Project One: Build a Relational Database

Stephen Robinson

October 24, 2019

Introduction

The purpose of this project was to create a simple relational database utilizing Oracle's SQL Developer and Data Modeler tools as well as various beginner level concepts from the course text *Databases Illuminated*. Per the assignment page on the course moodle, the project requirements are as follows:

1. Provide an explanation of my experience in completing this project.
2. Create and provide a data dictionary.
3. Develop and provide a logical data model in Bachman notation.
4. Develop and provide a relational model.
5. Generate and provide a DDL script for creating our database (with SET ECHO ON)
6. Provide sample SQL Inserts that were used to input data in our database.
7. Provide example SQL Selects to prove our database is functioning as intended.

Implementation

For this project I decided to utilize 'Project 3' from the potential project pool. This project seeks to develop a database to be used by a small-scale car dealership. The project consists of several tables which serve to provide a working product which accounts for many of the needs a car dealership might have. The tables are as follows:

1. **CarCustomer** - which stores information pertaining to the customer of the dealership
2. **CarSalesperson** - which stores information pertaining to the sales employees of a dealership
3. **CarNewCar** - which stores information pertaining to the new car inventory of a dealership
4. **CarSale** - which stores information pertaining to the sales made by employees of the dealership
5. **CarManifest** - which stores the primary keys of both the CarSale and CarNewCar tables in order to facilitate their M : N relationship.
6. **CarRegistration** - which stores information relating to registration of a sold car.
7. **CarWarranty** - which stores information describing the warranty purchased for a sold car.
8. **CarSaleSurvey** - which stores information relating to customer satisfaction.

Each table contains the appropriate primary and foreign keys and the totality of this project satisfies all of the technical requirements outlined in the introduction.

Mastery of the basic design concepts as required by this project are all demonstrated within my data dictionary, logical data model, relational model, and resultant DDL script which are all in the appendix at the end of this document.

Developing the data dictionary was perhaps the most daunting process I experience in this project. Initially, I found it difficult to not design my database on the fly as I was capturing the relevant information from the 'Project 3' document. Finding relevant phrases and business needs was not necessarily difficult but certainly time-consuming. The resultant data dictionary represents an initial assessment of what information might be useful in my ideal database. As I began to design the logical model of my project, it became apparent that in order to implement this data dictionary completely, I would have to greatly exceed the

necessary project requirements (I know of at least 18 tables that would have been necessary). In order to keep this project at a reasonable length for implementation and grading, I elected to create a truncated version of my ideal product. I have, however, included the full data dictionary as a testament to the trimming I felt necessary as I completed my logical model.

Using the SQL Data Modeler tool from Oracle was a dream. The interface was easy to grasp and navigate given the in-class tutorials. Having the ability to go from my completed logical data model to the relational model and then to a DDL script in a matter of just a few clicks was nothing short of amazing.

In working with the database itself, again, the in-class tutorials were extremely beneficial. Having completed CSCI-344, I had some experience with the basic SQL statements required but without guidance in-class, navigating the SQL Developer tool would have been a daunting task. Overall the experience with this tool was also positive. A few sample inserts from my database are included below first with using a full insert statement including all the column names (CarCustomer table):

```
/*CUSTOMERS&*/

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (01, 'Gerald', 'Arbor', '1 Glendwood Ave.', 'Carmine', 'IL', 47824, 78923942, 423, 'Facebook');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (02, 'Gabby', 'Baxter', '5 Glendwood Ave.', 'Carmine', 'IL', 47824, 78923943, 423, 'Facebook');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (03, 'April', 'Carlyle', '10 Simple Ave.', 'Carmine', 'IL', 47824, 78923944, 423, 'Twitter');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (04, 'Stanley', 'Dabbuk', '15 Simple Ave.', 'Carmine', 'IL', 47824, 78923542, 423, 'Twitter');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (05, 'Kim', 'Edwards', '17 Graveyard Ave.', 'Carmine', 'IL', 47824, 78923972, 423, 'Craigslist');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (06, 'Tan', 'Fallon', '19 Glendwood Ave.', 'Carmine', 'IL', 47824, 78923882, 423, 'Craigslist');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (07, 'Sandford', 'Gentry', '56 Graveyard Ave.', 'Carmine', 'IL', 47824, 78921942, 423, 'Billboard');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (08, 'Ruth', 'Hyde', '16 Graveyard Ave.', 'Carmine', 'IL', 47824, 78923977, 423, 'Billboard');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (09, 'Bernard', 'Ingle', '13 Tupelo Ave.', 'Carmine', 'IL', 47824, 78923992, 423, 'Facebook');

INSERT INTO CARCUSTOMER (custid, firstname, lastname, street, city, state, zip, phonenumber, areacode, adseen)
VALUES (10, 'Stephen', 'Jackson', '11 Tupelo Ave.', 'Carmine', 'IL', 47824, 78923947, 423, 'Newspaper');
```

