

PRINCIPLES OF DATABASE SYSTEMS PROJECT - PART A

SECTION - B

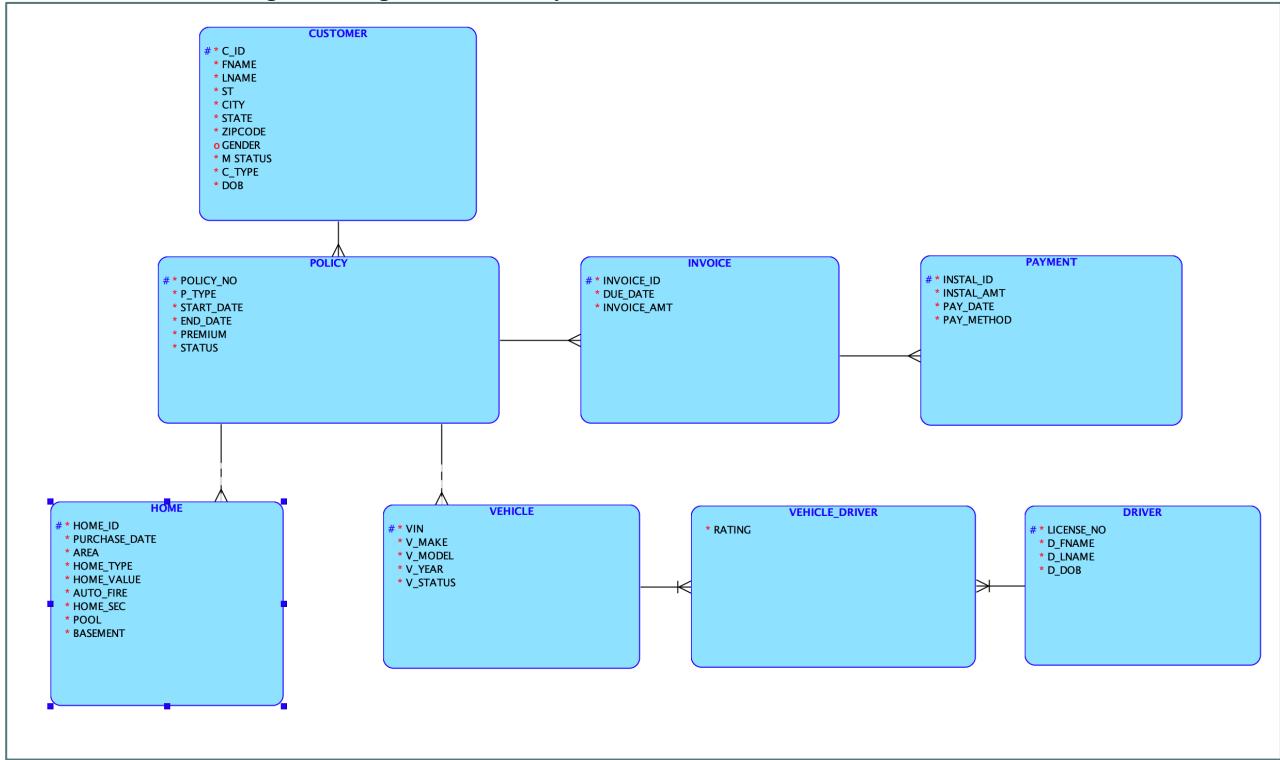
Submitted by
Yashodhan Joshi – yj1400
Rachana Swamy – rms816

Submitted on
04/10/2020

We Do Secure (WDS) Business Case:

- a) Create a logical E-R model for database schema with appropriate relationships amongst them.

Figure 1: Logical E-R model for WDS Home and Auto insurance database



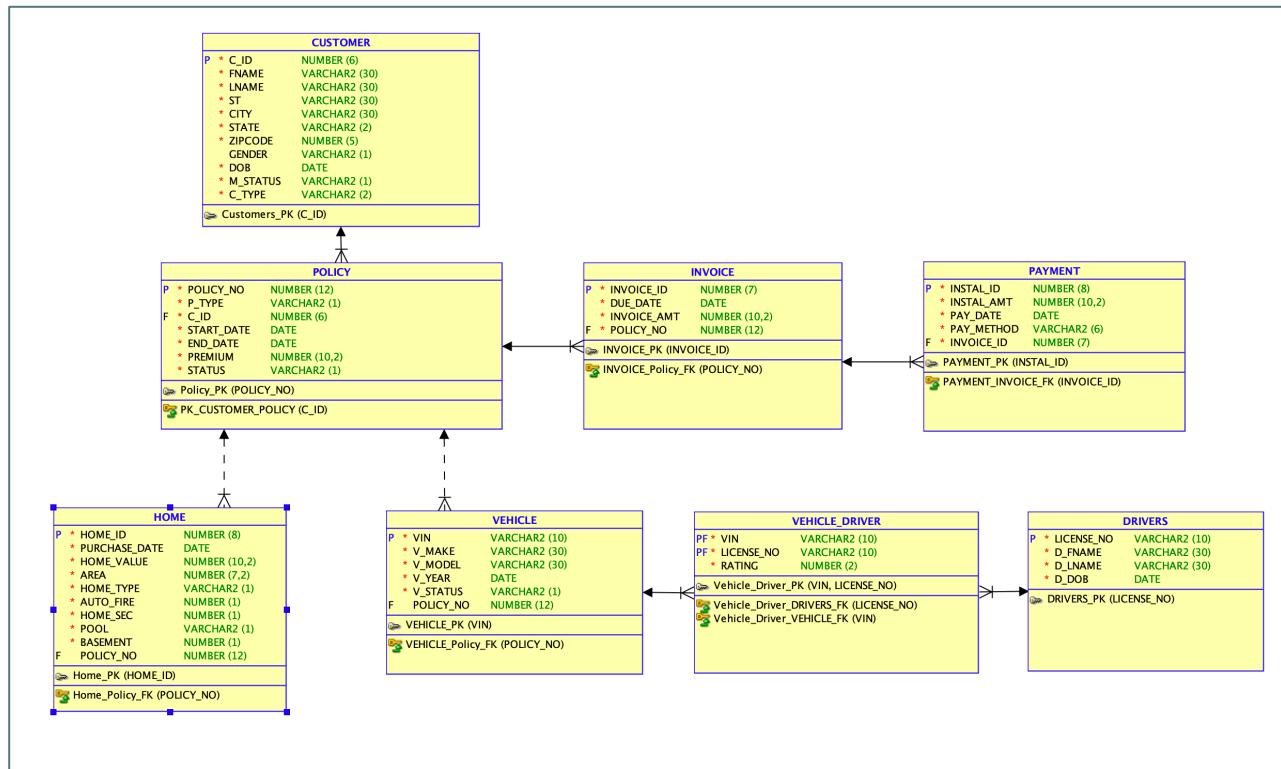
Assumptions:

- Similar to life insurance policies, we have assumed that every home and auto insurance policy has only one name who is the only and main holder, thus Customer-Policy entities have a one-to-many cardinality but not many to many.
- home insurance and auto insurance have a different policy number for the same customer
- p_type can be either A or H which refer to auto insurance policy or home insurance policy respectively. This helps us analyze the policies individually, e.g. the maximum auto insurance premium.
- insurance policies only insure one type of object, e.g. car, home or life. Hence we combined the tables for home and auto insurance policies, but they can still be differentiated using the p_type column and the starting number of each policy
- Vehicle – Driver is a many-to-many relationship assumed by us, since each vehicle can have multiple drivers and each driver can have multiple cars. Hence an intersect entity was added
- Gender is optional, since some people prefer not to reveal their gender nowadays.

