
    PRIMARY KEY  CLUSTERED (CertType ASC)
)
go

CREATE TABLE Certification
(
   PersonID             char(5)  NOT NULL ,
   CertType             char(1)  NOT NULL ,
   DatePassed           date  NULL ,
    PRIMARY KEY  CLUSTERED (PersonID ASC,CertType ASC),
    FOREIGN KEY (PersonID) REFERENCES PrMaintenance(PersonID),
 FOREIGN KEY (CertType) REFERENCES CertificationType(CertType)
)
Go
```

4. ***Test data loading queries***

BEAR Test Data

```
Renter
INSERT INTO RENTER VALUES(1, 'Anne Green', '1995-01-02', 'IN24340020', '233
Main St.');
INSERT INTO RENTER VALUES(2, 'John Dough', '1990-11-03', 'IN24388033', '123
1st St.');
INSERT INTO RENTER VALUES(3, 'Bill Smith', '2000-10-04', 'IN67800000', '444 W
State.');
INSERT INTO RENTER VALUES(4, 'Gail White', '1997-06-06', 'KY44230100', 'PO Box
2388.');
INSERT INTO RENTER VALUES(8, 'Evan Black', '1995-04-12', 'IN24500001', '42
Blvd Pl');
INSERT INTO RENTER VALUES(9, 'Bill Brown', '1992-07-15', 'TX99923411', '888
Tudor Dr');

BikeType
INSERT INTO BIKETYPE VALUES('R', 'Road');
INSERT INTO BIKETYPE VALUES('T', 'Tandem');


Status
INSERT INTO STATUS VALUES('R', 'Rented');
INSERT INTO STATUS VALUES('A', 'Available');
INSERT INTO STATUS VALUES('M', 'Maintenance');

VehicleType
INSERT INTO VEHICLETYPE VALUES('B', 'Bicycle');
INSERT INTO VEHICLETYPE VALUES('E', 'Electric Vehicle');

CertificationType
INSERT INTO CERTIFICATIONTYPE VALUES('B', 'Bicycle');
INSERT INTO CERTIFICATIONTYPE VALUES('E', 'Electric');
INSERT INTO CERTIFICATIONTYPE VALUES('K', 'Kiosk');

Personnel
INSERT INTO PERSONNEL VALUES('P2108', 'Joe Brown', '2016-11-03', NULL);
INSERT INTO PERSONNEL VALUES('P2109', 'Jane Smith', '2016-11-03', NULL);
INSERT INTO PERSONNEL VALUES('P2202', 'Frank Martin', '2016-11-05', NULL);
INSERT INTO PERSONNEL VALUES('P2213', 'Anne Dough', '2016-11-10', '2016-11-
20');
INSERT INTO PERSONNEL VALUES('P2303', 'Mike Green', '2016-11-10', NULL);
INSERT INTO PERSONNEL VALUES('P2309', 'Alice Grey', '2017-01-10', NULL);
INSERT INTO PERSONNEL VALUES('P2400', 'Joe Brown', '2017-01-10', NULL);
INSERT INTO PERSONNEL VALUES('P0000', 'Bob Bureaucrat', NULL, NULL);
INSERT INTO PERSONNEL VALUES('P0101', 'Mary Manager', NULL, NULL);
INSERT INTO PERSONNEL VALUES('P0102', 'Fred Foreman', NULL, NULL);
INSERT INTO PERSONNEL VALUES('P0103', 'Sam Supervisor', NULL, NULL);

PrManagement
INSERT INTO PRMANAGEMENT VALUES('P0101',NULL);
INSERT INTO PRMANAGEMENT VALUES('P0102',NULL);
INSERT INTO PRMANAGEMENT VALUES('P0103',NULL);

PrMaintenance
```

```
INSERT INTO PRMAINTENANCE VALUES('P2108');
INSERT INTO PRMAINTENANCE VALUES('P2109');
INSERT INTO PRMAINTENANCE VALUES('P2202');
INSERT INTO PRMAINTENANCE VALUES('P2303');

PrDirector
INSERT INTO PRDIRECTOR VALUES('P0000',NULL);

Certification
INSERT INTO CERTIFICATION VALUES('P2108', 'B', '2016-11-06');
INSERT INTO CERTIFICATION VALUES('P2108', 'E', '2016-11-10');
INSERT INTO CERTIFICATION VALUES('P2108', 'K', '2016-11-04');
INSERT INTO CERTIFICATION VALUES('P2109', 'B', '2016-11-06');
INSERT INTO CERTIFICATION VALUES('P2202', 'B', '2016-11-06');
INSERT INTO CERTIFICATION VALUES('P2303', 'B', '2016-11-25');
INSERT INTO CERTIFICATION VALUES('P2303', 'E', '2016-11-20');

Site
INSERT INTO SITE VALUES('AS', 'Airport Site', '44 Airport Dr.', 'P0102');
INSERT INTO SITE VALUES('DS', 'Downtown Site', '423 Main St.', 'P0101');
INSERT INTO SITE VALUES('SS', 'Suburban Site', '22 Center Circle', 'P0103');