And another insert using a truncated format which omits the column names (CarNewCar table):

```
/*INVENTORY*/

INSERT INTO CARNEWCAR VALUES('10751DJBF938', 'FORD', 'MUSTANG', '16-JAN-2019', 55000.00, 'MI', 2, 3500, '21-JAN-2019', 38);
INSERT INTO CARNEWCAR VALUES('10751ADBF938', 'FORD', 'MUSTANG', '16-JAN-2019', 55000.00, 'MI', 2, 3500, '21-JAN-2019', 27);
INSERT INTO CARNEWCAR VALUES('1078ABADV938', 'FORD', 'MUSTANG', '16-JAN-2019', 55000.00, 'MI', 2, 3500, '21-JAN-2019', 65);
INSERT INTO CARNEWCAR VALUES('10751D88BA88', 'FORD', 'F150', '16-FEB-2019', 95000.00, 'MI', 4, 6500, '21-FEB-2019', 32);
INSERT INTO CARNEWCAR VALUES('10751DJBF938', 'FORD', 'F150', '16-FEB-2019', 95000.00, 'MI', 4, 6500, '21-FEB-2019', 39);
INSERT INTO CARNEWCAR VALUES('10778BVSF938', 'FORD', 'F150', '16-FEB-2019', 95000.00, 'MI', 4, 6500, '21-FEB-2019', 11);
INSERT INTO CARNEWCAR VALUES('10789ABBF938', 'FORD', 'F150', '16-MAR-2019', 95000.00, 'MI', 4, 6500, '21-MAR-2019', 59);
INSERT INTO CARNEWCAR VALUES('1075QIBV9SCC', 'FORD', 'EXPLORER', '16-AUG-2019', 75000.00, 'MI', 4, 4500, '21-AUG-2019', 16);
INSERT INTO CARNEWCAR VALUES('1075649FJKS7', 'FORD', 'EXPLORER', '16-AUG-2019', 75000.00, 'MI', 4, 4500, '21-AUG-2019', 17);
INSERT INTO CARNEWCAR VALUES('107889SBV938', 'FORD', 'EXPLORER', '16-AUG-2019', 75000.00, 'MI', 4, 4500, '21-AUG-2019', 13);
```

Errors and Fixes

Though this project was vastly more complex than my previous foray into SQL in CSCI-344, I did not feel that issues encountered were any more or less difficult to deal with than before.

The vast majority of issues encountered in this project were related to an incorrect relational model. Fixing these issues required me to drop all of my tables, delete my relational model from the SQL Data Modeler tool, make the necessary changes in my logical model and then generate everything all over again. Luckily I only had to do this a couple of times and the tools made each time a fairly painless experience.