- Policy to home and policy to vehicle have optional cardinality on the many side, since a policy might be a home policy or a vehicle policy
- Also, one policy may cover multiple homes or multiple vehicles
- We added customer's date of birth – in order to use age of the customer as one of the factors for determining the value of premium amount
- Rating is an attribute to help determine the auto policy premium amount

b) Create a relational model, depicting all entities, attributes (name, type, size, and mandatory/optional, primary key), relationships (foreign keys).

Figure 2: Relational model for WDS Home and Auto insurance database



- c) Use relational model to create the schema, and their objects, primary keys, foreign keys, and other constraints. Submit DDL code.

```

1  CREATE TABLE customer (
2      c_id      NUMBER(6) NOT NULL,
3      fname     VARCHAR2(30) NOT NULL,
4      lname     VARCHAR2(30) NOT NULL,
5      st        VARCHAR2(30) NOT NULL,
6      city      VARCHAR2(30) NOT NULL,
7      state     VARCHAR2(2) NOT NULL,
8      zipcode   NUMBER(5) NOT NULL,
9      gender    VARCHAR2(1),
10     dob       DATE NOT NULL,
11     m_status  VARCHAR2(1) NOT NULL,
12     c_type    VARCHAR2(2) NOT NULL
13 );
14
15     COMMENT ON COLUMN customer.c_id IS 'Customer Id';
16
17     COMMENT ON COLUMN customer.fname IS 'First Name';
18
19     COMMENT ON COLUMN customer.lname IS 'Last Name';
20
21     COMMENT ON COLUMN customer.st IS 'Street Address';
22
23     COMMENT ON COLUMN customer.city IS 'Name of the city';
24
25     COMMENT ON COLUMN customer.state IS 'State Code';
26
27     COMMENT ON COLUMN customer.zipcode IS 'Postal Code';
28
29     COMMENT ON COLUMN customer.gender IS 'Gender';
30
31     COMMENT ON COLUMN customer.dob IS 'Date of Birth';
32
33     COMMENT ON COLUMN customer.m_status IS 'Marital Status';
34
35     COMMENT ON COLUMN customer.c_type IS 'Customer Type (A / H / AH)';
36
37     ALTER TABLE customer ADD CONSTRAINT customers_pk PRIMARY KEY ( c_id );
38
39
40     CREATE TABLE drivers (
41         license_no  VARCHAR2(10) NOT NULL,
42         d_fname     VARCHAR2(30) NOT NULL,
43         d_lname     VARCHAR2(30) NOT NULL,
44         d_dob       DATE NOT NULL
45 );
46
47     COMMENT ON COLUMN drivers.license_no IS 'Driver''s license number';
48
49     COMMENT ON COLUMN drivers.d_fname IS 'FIRST NAME';
50
51     COMMENT ON COLUMN drivers.d_lname IS 'LAST NAME';
52
53     COMMENT ON COLUMN drivers.d_dob IS 'Driver''s date of birth';
54
55     ALTER TABLE drivers ADD CONSTRAINT drivers_pk PRIMARY KEY ( license_no );

```

```

56
57 CREATE TABLE home (
58     home_id      NUMBER(8) NOT NULL,
59     purchase_date DATE NOT NULL,
60     home_value   NUMBER(10, 2) NOT NULL,
61     area         NUMBER(7, 2) NOT NULL,
62     home_type    VARCHAR2(1) NOT NULL,
63     auto_fire    NUMBER(1) NOT NULL,
64     home_sec     NUMBER(1) NOT NULL,
65     pool          VARCHAR2(1),
66     basement     NUMBER(1) NOT NULL,
67     policy_no    NUMBER(12)
68 );
69
70 COMMENT ON COLUMN home.home_id IS 'Unique Id for each home insured';
71
72 COMMENT ON COLUMN home.purchase_date IS 'Day the house was purchased';
73
74 COMMENT ON COLUMN home.home_value IS 'Home purchase value';
75
76 COMMENT ON COLUMN home.area IS 'Area in Sqft';
77
78 COMMENT ON COLUMN home.home_type IS 'Type of Home';
79
80 COMMENT ON COLUMN home.auto_fire IS 'Auto Fire Notification';
81
82 COMMENT ON COLUMN home.home_sec IS 'Home Security System';
83
84 COMMENT ON COLUMN home.pool IS 'Swimming Pool';
85
86 COMMENT ON COLUMN home.basement IS 'DOES THE HOUSE HAVE A BASEMENT';
87
88 COMMENT ON COLUMN home.policy_no IS 'Policy Number';
89
90 ALTER TABLE home ADD CONSTRAINT home_pk PRIMARY KEY ( home_id );
91
92 CREATE TABLE invoice (
93     invoice_id   NUMBER(7) NOT NULL,
94     due_date     DATE NOT NULL,
95     invoice_amt  NUMBER(10, 2) NOT NULL,
96     policy_no    NUMBER(12) NOT NULL
97 );
98
99 COMMENT ON COLUMN invoice.invoice_id IS 'INVOICE NUMBER FOR HOME POLICY';
100
101 COMMENT ON COLUMN invoice.due_date IS 'DAY PAYMENT IS DUE';
102
103 COMMENT ON COLUMN invoice.invoice_amt IS 'AMOUNT DUE';
104
105 COMMENT ON COLUMN invoice.policy_no IS 'Policy Number';
106
107 ALTER TABLE invoice ADD CONSTRAINT invoice_pk PRIMARY KEY ( invoice_id );
108

```

```

109  CREATE TABLE payment (
110      instal_id    NUMBER(8) NOT NULL,
111      instal_amt   NUMBER(10, 2) NOT NULL,
112      pay_date     DATE NOT NULL,
113      pay_method   VARCHAR2(6) NOT NULL,
114      invoice_id   NUMBER(7) NOT NULL
115  );
116
117  COMMENT ON COLUMN payment.instal_id IS 'INSTALLMENT ID';
118
119  COMMENT ON COLUMN payment.instal_amt IS 'AMOUNT PAID IN INSTALMENT';
120
121  COMMENT ON COLUMN payment.pay_date IS 'PAYMENT DATE';
122
123  COMMENT ON COLUMN payment.pay_method IS 'Method of PAYMENT';
124
125  ALTER TABLE payment ADD CONSTRAINT payment_pk PRIMARY KEY ( instal_id );
126
127  CREATE TABLE policy (
128      policy_no    NUMBER(12) NOT NULL,
129      p_type       VARCHAR2(1) NOT NULL,
130      c_id         NUMBER(6) NOT NULL,
131      start_date   DATE NOT NULL,
132      end_date     DATE NOT NULL,
133      premium      NUMBER(10, 2) NOT NULL,
134      status        VARCHAR2(1) NOT NULL
135  );
136
137  COMMENT ON COLUMN policy.policy_no IS 'Policy Number';
138
139  COMMENT ON COLUMN policy.p_type IS 'Primary Type (A/H)';
140
141  COMMENT ON COLUMN policy.c_id IS 'Customer ID';
142
143  COMMENT ON COLUMN policy.start_date IS 'POLICY START DATE';
144
145  COMMENT ON COLUMN policy.end_date IS 'POLICY END DATE';
146
147  COMMENT ON COLUMN policy.premium IS 'MONTHLY PREMIUM AMOUNT';
148
149  COMMENT ON COLUMN policy.status IS 'POLICY STATUS ( Current/Expired)';
150
151  ALTER TABLE policy ADD CONSTRAINT policy_pk PRIMARY KEY ( policy_no );

