star Motors Inc

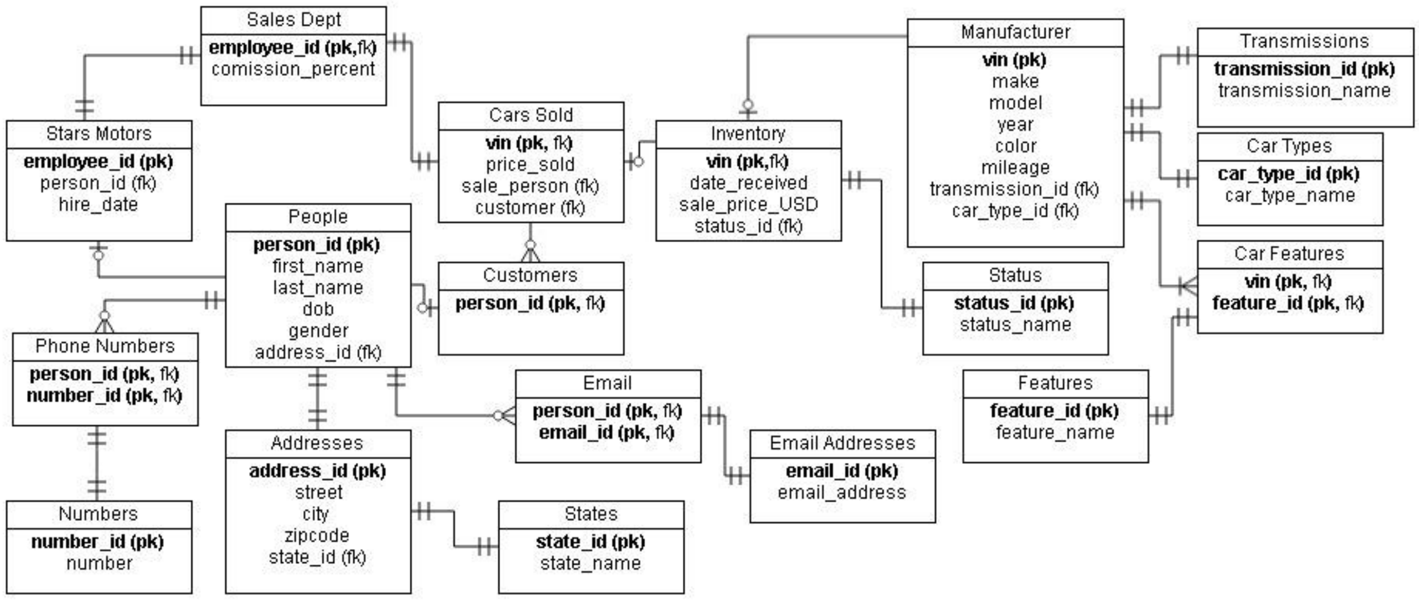
Executive Summary

In this document we will cover database overview for a popular car dealership. The document focuses its design and implementation for the Star Motors Inc. Some of the important users for this database schema include sales department of Star Motors', and of course customers. The design for Star Motors includes how schemas are created and discusses their functional dependencies and it also includes schemas which are necessary to sell a car. For example, the *cars_sold* table gives us information on the purchase of the car such as the sales person who sold the car and the commission the sales person received. To see how this database works in a real world environment some example queries are fetched from this database towards the end of the document. This includes reports such as top three employees with highest commission earned and customers who bought the same car such as Toyota. The commission earned for any sales person is never stored in this database because if it was stored, then it would become a part of a history and it is actually not needed to store because it can be calculated and same reason goes for the age of a person in people table because the age also can be calculated. Views are also discussed and it was created to calculate complex queries such as the sales person who sold the most cars. This database also provides sample data to its viewers which can help clients get a view of the schema structure. Although it is not limited to such finite number of schemas listed in this database, however the structure of this database makes it possible to implement more schemas as needed. I propose that the Star Motors Inc database is in third normal form because there are no partial dependencies and all non key attributes are dependent on the keys nothing but the whole key.

Table of Contents

Entity Relational Diagram

Key:
Pk = Primary key
Fk = Foreign Key



Transmissions Table

There are many cars in this world running on different transmissions. This table ensures that some of the popular transmissions are available at Star Motors Inc.