Using SELECT statements to check for errors and ensure that my table was behaving as it should proved to be a huge help in determining the validity of my design. Below is a SELECT * of my CarCustomer table:

```
/* SELECT ALL */

SELECT * FROM CARCUSTOMER;
```

Script Output

Task completed in 0.167 seconds

CUSTID	FIRSTNAME	LASTNAME	STREET	CITY	ST	ZIP	PHONENUMBER	AREACODE	ADSEEN
1	Gerald	Arbor	1 Glendwood Ave.	Carmine	IL	47824	78923942	423	Facebook
2	Gabby	Baxter	5 Glendwood Ave.	Carmine	IL	47824	78923943	423	Facebook
3	April	Carlyle	10 Simple Ave.	Carmine	IL	47824	78923944	423	Twitter
4	Stanley	Dabbuk	15 Simple Ave.	Carmine	IL	47824	78923542	423	Twitter
5	Kim	Edwards	17 Graveyard Ave.	Carmine	IL	47824	78923972	423	Craigslist
6	Tan	Fallon	19 Glendwood Ave.	Carmine	IL	47824	78923882	423	Craigslist
7	Sandford	Gentry	56 Graveyard Ave.	Carmine	IL	47824	78921942	423	Billboard
8	Ruth	Hyde	16 Graveyard Ave.	Carmine	IL	47824	78923977	423	Billboard
9	Bernard	Ingle	13 Tupelo Ave.	Carmine	IL	47824	78923992	423	Facebook
10	Stephen	Jackson	11 Tupelo Ave.	Carmine	IL	47824	78923947	423	Newspaper

10 rows selected.

This example is certainly functional, but probably fairly boring to a business owner. I utilized SELECT twice more to generate a couple of reports that I thought might be interesting. The first is a report for the sales team which capture customer names and contact information for customers who elected for the least expensive warranty. Information like this could be given to the sales team during downtime to help drive additional revenue through 'upselling' by having the salesperson call and reiterate just how much their car actually costs and the benefits of a better warranty:

```
SELECT lastName, firstName, areaCode, phoneNumber, carsale.saleprice
FROM CARCUSTOMER, CARSALE where custID = CARCUSTOMER_CUSTID AND SALEPRICE > 50000.00 AND carsale.carwarranty_warrantytype = 'Manufacturer';
```

Script Output

Task completed in 0.128 seconds

LASTNAME	FIRSTNAME	AREACODE	PHONENUMBER	SALEPRICE
Fallon	Tan	423	78923882	60000
Ingle	Bernard	423	78923992	90000

The second report is intended to be used by the owner to followup with customers who had a poor experience with their salesperson.

```
SELECT lastName, firstName, areaCode, phoneNumber
FROM CARCUSTOMER, CARSALE, CARSALESURVEY WHERE custID = carsale.carcustomer_custid
AND carsale.invoiceno = carsalesurvey.CARSALE_invoiceno
AND salespersonrating < 7;
```

Script Output

Task completed in 0.122 seconds

LASTNAME	FIRSTNAME	AREACODE	PHONENUMBER
Fallon	Tan	423	78923882
Baxter	Gabby	423	78923943
Hyde	Ruth	423	78923977

Conclusion

As the first attempt at developing a larger scale project using SQL and these Oracle tools, I am satisfied with the result. Though the scope of this project is narrow, I don't feel as though the quality of the work or the learning experience suffered as a result. Admittedly, I still found myself throughout the process discovering new techniques that I would like to implement in personal projects. I am certain that my continued efforts and curiosity will grow my fluency with SQL.

Appendix

Initial Data Dictionary:

Customer Records

custFirstName - The customers first name

custLastName - The customers last name

custStreet - Street of the customers physical address

custCity - City of the customers physical address

custState - State of the customers physical address

custZip - Zip Code of the customers physical address

custAreaCode - Area Code of the customers primary phone number

custPhoneNumber - Customers primary phone number

custLicenseNum - Customers drivers license number

custAdSeen - Ad that brought the customer to the dealership

custSaleMade - Denotes whether a customer has made a purchase

Car Records

carVIN - Vehicle ID number for a car

carMake - Make of a car

carModel - Model of a car

carPrice - List price of a car

carManufactureDate - Date the car was manufactured

carColor - Color of a car

carDoors - The number of doors of a car

carWeight - Weight of a car

carManufacturePlace - Place in which the car was manufactured

carOptions - List of options specific to a particular car

carMileage - Current mileage of a car Salesperson Records

salespersonFirstName - First name of a salesperson

salespersonLastName - Last name of a salesperson

Sale Records

saleListPrice - The sticker price from the manufacturer for a car being sold

saleFinalPrice - Price at which the car is being sold after negotiation

saleTax - Tax to be collected for sale of the car

saleLicenseFee - License fee for the car being sold

saleVIN - VIN of the car being sold

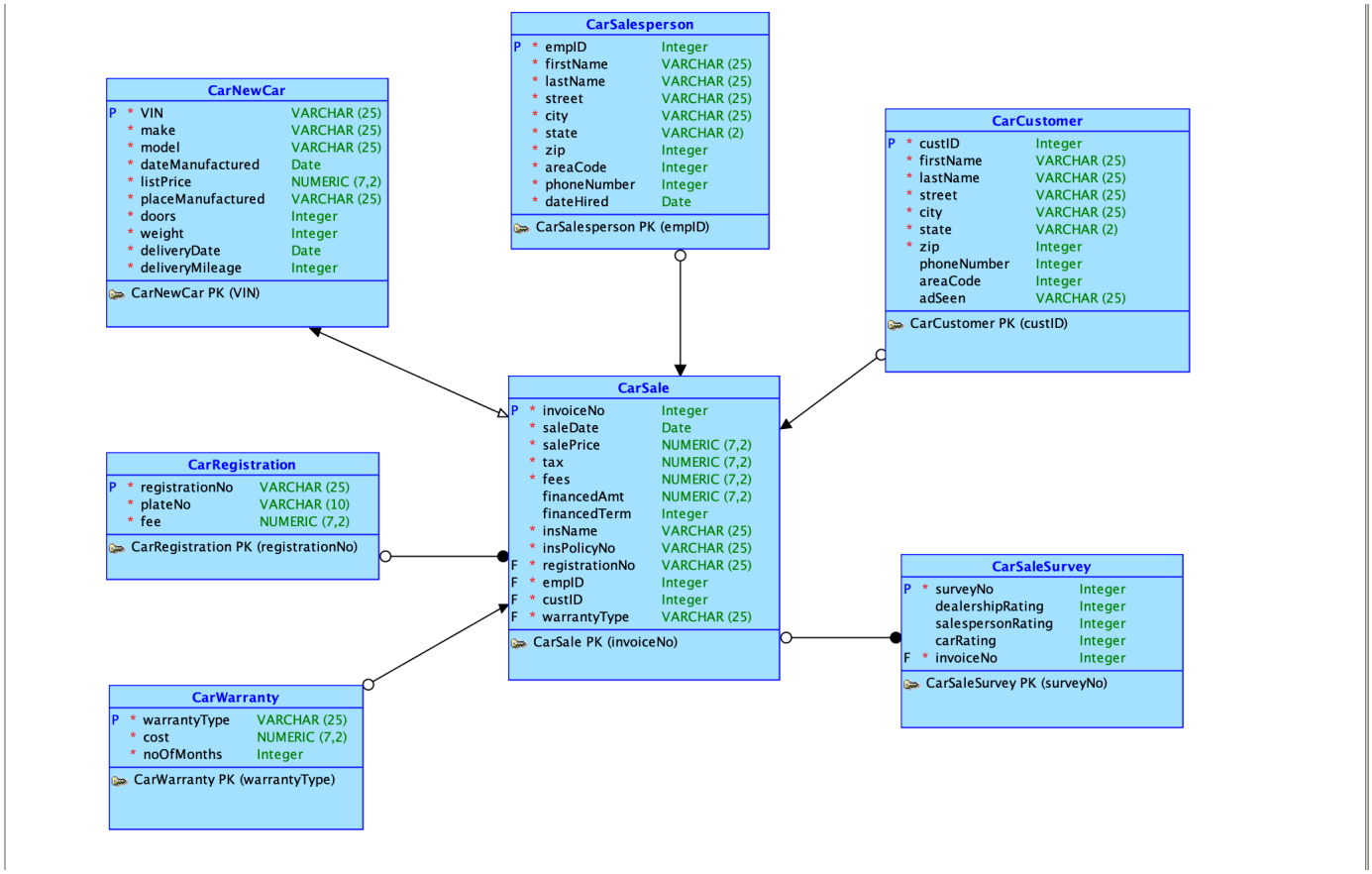
saleCarMileage - current mileage of the car being sold at the time of the sale