```

```

152
153 CREATE TABLE vehicle (
154     vin          VARCHAR2(10) NOT NULL,
155     v_make       VARCHAR2(30) NOT NULL,
156     v_model      VARCHAR2(30) NOT NULL,
157     v_year       DATE NOT NULL,
158     v_status     VARCHAR2(1) NOT NULL,
159     policy_no    NUMBER(12)
160 );
161
162 COMMENT ON COLUMN vehicle.vin IS 'VEHICLE IDENTIFICATION NUMBER';
163
164 COMMENT ON COLUMN vehicle.v_make IS 'VEHICLE MAKE AND MODEL';
165
166 COMMENT ON COLUMN vehicle.v_model IS 'Vehicle Model';
167
168 COMMENT ON COLUMN vehicle.v_year IS 'YEAR THE CAR WAS BOUGHT';
169
170 COMMENT ON COLUMN vehicle.v_status IS 'STATUS IS L,F,O';
171
172 ALTER TABLE vehicle ADD CONSTRAINT vehicle_pk PRIMARY KEY ( vin );
173
174 CREATE TABLE vehicle_driver (
175     vin          VARCHAR2(10) NOT NULL,
176     license_no   VARCHAR2(10) NOT NULL,
177     rating       NUMBER(2) NOT NULL
178 );
179
180 COMMENT ON COLUMN vehicle_driver.vin IS 'Vehicle Id Number';
181
182 COMMENT ON COLUMN vehicle_driver.license_no IS 'Driver''s license number';
183
184 COMMENT ON COLUMN vehicle_driver.rating IS 'DRIVER RATINGS ';
185
186 ALTER TABLE vehicle_driver ADD CONSTRAINT vehicle_driver_pk ]
187     PRIMARY KEY ( vin, license_no );
188
189 ALTER TABLE home
190     ADD CONSTRAINT home_policy_fk FOREIGN KEY ( policy_no )
191         REFERENCES policy ( policy_no );
192
193 ALTER TABLE invoice
194     ADD CONSTRAINT invoice_policy_fk FOREIGN KEY ( policy_no )
195         REFERENCES policy ( policy_no );
196
197 ALTER TABLE payment
198     ADD CONSTRAINT payment_invoice_fk FOREIGN KEY ( invoice_id )
199         REFERENCES invoice ( invoice_id );
200
201 ALTER TABLE policy
202     ADD CONSTRAINT pk_customer_policy FOREIGN KEY ( c_id )
203         REFERENCES customer ( c_id );
204
205 ALTER TABLE vehicle_driver
206     ADD CONSTRAINT vehicle_driver_drivers_fk FOREIGN KEY ( license_no )
207         REFERENCES drivers ( license_no );
208
209 ALTER TABLE vehicle_driver
210     ADD CONSTRAINT vehicle_driver_vehicle_fk FOREIGN KEY ( vin )
211         REFERENCES vehicle ( vin );
212
213 ALTER TABLE vehicle
214     ADD CONSTRAINT vehicle_policy_fk FOREIGN KEY ( policy_no )
215         REFERENCES policy ( policy_no );

```

--- Oracle SQL Developer Data Modeler Summary Report:	

--- CREATE TABLE	8
--- CREATE INDEX	0
--- 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

d) Write commands and apply necessary CHECK constraints to apply defined business rules to enforce data consistency.

```
217 ----- Constraints -----
218
219 ALTER TABLE customer
220     ADD CONSTRAINT c_cust_id CHECK ( c_id BETWEEN 100000 AND 999999);
221
222 ALTER TABLE customer
223     ADD CONSTRAINT c_cust_name CHECK ( fname = UPPER(fname) AND lname = UPPER(lname)) ;
224
225 ALTER TABLE customer
226     ADD CONSTRAINT c_cust_city CHECK (city = UPPER(city));
227
228 ALTER TABLE customer
229     ADD CONSTRAINT c_cust_state CHECK (state = UPPER(state) AND LENGTH(state)>=2);
230
231 ALTER TABLE customer
232     ADD CONSTRAINT c_cust_zipcode CHECK ( LENGTH(zipcode)>=5 );
233
234 --ALTER TABLE customer
235 --    MODIFY ( gender DEFAULT NULL);
236
237 ALTER TABLE customer
238     ADD CONSTRAINT c_cust_gender CHECK ( gender IN ('M','F') );
239
240 ALTER TABLE customer
241     ADD CONSTRAINT c_cust_mstatus CHECK ( m_status IN ('M','S','W'));
242
243 ALTER TABLE customer
244     ADD CONSTRAINT c_cust_ctype CHECK ( c_type IN ('A','H','AH'));
245
246 ALTER TABLE policy
247     ADD CONSTRAINT c_policy_no CHECK ( policy_no BETWEEN 10000000000 AND 99999999999);
248
249 ALTER TABLE policy
250     ADD CONSTRAINT c_policy_ptype CHECK ( p_type IN ('A','H'));
251
252 ALTER TABLE policy
253     ADD CONSTRAINT c_policy_date CHECK ( end_date > start_date);
254
255 ALTER TABLE policy
256     ADD CONSTRAINT c_policy_status CHECK ( status IN ('C','P'));
257
258 ALTER TABLE invoice
259     ADD CONSTRAINT c_invoice_id CHECK ( invoice_id BETWEEN 100000 AND 999999);
260
261 ALTER TABLE payment
262     ADD CONSTRAINT c_payment_instal_id CHECK ( instal_id BETWEEN 1000000 AND 9999999);
263
264 ALTER TABLE payment
265     ADD CONSTRAINT c_payment_method CHECK ( pay_method IN ('CREDIT','DEBIT','PAYPAL','CHEQUE'));
266
267 ALTER TABLE home
268     ADD CONSTRAINT c_home_id CHECK ( home_id BETWEEN 1000000 AND 9999999);
269
270 ALTER TABLE home
271     ADD CONSTRAINT c_home_type CHECK ( home_type IN ('S','M','C','T'));
```

```
272  
273 ALTER TABLE home  
274     ADD CONSTRAINT c_home_fire CHECK ( auto_fire IN (1,0));  
275  
276 ALTER TABLE home  
277     ADD CONSTRAINT c_home_sec CHECK ( home_sec IN (1,0));  
278  
279 ALTER TABLE home  
280     ADD CONSTRAINT c_home_pool CHECK ( pool IN ('U','O','I','M'));  
281  
282 ALTER TABLE home  
283     ADD CONSTRAINT c_home_basement CHECK ( basement IN (1,0));  
284  
285 ALTER TABLE vehicle  
286     ADD CONSTRAINT c_vehicle_vin CHECK ( LENGTH(vin)=10 );  
287  
288 ALTER TABLE vehicle  
289     ADD CONSTRAINT c_vehicle_vstatus CHECK ( v_status IN ('L','F','O'));  
290  
291 ALTER TABLE drivers  
292     ADD CONSTRAINT c_license_no CHECK ( LENGTH(license_no)=10 );  
293  
294 ALTER TABLE drivers  
295     ADD CONSTRAINT c_driver_name CHECK ( d_fname = UPPER(d_fname) AND d_lname = UPPER(d_lname) );  
296
```