Sample Data →

```
create table transmissions(
	transmission_id char(3) not null unique,
	transmission_name text,
	primary key (transmission_id)
);
```

transmission_id	transmission_name
AUT	Automatic
MNU	Manual
CVT	Continuous Variable Transmission
SAT	Semi Automatic Transmission
TTG	Tip Tronic Gearbox

Functional Dependencies
Transmission : (transmission_id) → transmission_name

Car Types Table

Many cars are available in different shapes and sizes. This table makes it possible for our cars to support all types of cars. This table also helps to accommodate customer needs.

```
create table car_types(
	car_type_id char(3) not null unique,
	car_type_name text not null,
	primary key (car_type_id)
);
```

Sample Data →

car_type_id	car_type_name
CPE	Coupe
SDE	Sedan
SPT	Sports
VAN	Mini-Van
JPE	Jeep
TRK	Truck
SUV	SUV

Functional Dependencies
car_types : (car_type_id) → car_type_name

Manufacturer Table

List of popular cars are maintained in this table. There is no optional participation on transmission_id and car_type_id because we want to make sure that each car has a transmission and belongs to a some type.

```
create table manufacturer(
        vin char (12) not null unique,
        make text not null,
        model text not null,
        year int not null,
        color text not null,
        mielage int not null,
        transmission_id char(3) not null references transmissions (transmission_id),
        car_type_id char(3) not null references car_types (car_type_id),
        primary key (vin)
);
```

Functional Dependencies

Manufacturer : (vin) → make, model year, color, mileage, transmission_id, car_type_id

Sample Data

vin	make	model	year	color	Mileage	transmission_id	car_type_id
OP93SA123123	Ford	Mustang	2008	Black	2533	MNU	SPT
TK389D932443	Honda	Accord	2013	Grey	12556	AUT	AUT
KL43LP084J33	BMW	M6	2009	Grey	3201	MNU	SPT
MN33NMN39K44	Toyota	Camry	2007	Silver	25353	AUT	SDE
KO343UI804J4	Toyota	Corolla	2010	Silver	15033	AUT	VAN
NB33433FGH3J	Honda	CRV	2006	Black	125002	AUT	SUV
JPHO89L3434N	Jeep	Cherokee	2012	Red	56902	AUT	JPE
JN3429HH3N90	Acura	TL	2007	Silver	12560	AUT	SDE
PL3599HIK2J9	Audi	R8	2011	Red	23566	MNU	SPT
GR3430GKL309	Honda	Civic-Si	2009	Blue	89320	MNU	SDE
JBNGKOI333L3	Toyota	Camry	2008	Silver	98566	AUT	SDE
H23NP992092M	Infiniti	G-35	2010	Black	22568	AUT	CPE
YHFNM7782N4G	Ford	Explorer	2011	Grey	65992	AUT	SUV
JNFUO67296NJ	Hyundai	Sonata	2012	Red	56665	SAT	SDE
POIRETGJ49H3	BMW	X6	2012	Black	10111	TTG	SUV
4IHTUNG84IRN	Mercedes-Benz	GL-450	2009	White	25665	TTG	SUV
CJN72922BU52	Mazda	RX-8	2007	Grey	25665	AUT	CPE
IBDF3IBF498J	Nissan	Maxima	2008	Black	45556	AUT	SDE
KSBFD334LJN9	Volks-Wagon	Jetta	2009	Red	55634	AUT	SDE

Features Table

At Star Motors the company is driven to make its customers happy. To make this possible the company offers many features for every car. We also wanted to make sure that every car that we sell should have many feature accommodated to it.

Sample Data →

```
create table features(
        feature_id char(3) not null unique,
        feature_name not null,
        primary key (fid)
);
```

feature_id	feature_name
PRL	Power Lock and Windows
TCS	Traction Control System
HTM	Heated Mirrors
NVS	Navigation System
BLU	Bluetooth
CRU	Cruise Control
SPK	Sports Package
HTS	Heated Seats
DVD	DVD Video System
SNF	Sunroof
LTS	Leather Seats
4WD	4 Wheel Drive
AWD	All Wheel Drive
FWD	Front Wheel Drive
RWD	Rear Wheel Drive
MBF	Monroof
ABS	Anti-Lock Brake System
RKE	Remote Keyless Entry