saleSalesperson - Salesperson who closed the sale

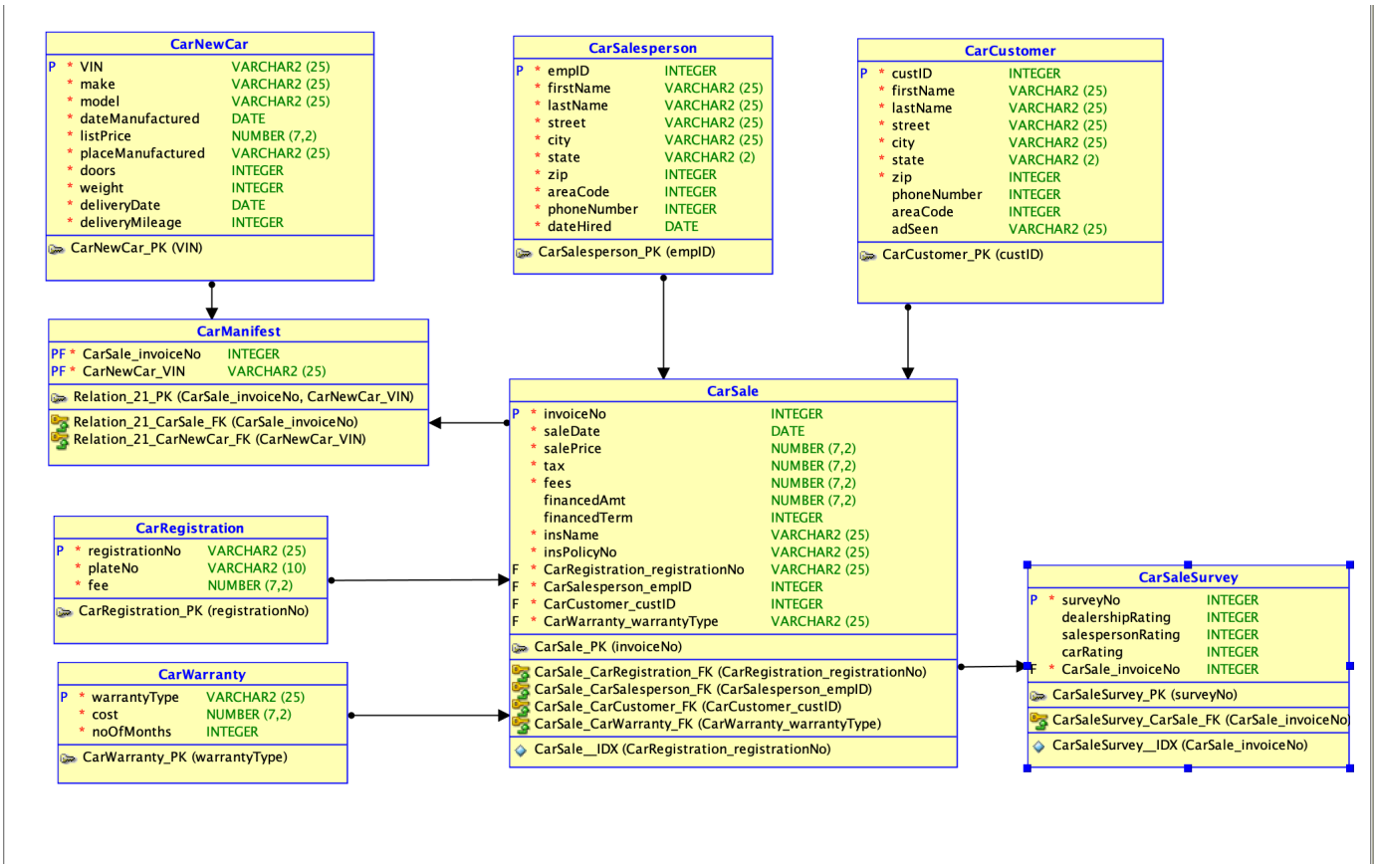
saleWarrantyInfo - Warranty information of the vehicle being sold