e) Populate meaningful sample data for all entities (10 to 15 records per entity).

Figure 3: Insert statements for table: CUSTOMER

```

346 ----- insert -----
347 INSERT INTO CUSTOMER VALUES (100001,'MARSHAL','ERICKSON','215 BROADWAY','NEW YORK','NY',11222,'M',
348 TO_DATE('07/02/1986 00:00:00','MM/DD/YYYY HH24:MI:SS'),'M','A');
349 INSERT INTO CUSTOMER VALUES (100002,'LILY','ALDRIN','332 LAFAYETTE','NEW YORK','NY',11224,'F',
350 TO_DATE('06/24/1986 00:00:00','MM/DD/YYYY HH24:MI:SS'),'M','H');
351 INSERT INTO CUSTOMER VALUES (100003,'BARNEY','STINSON','562 UPPER WEST','NEW YORK','NY',11227,'M',
352 TO_DATE('10/12/1988 00:00:00','MM/DD/YYYY HH24:MI:SS'),'S','H');
353 INSERT INTO CUSTOMER VALUES (100004,'TED','MOSBY','215 BROADWAY','NEW YORK','NY',11222,'M',
354 TO_DATE('08/11/1987 00:00:00','MM/DD/YYYY HH24:MI:SS'),'S','H');
355 INSERT INTO CUSTOMER VALUES (100005,'ROBIN','SCHERBATSKY','452 WOODBRIDGE','BROOKLYN','NY',11242,'F',
356 TO_DATE('05/05/1989 00:00:00','MM/DD/YYYY HH24:MI:SS'),'W','H');
357 INSERT INTO CUSTOMER VALUES (100006,'OSCAR','MYER','224 DANBURY','COOKS COUNTY','WA',12345,'M',
358 TO_DATE('02/09/1956 00:00:00','MM/DD/YYYY HH24:MI:SS'),'W','AH');
359 INSERT INTO CUSTOMER VALUES (100007,'QUINTIN','TARANTINO','BEVERLY HILL','LOS ANGELES','CA',23419,'M',
360 TO_DATE('04/08/1966 00:00:00','MM/DD/YYYY HH24:MI:SS'),'M','AH');
361 INSERT INTO CUSTOMER VALUES (100008,'MARK','ZUCKERBERG','452 COFFEE LANE','REDMOND','WA',89675,'M',
362 TO_DATE('07/07/1990 00:00:00','MM/DD/YYYY HH24:MI:SS'),'M','AH');
363 INSERT INTO CUSTOMER VALUES (100009,'ELON','MUSK','333 83RD ST','BROOKLYN','NY',11667,'M',
364 TO_DATE('06/22/1980 00:00:00','MM/DD/YYYY HH24:MI:SS'),'S','H');
365 INSERT INTO CUSTOMER VALUES (100010,'ANDY','HAMITION','111 LA GUARDIA','NEW YORK','NY',11236,'M',
366 TO_DATE('03/12/1974 00:00:00','MM/DD/YYYY HH24:MI:SS'),'M','AH');
367 INSERT INTO CUSTOMER VALUES (100011,'YASHI','JOSHI','8020','BROOKLYN','NY',189879,'M',
368 TO_DATE('09/09/1989 00:00:00','MM/DD/YYYY HH24:MI:SS'),'S','A');
369 INSERT INTO CUSTOMER VALUES (100012,'RACHANA','SWAMY','555 25 ST','BROOKLYN','NY',11217,'F',
370 TO_DATE('05/08/1966 00:00:00','MM/DD/YYYY HH24:MI:SS'),'S','A');
371 INSERT INTO CUSTOMER (C_ID,FNAME,LNAME,ST,CITY,STATE,ZIPCODE,DOB,M_STATUS,C_TYPE)
372 VALUES (100013,'JANE','DOE','111 1ST','BROOKLYN','NY',11111,TO_DATE('01/01/2001 00:00:00','MM/DD/YYYY HH24:MI:SS'),'W','H');
373