Vehicle
INSERT INTO VEHICLE VALUES('B0001', '2016-08-01', 'B', 'SS', 'R',3);
INSERT INTO VEHICLE VALUES('B0011', '2016-10-01', 'B', 'DS', 'A', 3);
INSERT INTO VEHICLE VALUES('B0003', '2016-08-01', 'B', 'SS', 'A', 1);
INSERT INTO VEHICLE VALUES('T0002', '2016-08-01', 'B',  'SS', 'A', 4);
INSERT INTO VEHICLE VALUES('E0012', '2016-09-01', 'E', 'AS', 'R' , 4);
INSERT INTO VEHICLE VALUES('E0014', '2016-09-01', 'E', 'AS', 'R' , 3);
INSERT INTO VEHICLE VALUES('E0444', '2016-11-01', 'E',  'SS', 'A', 3);
INSERT INTO VEHICLE VALUES('E0523', '2016-12-01', 'E',  'SS', 'A', 0);
INSERT INTO VEHICLE VALUES('E0524', '2016-12-01', 'E',  'SS', 'M', 0);
INSERT INTO VEHICLE VALUES('B0010', NULL, NULL, NULL, NULL, NULL);


VhBike
INSERT INTO VHBIKE VALUES('B0001', 'R', 26, 8.00, 56.00);
INSERT INTO VHBIKE VALUES('B0011', 'R', 26, 8.00, 56.00);
INSERT INTO VHBIKE VALUES('B0003', 'R', 24, 6.00, 43.00);
INSERT INTO VHBIKE VALUES('T0002', 'T', 26, 10.00, 70.00);

VhElectricAuto
INSERT INTO VHELECTRICAUTO VALUES('E0012', 'EV10000234', 2592, 9.00, 0.40);
INSERT INTO VHELECTRICAUTO VALUES('E0014', 'EV10000235', 3505, 9.00, 0.40);
INSERT INTO VHELECTRICAUTO VALUES('E0444', 'EV10000500', 1200, 10.00, 0.40);
INSERT INTO VHELECTRICAUTO VALUES('E0523', 'EV10000600', 500, 11.00, 0.50);
INSERT INTO VHELECTRICAUTO VALUES('E0524', 'EV10000234', 10, 11.00, 0.50);


Maintenance
INSERT INTO MAINTENANCE VALUES('P2108', 'E0012', '2017-01-10', '2017-01-10',
'Broken wiper');
INSERT INTO MAINTENANCE VALUES('P2109', 'B0003', '2017-01-11', '2017-01-11',
'Flat tire');
INSERT INTO MAINTENANCE VALUES('P2108', 'B0001', '2017-11-10', '2017-11-10',
'Broken Chain');
INSERT INTO MAINTENANCE VALUES('P2109', 'B0003', '2017-11-11', '2017-11-11',
'Flat tire');
INSERT INTO MAINTENANCE VALUES('P2202', 'B0010', '2017-01-20', '2017-01-20',
'New wheel');
```

```
INSERT INTO MAINTENANCE VALUES('P2303', 'E0524', '2017-12-20', '2017-12-21',
'Initial checks');
```

Rent

```
INSERT INTO RENT VALUES(1, 'B0001', '2016-12-10 10:30', '2016-12-10 12:30');
INSERT INTO RENT VALUES(1, 'B0011', '2016-12-16 09:30', '2016-12-16 12:30');
INSERT INTO RENT VALUES(3, 'B0001', '2016-12-11 10:00', '2016-12-11 16:00');
INSERT INTO RENT VALUES(4, 'B0001', '2016-12-16 10:30', '2016-12-16 18:30');
INSERT INTO RENT VALUES(8, 'B0003', '2016-12-16 09:30', '2016-12-16 12:30');
INSERT INTO RENT VALUES(8, 'T0002', '2016-12-19 10:30', '2016-12-19 12:30');
INSERT INTO RENT VALUES(9, 'B0011', '2016-12-10 10:30', '2016-12-10 12:30');
INSERT INTO RENT VALUES(1, 'B0011', '2016-12-16 13:30', '2016-12-16 16:30');

INSERT INTO RENT VALUES(1, 'E0012', '2016-11-26 09:30', '2016-11-26 18:30');
INSERT INTO RENT VALUES(2, 'E0012', '2016-11-27 10:30', '2016-11-27 18:30');
INSERT INTO RENT VALUES(2, 'E0014', '2016-11-26 10:30', '2016-11-26 14:30');
INSERT INTO RENT VALUES(4, 'E0444', '2016-11-29 09:30', '2016-11-30 09:30');
INSERT INTO RENT VALUES(8, 'E0012', '2016-10-14 09:30', '2016-10-15 13:30');
INSERT INTO RENT VALUES(8, 'E0523', '2016-12-26 09:30', '2017-01-02 18:30');
INSERT INTO RENT VALUES(9, 'E0524', '2016-12-29 06:00', '2016-12-31 13:00');
INSERT INTO RENT VALUES(2, 'E0012', '2016-12-27 10:30', '2016-12-27 18:30');
```