Functional Dependencies

Features : (feature_id) → feature_name

Car Features Table

```
create table car_features(
        vin char(12) references manufacturer (vin),
        feature_id char(3) not null references features (feature_id),
        primary key (vin,feature_id)
);
```

Sample Data

vin	feature_id
OP93SA123123	PRL
OP93SA123123	HTM
OP93SA123123	SPK
OP93SA123123	HTS
OP93SA123123	RWD
OP93SA123123	LTS
OP93SA123123	TCS
TK389D932443	TCS
TK389D932443	PRL
TK389D932443	SNF
TK389D932443	RWD
TK389D932443	ABS
TK389D932443	RKE
KL43LP084J33	TCS
KL43LP084J33	HTM
KL43LP084J33	MNF
KL43LP084J33	SNF
KL43LP084J33	RKE
KL43LP084J33	ABS
MN33NMN39K44	ABS
MN33NMN39K44	RKE
MN33NMN39K44	MNF
MN33NMN39K44	CRU
MN33NMN39K44	RWD
MN33NMN39K44	DVD
MN33NMN39K44	BLU
MN33NMN39K44	HTM
MN33NMN39K44	NVS

Functional Dependencies

Car Features : (vin,feature_id) →

Status Table

Using this table provides us with a better way to serve our customers. This table offers a status variable through which our sales department can deal with customers in a friendly way. For example let's say a customer is interested in buying a car but cannot afford at the moment but wanted to put a down payment (which is refundable if not purchased because we would like to keep our customers happy in every way) for a future payment plan, this variable can be used at such times.

Sample Data

status_id	status_name
1	For-Sale
2	On-Hold
3	Sold

```
create table status(
        status_id int not null unique,
        status_name text not null,
        primary key (status_id)
);
```

Functional Dependencies

Status : (status_id) → status_name

Inventory Table – The inventory of our company

```
create table inventory(
vin char(12) references manufacturer (vin),
date_received date not null,
sale_price decimal not null check (sale_price > 0),
status_id int not null references status (status_id) check (status_id between 1 and 2),
primary key (vin)
);
```

Sample Data

Vin	date_received	sale_price_USD	status_id
OP93SA123123	2012-12-09	21235.68	1
TK389D932443	2012-09-05	15665.89	1
KL43LP084J33	2013-01-08	56995.59	1
MN33NMN39K44	2013-04-18	12555.84	1
KO343UI804J4	2013-09-24	14665.85	1
NB33433FGH3J	2012-05-27	18356.59	2
JPHO89L3434N	2012-10-28	19688.81	1
PL3599HIK2J9	2012-06-28	76991.99	1
GR3430GKL309	2012-09-07	17995.48	2
JBNGKOI333L3	2012-01-18	11665.99	1
H23NP992092M	2012-04-25	26966.45	1
YHFNM7782N4G	2012-09-19	22339.94	1
JNFUO67296NJ	2012-05-07	20661.89	1
CJN72922BU52	2012-12-29	16485.58	2
IBDF3IBF498J	2012-01-16	18995.98	1
KSBFD334LJN9	2012-02-04	14665.68	1
POIRETGJ49H3	2012-01-09	86664.97	1
4IHTUNG84IRN	2012-07-17	46698.91	1
SIDHBVK38323	2012-10-17	22386.84	2
SIDHBVK38323	2102-08-18	22694.88	1
JN3429HH3N90	2012-05-29	24994.98	1
4ITH984HUN90	2012-05-18	32599.84	2
SDKJFN39398N	2012-04-25	98993.54	1

Functional Dependencies

Inventory : (vin) → date_received, sale_price, status_id

States Table

```
create table states(
        state_id char(2) not null unique,
        state_name text not null,
        primary key (state_id)
);
```

Functional Dependencies
States : (state_id) → state_name

Sample Data

State_id	State_name
WA	Washington
NY	New York
CA	California
AZ	Arizona
TX	Texas
FL	Florida
NC	North Carolina
MI	Michigan
KS	Kansas
UT	Utah
MD	Maryland
OH	Ohio
PA	Pennsylvania
NJ	New Jersey
CT	Connecticut
SC	South Carolina
NM	New Mexico
AL	Alabama
HI	Hawaii
IN	Indiana
ME	Maine
NV	Nevada
WI	Wisconsin