```

Figure 4: Insert statements for table: POLICY

```

375
376 INSERT INTO POLICY (POLICY_NO,P_TYPE,C_ID,START_DATE,END_DATE,PREMIUM,STATUS)
377 VALUES (100000000101,'A',100001,TO_DATE('07/01/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),TO_DATE('06/30/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),425,'C');
378 INSERT INTO POLICY VALUES (100000000201,'H',100002,TO_DATE('02/28/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
379 TO_DATE('03/30/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),1000,'C');
380 INSERT INTO POLICY VALUES (100000000202,'H',100003,TO_DATE('08/01/2017 00:00:00','MM/DD/YYYY HH24:MI:SS'),
381 TO_DATE('09/30/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),1100,'P');
382 INSERT INTO POLICY VALUES (100000000203,'H',100004,TO_DATE('09/13/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
383 TO_DATE('09/12/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),700,'C');
384 INSERT INTO POLICY VALUES (100000000204,'H',100005,TO_DATE('01/01/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
385 TO_DATE('06/01/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),900,'C');
386 INSERT INTO POLICY VALUES (100000000205,'H',100006,TO_DATE('04/04/2018 00:00:00','MM/DD/YYYY HH24:MI:SS'),
387 TO_DATE('04/03/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),1100,'C');
388 INSERT INTO POLICY VALUES (100000000102,'A',100006,TO_DATE('04/04/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
389 TO_DATE('04/03/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),2400,'C');
390 INSERT INTO POLICY VALUES (100000000206,'H',100007,TO_DATE('10/14/2017 00:00:00','MM/DD/YYYY HH24:MI:SS'),
391 TO_DATE('10/13/2018 00:00:00','MM/DD/YYYY HH24:MI:SS'),9000,'P');
392 INSERT INTO POLICY VALUES (100000000103,'A',100007,TO_DATE('10/14/2017 00:00:00','MM/DD/YYYY HH24:MI:SS'),
393 TO_DATE('10/03/2018 00:00:00','MM/DD/YYYY HH24:MI:SS'),1400,'P');
394 INSERT INTO POLICY VALUES (100000000104,'A',100007,TO_DATE('04/08/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),
395 TO_DATE('04/07/2022 00:00:00','MM/DD/YYYY HH24:MI:SS'),2200,'C');
396 INSERT INTO POLICY VALUES (100000000207,'H',100008,TO_DATE('10/14/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
397 TO_DATE('10/13/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),29000,'C');
398 INSERT INTO POLICY VALUES (100000000105,'A',100008,TO_DATE('10/14/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
399 TO_DATE('10/03/2021 00:00:00','MM/DD/YYYY HH24:MI:SS'),400,'C');
400 INSERT INTO POLICY VALUES (100000000106,'A',100009,TO_DATE('04/07/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
401 TO_DATE('03/07/2020 00:00:00','MM/DD/YYYY HH24:MI:SS'),1200,'C');
402 INSERT INTO POLICY VALUES (100000000107,'A',100010,TO_DATE('11/14/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
403 TO_DATE('11/03/2021 00:00:00','MM/DD/YYYY HH24:MI:SS'),900,'C');
404 INSERT INTO POLICY VALUES (100000000108,'A',100010,TO_DATE('07/07/2018 00:00:00','MM/DD/YYYY HH24:MI:SS'),
405 TO_DATE('07/06/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),900,'P');
406 INSERT INTO POLICY VALUES (100000000208,'H',100010,TO_DATE('10/14/2017 00:00:00','MM/DD/YYYY HH24:MI:SS'),
407 TO_DATE('10/13/2021 00:00:00','MM/DD/YYYY HH24:MI:SS'),12000,'C');
408 INSERT INTO POLICY VALUES (100000000209,'H',100010,TO_DATE('03/14/2015 00:00:00','MM/DD/YYYY HH24:MI:SS'),
409 TO_DATE('03/13/2017 00:00:00','MM/DD/YYYY HH24:MI:SS'),10000,'P');
410 INSERT INTO POLICY VALUES (100000000109,'A',100011,TO_DATE('09/14/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
411 TO_DATE('09/03/2021 00:00:00','MM/DD/YYYY HH24:MI:SS'),300,'C');
412 INSERT INTO POLICY VALUES (100000000110,'A',100011,TO_DATE('10/14/2016 00:00:00','MM/DD/YYYY HH24:MI:SS'),
413 TO_DATE('10/13/2017 00:00:00','MM/DD/YYYY HH24:MI:SS'),220,'P');
414 INSERT INTO POLICY VALUES (100000000111,'A',100012,TO_DATE('07/02/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
415 TO_DATE('07/02/2021 00:00:00','MM/DD/YYYY HH24:MI:SS'),200,'C');
416 INSERT INTO POLICY VALUES (100000000210,'H',100013,TO_DATE('10/14/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
417 TO_DATE('11/03/2022 00:00:00','MM/DD/YYYY HH24:MI:SS'),2400,'C');
418 INSERT INTO POLICY VALUES (100000000211,'H',100013,TO_DATE('12/14/2019 00:00:00','MM/DD/YYYY HH24:MI:SS'),
419 TO_DATE('11/03/2021 00:00:00','MM/DD/YYYY HH24:MI:SS'),4400,'C');
420

```

Figure 5: Insert statements for table: INVOICE

```

422
423     INSERT INTO INVOICE (INVOICE_ID, DUE_DATE, INVOICE_AMT, POLICY_NO)
424         VALUES ( 1001101,TO_DATE('07/02/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 425, 100000000101);
425     INSERT INTO INVOICE VALUES ( 1001201,TO_DATE('03/28/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 1000, 100000000201);
426     INSERT INTO INVOICE VALUES ( 1001202,TO_DATE('08/01/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 1100, 100000000202);
427     INSERT INTO INVOICE VALUES ( 1001203,TO_DATE('10/13/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 700, 100000000203);
428     INSERT INTO INVOICE VALUES ( 1001204,TO_DATE('02/01/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 900, 100000000204);
429     INSERT INTO INVOICE VALUES ( 1001205,TO_DATE('05/04/2018 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 5000, 100000000205);
430     INSERT INTO INVOICE VALUES ( 1001206,TO_DATE('10/04/2018 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 6000, 100000000205);
431     INSERT INTO INVOICE VALUES ( 1001107,TO_DATE('04/04/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 2400, 100000000102);
432     INSERT INTO INVOICE VALUES ( 1001208,TO_DATE('12/14/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 9000, 100000000206);
433     INSERT INTO INVOICE VALUES ( 1001109,TO_DATE('12/14/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 1400, 100000000103);
434     INSERT INTO INVOICE VALUES ( 1001110,TO_DATE('05/08/2020 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 2200, 100000000104);
435     INSERT INTO INVOICE VALUES ( 1001211,TO_DATE('11/14/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 20000, 100000000207);
436     INSERT INTO INVOICE VALUES ( 1001212,TO_DATE('12/14/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 9000, 100000000207);
437     INSERT INTO INVOICE VALUES ( 1001113,TO_DATE('11/14/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 400, 100000000105);
438     INSERT INTO INVOICE VALUES ( 1001114,TO_DATE('05/07/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 1200, 100000000106);
439     INSERT INTO INVOICE VALUES ( 1001115,TO_DATE('01/13/2020 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 900, 100000000107);
440     INSERT INTO INVOICE VALUES ( 1001116,TO_DATE('08/07/2018 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 900, 100000000108);
441     INSERT INTO INVOICE VALUES ( 1001217,TO_DATE('11/14/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 12000, 100000000208);
442     INSERT INTO INVOICE VALUES ( 1001218,TO_DATE('04/14/2015 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 10000, 100000000209);
443     INSERT INTO INVOICE VALUES ( 1001219,TO_DATE('10/14/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 300, 100000000109);
444     INSERT INTO INVOICE VALUES ( 1001120,TO_DATE('12/14/2016 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 220, 100000000110);
445     INSERT INTO INVOICE VALUES ( 1001121,TO_DATE('08/02/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 200, 100000000111);
446     INSERT INTO INVOICE VALUES ( 1001222,TO_DATE('11/14/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 2400, 100000000210);
447     INSERT INTO INVOICE VALUES ( 1001223,TO_DATE('01/14/2020 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 4400, 100000000211);
448