saleFinancingTerm - Number of months at which the vehicle is being financed

Logical Data Model:



Relational Model:



DDL:

```
-- Generated by Oracle SQL Developer Data Modeler 19.2.0.182.1216
--   at:        2019-10-24 19:04:00 EDT
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g

SET ECHO ON;

CREATE TABLE carcustomer (
    custid         INTEGER NOT NULL,
    firstname      VARCHAR2(25) NOT NULL,
    lastname       VARCHAR2(25) NOT NULL,
    street         VARCHAR2(25) NOT NULL,
    city           VARCHAR2(25) NOT NULL,
    state          VARCHAR2(2) NOT NULL,
    zip            INTEGER NOT NULL,
    phonenumber    INTEGER,
    areacode       INTEGER,
    adseen         VARCHAR2(25)
);

ALTER TABLE carcustomer ADD CONSTRAINT carcustomer_pk PRIMARY KEY ( custid );

CREATE TABLE carmanifest (
    carsale_invoiceno   INTEGER NOT NULL,
    carnewcar_vin       VARCHAR2(25) NOT NULL
);

ALTER TABLE carmanifest ADD CONSTRAINT relation_21_pk PRIMARY KEY ( carsale_invoiceno,
                                                                    carnewcar_vin );

CREATE TABLE carnewcar (
    vin                 VARCHAR2(25) NOT NULL,
    make                VARCHAR2(25) NOT NULL,
    model               VARCHAR2(25) NOT NULL,
    datemanufactured    DATE NOT NULL,
    listprice           NUMBER(7, 2) NOT NULL,
    placemanufactured   VARCHAR2(25) NOT NULL,
    doors               INTEGER NOT NULL,
    weight              INTEGER NOT NULL,
    deliverydate        DATE NOT NULL,
    deliverymileage     INTEGER NOT NULL
);
```

```
ALTER TABLE carregistration ADD CONSTRAINT carregistration_pk PRIMARY KEY ( registrationno );

CREATE TABLE carsale (
    invoiceno                       INTEGER NOT NULL,
    saledate                        DATE NOT NULL,
    saleprice                       NUMBER(7, 2) NOT NULL,
    tax                             NUMBER(7, 2) NOT NULL,
    fees                            NUMBER(7, 2) NOT NULL,
    financedamt                     NUMBER(7, 2),
    financedterm                    INTEGER,
    insname                         VARCHAR2(25) NOT NULL,
    inspolicyno                     VARCHAR2(25) NOT NULL,
    carregistration_registrationno  VARCHAR2(25) NOT NULL,
    carsalesperson_empid            INTEGER NOT NULL,
    carcustomer_custid              INTEGER NOT NULL,
    carwarranty_warrantytype        VARCHAR2(25) NOT NULL
);

CREATE UNIQUE INDEX carsale__idx ON
    carsale (
        carregistration_registrationno
    ASC );

ALTER TABLE carsale ADD CONSTRAINT carsale_pk PRIMARY KEY ( invoiceno );

CREATE TABLE carsalesperson (
    empid         INTEGER NOT NULL,
    firstname     VARCHAR2(25) NOT NULL,
    lastname      VARCHAR2(25) NOT NULL,
    street        VARCHAR2(25) NOT NULL,
    city          VARCHAR2(25) NOT NULL,
    state         VARCHAR2(2) NOT NULL,
    zip           INTEGER NOT NULL,
    areacode      INTEGER NOT NULL,
    phonenumber   INTEGER NOT NULL,
    datehired     DATE NOT NULL
);

ALTER TABLE carsalesperson ADD CONSTRAINT carsalesperson_pk PRIMARY KEY ( empid );
```

```
CREATE TABLE carsalesurvey (
    surveyno            INTEGER NOT NULL,
    dealershiprating    INTEGER,
    salespersonrating   INTEGER,
    carrating           INTEGER,