Addresses Table

```
create table addresses(
            address_id int not null unique,
            street text not null,
            city text not null,
            zipcode int not null,
            state_id char(2) not null references states (state_id),
            primary key (address_id)
);
```

Sample Data

address_id	street	city	zipcode	state_id
1	Cooper Street	Rochester	22901	WA
2	33 Dare Lane	Rome	11344	CA
3	993 Eagle Street	Water-Town	44909	AZ
4	34 Estate Road	Ocean City	4990	MD
5	30 Fair Avenue	James-Town	39009	NC
6	59 Front Lane	Seattle	85990	FL
7	32 Lanko Hills	Lanko Hills	43223	WA
8	234 Broadway	New York City	48923	CA
9	124 Sky Avenue	Middletown	8442	AZ
10	320 Pizza Street	Mind Game	85990	NJ
11	848 Bitwise lane	Beatles	32733	WI
12	24977 Fruit lane	Fruity	37822	NV
13	372 Mango Avenue	Fruity	32733	HI
14	372 Mango Avenue	Mongo	55645	NM
15	33 Cotton Road	Cotton	238	AL
16	32 High Street	High	88986	PA
17	273 Oil Avenue	Oil Spill	55995	MI
18	948 Front Lane	Big-Town	78952	UT
19	129 Onion Street	Onion Town	23189	NY
20	828 Park Avenue	New York City	10001	NY

Functional Dependencies
Addresses : (address_id) → street, city, zipcode, state_id

Number Table

```
create table numbers(
            number_id int not null unique,
            number char(12) not null,
            primary key (number_id)
);
```

Sample Data

number_id	number
849	859-250-1505
432	464-765-5356
656	678-235-5443
850	800-155-4005
904	792-915-0051
815	816-842-7721
64	813-888-0232
223	389-994-9768
318	348-625-6556
597	846-816-6509
965	943-648-8866
983	935-235-8848
655	815-515-6715
889	954-655-0312
964	841-546-5588
912	921-659-0943
624	415-326-8635
332	518-566-5629
118	290-556-6034
548	028-665-9452
888	059-495-8129
608	220-569-7892
951	209-795-9433
514	915-260-3264
218	859-029-7520

Functional Dependencies

Numbers : (number_id) → number

Email Addresses

```
create table email_addresses(
        email_id int not null unique,
        emdil_address text not null,
        primary key (email_id)
);
```

Functional Dependencies
Email_Addresses : (email__id) → email_address

Sample Data

email_id	email_address
213	flyone@gmail.com
439	pole30@hotmail.com
399	opengl@yahoo.com
390	eaglemaster@gmail.com
585	moneyroll@gmail.com
959	petm20@yahoo.com
915	myhome24@yahoo.com
659	gamer10@yahoo.com
89	lanes40@yahoo.com
985	water390@gmail.com
958	drive39@gmail.com
995	taller14@hotmail.com
988	buzzbee90@gmail.com
381	oilmail42@gmail.com
234	skinnyjeans34@gmail.com
753	realcorn338@gmail.com
266	kolem_434@yahoo.com
512	earlybird98@gmail.com
558	heman4u3@hotmail.com
705	lovelygreen24@gmail.com
658	gummybear54@gmail.com

People Table

```
create table people(
            person_id int not null unique,
            first_name text not null,
            last_name text not null,
            dob date not null,
            gender char (1) not null check (gender = 'M' or gender = 'F'),
            address_id int not null references addresses (address_id),
            primary key (person_id)
);
```

Sample Data

Person_id	First_name	Last_name	DOB	Gender	Address_id
1	Susan	Smith	1986-07-15	F	1
2	John	Mayes	1958-04-25	M	2
3	Jhonny	Flores	1973-01-09	M	3
4	Jason	Moore	1984-08-08	M	4
5	Ashley	Adams	1985-02-19	F	5
6	Stephanie	Malcom	1976-06-21	F	6
7	Mike	Mildanado	1962-12-21	M	7
8	Tom	Jane	1976-03-25	M	8
9	Nick	Mayes	1956-09-26	M	9
10	Daniel	Craig	1955-11-22	M	10
11	Victor	Hayes	1981-07-02	M	11
12	Jose	Cardo	1989-12-22	M	12
13	Mark	Russel	1977-11-25	M	13
14	Jerry	Mills	1988-02-19	M	14
15	Jennifer	Mills	1991-11-28	F	15
16	Amanda	Styles	1992-06-20	F	16
17	Steven	Stames	1956-01-22	M	17
18	James	Flores	1988-03-22	M	18
19	Joann	Jiane	1985-09-09	F	19
20	Jason	King	1993-12-21	M	20