```

Figure 6: Insert statements for table: PAYMENT

```

449     --SELECT COUNT(*) FROM INVOICE;
450
451     INSERT INTO PAYMENT (INSTAL_ID, INSTAL_AMT, PAY_DATE, PAY_METHOD, INVOICE_ID)
452         VALUES (10001101, 425, TO_DATE('07/01/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CREDIT', 1001101);
453     INSERT INTO PAYMENT VALUES ( 10001201, 1000, TO_DATE('03/18/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001201);
454     INSERT INTO PAYMENT VALUES ( 10001202, 1100, TO_DATE('08/01/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'DEBIT', 1001202);
455     INSERT INTO PAYMENT VALUES ( 10001203, 700, TO_DATE('10/13/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'DEBIT', 1001203);
456     INSERT INTO PAYMENT VALUES ( 10001204, 900, TO_DATE('01/15/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'DEBIT', 1001204);
457     INSERT INTO PAYMENT VALUES ( 10001205, 5000, TO_DATE('05/04/2018 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001205);
458     INSERT INTO PAYMENT VALUES ( 10001102, 2400, TO_DATE('04/02/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'PAYPAL', 1001202);
459     INSERT INTO PAYMENT VALUES ( 10001206, 9000, TO_DATE('12/10/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001206);
460     INSERT INTO PAYMENT VALUES ( 10001103, 1400, TO_DATE('12/01/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001203);
461     INSERT INTO PAYMENT VALUES ( 10001104, 2200, TO_DATE('05/01/2020 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001204);
462     INSERT INTO PAYMENT VALUES ( 10001207, 10000, TO_DATE('11/10/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001107);
463     INSERT INTO PAYMENT VALUES ( 10001208, 10000, TO_DATE('11/13/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001107);
464     INSERT INTO PAYMENT VALUES ( 10001105, 400, TO_DATE('11/09/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'PAYPAL', 1001205);
465     INSERT INTO PAYMENT VALUES ( 10001106, 1200, TO_DATE('05/01/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CREDIT', 1001206);
466     INSERT INTO PAYMENT VALUES ( 10001107, 900, TO_DATE('01/15/2020 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'DEBIT', 1001107);
467     INSERT INTO PAYMENT VALUES ( 10001108, 900, TO_DATE('08/07/2018 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'DEBIT', 1001208);
468     INSERT INTO PAYMENT VALUES ( 10001209, 10000, TO_DATE('11/01/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001208);
469     INSERT INTO PAYMENT VALUES ( 10001210, 2000, TO_DATE('11/04/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CREDIT', 1001208);
470     INSERT INTO PAYMENT VALUES ( 10001211, 10000, TO_DATE('04/04/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CREDIT', 1001219);
471     INSERT INTO PAYMENT VALUES ( 10001109, 300, TO_DATE('10/14/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'DEBIT', 1001109);
472     INSERT INTO PAYMENT VALUES ( 10001110, 220, TO_DATE('12/11/2016 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'DEBIT', 1001110);
473     INSERT INTO PAYMENT VALUES ( 10001111, 200, TO_DATE('08/01/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'PAYPAL', 1001110);
474     INSERT INTO PAYMENT VALUES ( 10001212, 2400, TO_DATE('10/14/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CREDIT', 1001211);
475     INSERT INTO PAYMENT VALUES ( 10001213, 4400, TO_DATE('01/02/2020 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'CHEQUE', 1001211);
476

```

Figure 7: Insert statements for table: HOME

```

478
479     INSERT INTO HOME (HOME_ID, PURCHASE_DATE, HOME_VALUE, AREA, HOME_TYPE, AUTO_FIRE, HOME_SEC, POOL, BASEMENT, POLICY_NO)
480         VALUES (10003001,TO_DATE('03/01/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 400000, 1100, 'S', 1, 1, Null, 0, 100000000201 );
481     INSERT INTO HOME VALUES ( 10003002,TO_DATE('07/01/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 250000, 700, 'M', 1, 1, Null, 1, 100000000202 );
482     INSERT INTO HOME VALUES ( 10003003,TO_DATE('09/13/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 150000, 500, 'S', 1, 1, Null, 0, 100000000203 );
483     INSERT INTO HOME VALUES ( 10003004,TO_DATE('01/05/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 190000, 800, 'M', 1, 1, 'O', 1, 1, 100000000204 );
484     INSERT INTO HOME VALUES ( 10003005,TO_DATE('03/04/2018 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 500000, 1500, 'T', 1, 1, 'I', 1, 1, 100000000205 );
485     INSERT INTO HOME VALUES ( 10003006,TO_DATE('10/19/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 300000, 1100, 'C', 1, 0, Null, 0, 100000000206 );
486     INSERT INTO HOME VALUES ( 10003007,TO_DATE('10/18/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 1400000, 2100, 'T', 1, 1, 'U', 0, 100000000207 );
487     INSERT INTO HOME VALUES ( 10003008,TO_DATE('10/18/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 2400000, 3100, 'T', 1, 1, 'M', 1, 1, 100000000207 );
488     INSERT INTO HOME VALUES ( 10003009,TO_DATE('10/16/2017 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 800000, 1300, 'C', 0, 1, Null, 0, 100000000208 );
489     INSERT INTO HOME VALUES ( 10003010,TO_DATE('03/25/2015 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 1000000, 1500, 'S', 1, 1, Null, 0, 100000000209 );
490     INSERT INTO HOME VALUES ( 10003011,TO_DATE('04/21/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 600000, 12400, 'M', 1, 1, Null, 1, 100000000210 );
491     INSERT INTO HOME VALUES ( 10003012,TO_DATE('12/15/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 500000, 900, 'S', 1, 1, Null, 0, 100000000211 );
492     INSERT INTO HOME VALUES ( 10003013,TO_DATE('12/15/2019 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 500000, 900, 'S', 1, 1, Null, 0, 100000000211 );
493

```

Figure 8: Insert statements for table: VEHICLE

```

495
496 INSERT INTO VEHICLE (VIN, V_MAKE, V_MODEL, V_YEAR, V_STATUS, POLICY_NO) VALUES ('M121314121', 'HONDA', 'ACCORD', TO_DATE('2018', 'YYYY'), '0', 100000000101);
497 INSERT INTO VEHICLE VALUES ('M128299909', 'BMW', 'M3', TO_DATE('2019', 'YYYY'), '0', 100000000102);
498 INSERT INTO VEHICLE VALUES ('M128537088', 'VW', 'POLO', TO_DATE('2019', 'YYYY'), '0', 100000000102);
499 INSERT INTO VEHICLE VALUES ('M121105964', 'HONDA', 'CRV', TO_DATE('2016', 'YYYY'), 'L', 100000000103);
500 INSERT INTO VEHICLE VALUES ('N122644567', 'VW', 'PASSAT', TO_DATE('2015', 'YYYY'), 'F', 100000000104);
501 INSERT INTO VEHICLE VALUES ('N125277889', 'VW', 'POLO', TO_DATE('2017', 'YYYY'), 'L', 100000000105);
502 INSERT INTO VEHICLE VALUES ('N332314978', 'BMW', 'M5', TO_DATE('2017', 'YYYY'), 'F', 100000000106);
503 INSERT INTO VEHICLE VALUES ('0456314111', 'AUDI', 'Q7', TO_DATE('2020', 'YYYY'), 'F', 100000000107);
504 INSERT INTO VEHICLE VALUES ('0121314123', 'HONDA', 'DYNAMO', TO_DATE('2015', 'YYYY'), 'L', 100000000108);
505 INSERT INTO VEHICLE VALUES ('C177894532', 'BAJAJ', 'CHETAK', TO_DATE('2015', 'YYYY'), '0', 100000000109);
506 INSERT INTO VEHICLE VALUES ('C167731467', 'VW', 'PHAETON', TO_DATE('2019', 'YYYY'), '0', 100000000110);
507 INSERT INTO VEHICLE VALUES ('D990131479', 'VW', 'POLO', TO_DATE('2018', 'YYYY'), '0', 100000000111);
508