    carsale_invoiceno   INTEGER NOT NULL
);

CREATE UNIQUE INDEX carsalesurvey__idx ON
    carsalesurvey (
        carsale_invoiceno
    ASC );

ALTER TABLE carsalesurvey ADD CONSTRAINT carsalesurvey_pk PRIMARY KEY ( surveyno );

CREATE TABLE carwarranty (
    warrantytype   VARCHAR2(25) NOT NULL,
    cost           NUMBER(7, 2) NOT NULL,
    noofmonths     INTEGER NOT NULL
);

ALTER TABLE carwarranty ADD CONSTRAINT carwarranty_pk PRIMARY KEY ( warrantytype );

ALTER TABLE carsale
    ADD CONSTRAINT carsale_carcustomer_fk FOREIGN KEY ( carcustomer_custid )
        REFERENCES carcustomer ( custid );

ALTER TABLE carsale
    ADD CONSTRAINT carsale_carregistration_fk FOREIGN KEY ( carregistration_registrationno )
        REFERENCES carregistration ( registrationno );

ALTER TABLE carsale
    ADD CONSTRAINT carsale_carsalesperson_fk FOREIGN KEY ( carsalesperson_empid )
        REFERENCES carsalesperson ( empid );

ALTER TABLE carsale
    ADD CONSTRAINT carsale_carwarranty_fk FOREIGN KEY ( carwarranty_warrantytype )
        REFERENCES carwarranty ( warrantytype );

ALTER TABLE carsalesurvey
    ADD CONSTRAINT carsalesurvey_carsale_fk FOREIGN KEY ( carsale_invoiceno )
        REFERENCES carsale ( invoiceno );
```

```
ALTER TABLE carmanifest
    ADD CONSTRAINT relation_21_carnewcar_fk FOREIGN KEY ( carnewcar_vin )
        REFERENCES carnewcar ( vin );

ALTER TABLE carmanifest
    ADD CONSTRAINT relation_21_carsale_fk FOREIGN KEY ( carsale_invoiceno )
        REFERENCES carsale ( invoiceno );

-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                             8
-- CREATE INDEX                             2
-- ALTER TABLE                             15
-- CREATE VIEW                              0
-- ALTER VIEW                               0
-- CREATE PACKAGE                           0
-- CREATE PACKAGE BODY                      0
-- CREATE PROCEDURE                         0
-- CREATE FUNCTION                          0
-- CREATE TRIGGER                           0
-- ALTER TRIGGER                            0
-- CREATE COLLECTION TYPE                   0
-- CREATE STRUCTURED TYPE                   0
-- CREATE STRUCTURED TYPE BODY              0
-- CREATE CLUSTER                           0
-- CREATE CONTEXT                           0
-- CREATE DATABASE                          0
-- CREATE DIMENSION                         0
-- CREATE DIRECTORY                         0
-- CREATE DISK GROUP                        0
-- CREATE ROLE                              0
-- CREATE ROLLBACK SEGMENT                  0
-- CREATE SEQUENCE                          0
-- CREATE MATERIALIZED VIEW                 0
-- CREATE MATERIALIZED VIEW LOG             0
-- CREATE SYNONYM                           0
-- CREATE TABLESPACE                        0
-- CREATE USER                              0
--
-- DROP TABLESPACE                          0
-- DROP DATABASE                            0
--
-- REDACTION POLICY                         0
--
-- ORDS DROP SCHEMA                         0
-- ORDS ENABLE SCHEMA                       0
-- ORDS ENABLE OBJECT                       0
--
-- ERRORS                                   0
-- WARNINGS                                 0
```