Functional Dependencies

People : (person_id) → first_name, last_name, dob, gender, address_id

Phone Numbers

In today's world people can be contacted in many ways. This table gives us an opportunity to support this new trend. It allows people in our company to have multiple phone numbers.

```
create table phone_numbers(
        person_id int references people (person_id),
        number_id int references numbers (number_id),
        primary key (person_id, number_id)
);
```

Sample Data →

Functional Dependencies

Phone_Numbers : (person_id, number_id) →

Person_id	Number_id
1	849
1	432
2	656
2	850
3	904
3	815
3	64
4	223
5	318
6	597
7	965
7	983
8	655
9	889
10	964
11	321
11	912
12	624
13	332
14	118
15	558
15	548
16	888
16	945
17	608
18	951
19	968
19	514
20	218

Email Table

This table supports people to have multiple email addresses.

```
create table email(
            person_id int references people (person_id),
            email_id int references email_addresses (email_id),
            primary key (person_id, email_id)
);
```

Sample Data →

person_id	email_id
1	213
2	439
3	399
4	390
4	585
5	959
5	659
5	89
6	985
7	995
8	988
9	381
9	234
10	753
10	266
11	512
12	558
12	705
13	658
13	506
14	648
15	228
16	618
17	644
18	577
19	524
19	556
20	789
20	918

Functional Dependencies

Email : (person_id, email_id) →

Stars Morots Table

```
create table stars_motors(
            employee_id int not null unique,
            person_id int unique references people (person_id),
            hire_date date not null,
            primary key (employee_id)
);
```

Sample Data

Employee_id	Person_id	Hire_date
111	1	2013-10-08
222	2	2012-01-18
333	3	2012-09-14
444	4	2012-06-25
555	5	2012-07-23

Functional Dependencies

Stars Motors : (employee_id) → person_id, hire_date

Customers Table

Sample Data →

Person_id
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

```
create table customers(
            person_id int references people (person_id) unique,
            primary key (person_id)
);
```

Functional Dependencies

Customers : (person_id) →

Sales Dept Table

The company offers commission up to some percentage on the sold price. Using this type of strategy can make our company move forward. This is also one of the possible ways to generate revenues.

```
create table sales_dept(
        employee_id int references stars_motors (employee_id),
        commission_percent real not null check (commission_percent < 5.0),
        primary key (employee_id)
);
```

Sample Data

employee_id	commission_percent
111	2.5
222	2.2
333	3.5
444	3.65
555	4.01

Functional Dependencies

Sales_dept : (employee_id) → commission_percent

Cars Sold Table

```
create table cars_sold(
            vin char(12) references inventory (vin),
            price_sold decimal not null check (price_sold >0),
            date_sold date not null,
            sales_person int not null references sales_dept (employee_id),
            customer int not null references customers (person_id),
            primary key (vin)
);
```

Sample Data

vin	price_sold	date_sold	sales_person	customer
OP93SA123123	20459.55	2013-01-22	111	6
TK389D932443	15223.58	2012-10-12	111	6
KL43LP084J33	15223.58	2013-02-19	222	7
MN33NMN39K44	12388.99	2013-06-02	333	8
KO343UI804J4	13999.44	2013-10-29	444	9
JPHO89L3434N	19688.81	2012-12-19	555	10
PL3599HIK2J9	75998.99	2012-09-27	111	11
JBNGKOI333L3	11665.99	2012-09-27	111	12
H23NP992092M	26459.18	2012-09-27	222	13
YHFNM7782N4G	22294.08	2012-11-29	333	14
JNFUO67296NJ	20559.99	2013-01-04	444	15
IBDF3IBF498J	18897.99	2012-02-24	555	16
KSBFD334LJN9	14536.39	2012-03-19	555	17
POIRETGJ49H3	85699.99	2012-03-25	333	18
4IHTUNG84IRN	45993.97	2012-09-28	444	19
SKBFK38947U3	21598.79	2012-08-27	555	20
JN3429HH3N90	26558.89	2012-06-20	555	7
SDKJFN39398N	100829.64	2012-05-10	333	9