```

Figure 9: Insert statements for table: DRIVERS

```

510
511 INSERT INTO DRIVERS (LICENSE_NO, D_FNAME, D_LNAME, D_DOB)
512     VALUES ('A456788148', 'MARSHAL', 'ERICKSON', TO_DATE('07/02/1986 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
513 INSERT INTO DRIVERS VALUES ('A667888148', 'OSCAR', 'MYER', TO_DATE('02/09/1956 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
514 INSERT INTO DRIVERS VALUES ('A455778148', 'QUINTIN', 'TARANTINTO', TO_DATE('04/08/1966 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
515 INSERT INTO DRIVERS VALUES ('A432798148', 'MARK', 'ZUCKERBERG', TO_DATE('07/07/1990 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
516 INSERT INTO DRIVERS VALUES ('A453388148', 'ANDY', 'HAMILTON', TO_DATE('03/12/1974 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
517 INSERT INTO DRIVERS VALUES ('A776788148', 'YASH', 'JOSHI', TO_DATE('09/09/1989 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
518 INSERT INTO DRIVERS VALUES ('A226798148', 'RACHANA', 'SWAMY', TO_DATE('05/08/1966 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
519 INSERT INTO DRIVERS VALUES ('A116778148', 'JASON', 'BOURNE', TO_DATE('03/12/1984 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
520 INSERT INTO DRIVERS VALUES ('A450988148', 'LESLIE', 'BOURNE', TO_DATE('03/12/1974 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
521 INSERT INTO DRIVERS VALUES ('A156548148', 'MIKAELA', 'BLOMKWIST', TO_DATE('04/11/1972 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
522 INSERT INTO DRIVERS VALUES ('A465488148', 'LISBETH', 'SALANDER', TO_DATE('03/10/1987 00:00:00', 'MM/DD/YYYY HH24:MI:SS'));
523

```

Figure 10: Insert statements for table: VEHICLE_DRIVER

```

525
526 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('M121314121', 'A456788148', 8);
527 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('M128299909', 'A667888148', 6);
528 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('M128537088', 'A455778148', 9);
529 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('M121105964', 'A455778148', 8);
530 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('M121314121', 'A432798148', 7);
531 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('N122644567', 'A453388148', 5);
532 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('N125277889', 'A776788148', 9);
533 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('N332314978', 'A226798148', 8);
534 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('0456314111', 'A450988148', 8);
535 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('0121314123', 'A156548148', 9);
536 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('C177894532', 'A465488148', 7);
537 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('C167731467', 'A432798148', 8);
538 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('D990131479', 'A465488148', 9);
539 INSERT INTO VEHICLE_DRIVER (VIN, LICENSE_NO, RATING) VALUES ('D990131479', 'A432798148', 8);
540

```

- f) List total number of records populated for each entity (just record counts, not full data set).

24	SELECT COUNT(*) FROM customer;
25	COUNT(*)
26	13

[Download CSV](#)

33	SELECT COUNT(*) FROM policy;
34	COUNT(*)
35	15

[Download CSV](#)

66	SELECT COUNT(*) FROM invoice;
67	COUNT(*)
68	24

[Download CSV](#)

93	SELECT COUNT(*) FROM PAYMENT;
94	COUNT(*)
95	24

[Download CSV](#)

109	SELECT COUNT(*) FROM HOME;
110	COUNT(*)
111	13

[Download CSV](#)

124	SELECT COUNT(*) FROM VEHICLE;
125	COUNT(*)
126	12

[Download CSV](#)

138	SELECT COUNT(*) FROM DRIVERS;
139	COUNT(*)
140	11

[Download CSV](#)

4	SELECT COUNT(*) FROM VEHICLE_DRIVER;
5	COUNT(*)
6	14

[Download CSV](#)

- g) Write data dictionary queries that details all tables, columns-data-type-size-mandatory/optional, constraints, and attribute comments of schema objects.

Figure 11: Data dictionary query listing the tables

81 REM: List of Tables									
82 select table_name									
83 from user_tables;									
84									
 : List of Tables									
 <table border="1"><thead><tr><th>TABLE_NAME</th></tr></thead><tbody><tr><td>CUSTOMER</td></tr><tr><td>DRIVERS</td></tr><tr><td>HOME</td></tr><tr><td>INVOICE</td></tr><tr><td>PAYMENT</td></tr><tr><td>POLICY</td></tr><tr><td>VEHICLE</td></tr><tr><td>VEHICLE_DRIVER</td></tr></tbody></table>	TABLE_NAME	CUSTOMER	DRIVERS	HOME	INVOICE	PAYMENT	POLICY	VEHICLE	VEHICLE_DRIVER
TABLE_NAME									
CUSTOMER									
DRIVERS									
HOME									
INVOICE									
PAYMENT									
POLICY									
VEHICLE									
VEHICLE_DRIVER									
Download CSV									
8 rows selected.									

Figure 12: Data dictionary query listing the columns, their data type, size, mandatory or optional