Functional Dependencies

Cars_Sold : (vin) → price_sold, date_sold, sales_person, customer

Views

This view helps us figure out the employee who has sold the most cars of all. It doesn’t matter in case of a tie, it pulls up both.

```
create view max_cars_sold
as
select person_id, first_name, last_name
from people
where person_id in(
select person_id
from employees
where employee_id in(
select employee_id
from sales_dept
where employee_id in(
select cs.sales_person
from cars_sold cs
group by cs.sales_person
having count (cs.sales_person) in(
select max("max_sold")
from (
select cs.sales_person, count (cs.sales_person) as "max_sold"
from
cars_sold cs, sales_dept sd
where cs.sales_person = sd.employee_id
group by cs.sales_person
order by count (cs.sales_person) desc) sub1))));
```

Query Result

persin_id	first_name	last_name
5	Ashley	Adams

Queres

Query1:
All customers who bought a Toyota

```
select c.person_id, p.first_name, p.last_name, m.make, m.model
from inventory i, manufacturer m, cars_sold cs, people p, customers c
where i.vin = m.vin
and m.make = 'Toyota'
and i.vin = cs.vin
and c.person_id = cs.customer
and c.person_id = p.person_id
```

Query Result

Person_id	First_name	Last_name	Make	Model
8	Tom	Jane	Toyota	Camry
9	Nick	Mayes	Toyota	Corolla
12	Jose	Cardo	Toyota	Camry

Query2:
Top three employees with who earned the highest commission.

```
select sales_person, max("Comission Earned")
from(
select  cs.sales_person,
sd.comission_percent / 100 * cs.price_sold as "Comission Earned"
from cars_sold cs, sales_dept sd
where cs.sales_person = sd.employee_id
order by "Comission Earned" desc) sub1
group by sales_person
order by max desc
limit 3
```

Query Result

sales_person	commission earned
333	3529.0374
111	1899.97475
444	1678.77994886327

Trigger

Create a trigger on cars sold table after insert. This trigger can make sure that the status of a vin being added to cars sold table is valid. The trigger should make sure that we only add those vin's to our cars sold table whose status has been set to "sold" in the inventory table.

```
Create function valid_status() returns trigger as $valid_status$
        Begin
                If cars_sold.vin = inventory.vin
                And inventory.status != 1
        Raise exception 'Car is not available for sale';
                End if
        End;
$valid_status$ language plpsql;

Create trigger valid_status after insert or update on cars_sold
For each row execute procedure valid_status();
```

Known Issues

Although this database model might able to support a short staff company, there are many things which could go wrong when the company grows in the future.

What would happen if the company decides to change its policies, such as what if the commission rate depends on the years employed? Many car dealerships offer financing to its customers, does this company offer financing?

- Implement a role through which an administrating team can maintain our database by allowing them access to all inserts, updates, alteration, deletes (not cascade).
- The current data model doesn't allow people in sales department to be customers. This raises a problem because a person working in sales department is not allowed to buy a car from his employer. This doesn't seem fair, and most of the companies might allow this option.
- To add more functionality to this schema, Star Motors could add financing schemas to assist customers who need finance assistance.

Future Enhancements

As the company grows there could be many things which could be added to this database to improve sales, to have a better database model, or to build a data model which can be easily adaptable to changes in the future.

- Have promotions sales around holiday seasons.
- Offer warranty on low mileage cars.
- Attract customers by offering free maintenance up to for a finite number of months
- Offer 0% APR to its customers
- New college graduates receive up to $2000 rebate towards their first brand new car depending on their major.

When does an employee get's a raise and does this company offers any benefits to employees?

- Implement a view which calculates total number of cars sold within a year and give that employee raise according to what he or she deserves.
- Every three year the person who sells the most cars gets a round trip air fare and hotel to Las Vegas.
- All employees get a 10% discount towards their first car if bought from Star Motors Inc and free maintenance for as long as that employee is employed.