85 REM: List of Table Columns																																																																																																						
86 select table_name, column_name, data_type, data_length, data_precision, nullable as "Optional (y/n)"																																																																																																						
87 from user_tab_columns order by table_name, column_id;																																																																																																						
88																																																																																																						
 : List of Table Columns																																																																																																						
 <table border="1"><thead><tr><th>TABLE_NAME</th><th>COLUMN_NAME</th><th>DATA_TYPE</th><th>DATA_LENGTH</th><th>DATA_PRECISION</th><th>Optional (y/n)</th></tr></thead><tbody><tr><td>CUSTOMER</td><td>C_ID</td><td>NUMBER</td><td>22</td><td>6</td><td>N</td></tr><tr><td>CUSTOMER</td><td>FNAME</td><td>VARCHAR2</td><td>30</td><td>–</td><td>N</td></tr><tr><td>CUSTOMER</td><td>LNAME</td><td>VARCHAR2</td><td>30</td><td>–</td><td>N</td></tr><tr><td>CUSTOMER</td><td>ST</td><td>VARCHAR2</td><td>30</td><td>–</td><td>N</td></tr><tr><td>CUSTOMER</td><td>CITY</td><td>VARCHAR2</td><td>30</td><td>–</td><td>N</td></tr><tr><td>CUSTOMER</td><td>STATE</td><td>VARCHAR2</td><td>2</td><td>–</td><td>N</td></tr><tr><td>CUSTOMER</td><td>ZIPCODE</td><td>NUMBER</td><td>22</td><td>5</td><td>N</td></tr><tr><td>CUSTOMER</td><td>GENDER</td><td>VARCHAR2</td><td>1</td><td>–</td><td>Y</td></tr><tr><td>CUSTOMER</td><td>DOB</td><td>DATE</td><td>7</td><td>–</td><td>N</td></tr><tr><td>CUSTOMER</td><td>M_STATUS</td><td>VARCHAR2</td><td>1</td><td>–</td><td>N</td></tr><tr><td>CUSTOMER</td><td>C_TYPE</td><td>VARCHAR2</td><td>2</td><td>–</td><td>N</td></tr><tr><td>DRIVERS</td><td>LICENSE_NO</td><td>VARCHAR2</td><td>10</td><td>–</td><td>N</td></tr><tr><td>DRIVERS</td><td>D_FNAME</td><td>VARCHAR2</td><td>30</td><td>–</td><td>N</td></tr><tr><td>DRIVERS</td><td>D_LNAME</td><td>VARCHAR2</td><td>30</td><td>–</td><td>N</td></tr><tr><td>DRIVERS</td><td>D_DOB</td><td>DATE</td><td>7</td><td>–</td><td>N</td></tr><tr><td>HOME</td><td>HOME_ID</td><td>NUMBER</td><td>22</td><td>8</td><td>N</td></tr></tbody></table>	TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION	Optional (y/n)	CUSTOMER	C_ID	NUMBER	22	6	N	CUSTOMER	FNAME	VARCHAR2	30	–	N	CUSTOMER	LNAME	VARCHAR2	30	–	N	CUSTOMER	ST	VARCHAR2	30	–	N	CUSTOMER	CITY	VARCHAR2	30	–	N	CUSTOMER	STATE	VARCHAR2	2	–	N	CUSTOMER	ZIPCODE	NUMBER	22	5	N	CUSTOMER	GENDER	VARCHAR2	1	–	Y	CUSTOMER	DOB	DATE	7	–	N	CUSTOMER	M_STATUS	VARCHAR2	1	–	N	CUSTOMER	C_TYPE	VARCHAR2	2	–	N	DRIVERS	LICENSE_NO	VARCHAR2	10	–	N	DRIVERS	D_FNAME	VARCHAR2	30	–	N	DRIVERS	D_LNAME	VARCHAR2	30	–	N	DRIVERS	D_DOB	DATE	7	–	N	HOME	HOME_ID	NUMBER	22	8	N
TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION	Optional (y/n)																																																																																																	
CUSTOMER	C_ID	NUMBER	22	6	N																																																																																																	
CUSTOMER	FNAME	VARCHAR2	30	–	N																																																																																																	
CUSTOMER	LNAME	VARCHAR2	30	–	N																																																																																																	
CUSTOMER	ST	VARCHAR2	30	–	N																																																																																																	
CUSTOMER	CITY	VARCHAR2	30	–	N																																																																																																	
CUSTOMER	STATE	VARCHAR2	2	–	N																																																																																																	
CUSTOMER	ZIPCODE	NUMBER	22	5	N																																																																																																	
CUSTOMER	GENDER	VARCHAR2	1	–	Y																																																																																																	
CUSTOMER	DOB	DATE	7	–	N																																																																																																	
CUSTOMER	M_STATUS	VARCHAR2	1	–	N																																																																																																	
CUSTOMER	C_TYPE	VARCHAR2	2	–	N																																																																																																	
DRIVERS	LICENSE_NO	VARCHAR2	10	–	N																																																																																																	
DRIVERS	D_FNAME	VARCHAR2	30	–	N																																																																																																	
DRIVERS	D_LNAME	VARCHAR2	30	–	N																																																																																																	
DRIVERS	D_DOB	DATE	7	–	N																																																																																																	
HOME	HOME_ID	NUMBER	22	8	N																																																																																																	

	HOME	PURCHASE_DATE	DATE	7	-	N
	HOME	HOME_VALUE	NUMBER	22	10	N
	HOME	AREA	NUMBER	22	7	N
	HOME	HOME_TYPE	VARCHAR2	1	-	N
	HOME	AUTO_FIRE	NUMBER	22	1	N
	HOME	HOME_SEC	NUMBER	22	1	N
	HOME	POOL	VARCHAR2	1	-	N
	HOME	BASEMENT	NUMBER	22	1	N
	HOME	POLICY_NO	NUMBER	22	12	Y
	INVOICE	INVOICE_ID	NUMBER	22	7	N
	INVOICE	DUE_DATE	DATE	7	-	N
	INVOICE	INVOICE_AMT	NUMBER	22	10	N
	INVOICE	POLICY_NO	NUMBER	22	12	N
	PAYMENT	INSTAL_ID	NUMBER	22	8	N
	PAYMENT	INSTAL_AMT	NUMBER	22	10	N
	PAYMENT	PAY_DATE	DATE	7	-	N
	PAYMENT	PAY_METHOD	VARCHAR2	6	-	N
	PAYMENT	INVOICE_ID	NUMBER	22	7	N
	POLICY	POLICY_NO	NUMBER	22	12	N
	POLICY	P_TYPE	VARCHAR2	1	-	N
	POLICY	C_ID	NUMBER	22	6	N
	POLICY	START_DATE	DATE	7	-	N
	POLICY	END_DATE	DATE	7	-	N
	POLICY	PREMIUM	NUMBER	22	10	N
	POLICY	STATUS	VARCHAR2	1	-	N
	VEHICLE	VIN	VARCHAR2	10	-	N
	VEHICLE	V_MAKE	VARCHAR2	30	-	N
	VEHICLE	V_MODEL	VARCHAR2	30	-	N
	VEHICLE	V_YEAR	DATE	7	-	N
	VEHICLE	V_STATUS	VARCHAR2	1	-	N
	VEHICLE	POLICY_NO	NUMBER	22	12	Y
	VEHICLE_DRIVER	VIN	VARCHAR2	10	-	N
	VEHICLE_DRIVER	LICENSE_NO	VARCHAR2	10	-	N
	VEHICLE_DRIVER	RATING	NUMBER	22	2	N

[Download CSV](#)
50 rows selected.

Figure 13: Data dictionary query listing the columns, their data type, size, mandatory or optional

Figure 14: Data dictionary query listing the columns and their comments

89	REM: List of Table Column Comments	
90		
91	select table_name,column_name,comments	
92	from user_col_comments	
93	order by table_name;	
 : List of Table Column Comments		
TABLE_NAME	COLUMN_NAME	COMMENTS
CUSTOMER	DOB	Date of Birth
CUSTOMER	GENDER	Gender
CUSTOMER	C_TYPE	Customer Type (A / H / AH)
CUSTOMER	M_STATUS	Marital Status
CUSTOMER	C_ID	Customer Id
CUSTOMER	ZIPCODE	Postal Code
CUSTOMER	STATE	State Code
CUSTOMER	CITY	Name of the city
CUSTOMER	ST	Street Address
CUSTOMER	LNAME	Last Name
CUSTOMER	FNAME	First Name
DRIVERS	D_LNAME	LAST NAME
DRIVERS	D_FNAME	FIRST NAME
DRIVERS	LICENSE_NO	Driver's license number
DRIVERS	D_DOB	Driver's date of birth
HOME	BASEMENT	DOES THE HOUSE HAVE A BASEMENT
HOME	POOL	Swimming Pool
HOME	HOME_SEC	Home Security System
HOME	AUTO_FIRE	Auto Fire Notification
HOME	HOME_TYPE	Type of Home
HOME	AREA	Area in Sqft
HOME	HOME_VALUE	Home purchase value
HOME	PURCHASE_DATE	Day the house was purchased
HOME	HOME_ID	Unique Id for each home insured
HOME	POLICY_NO	Policy Number
INVOICE	POLICY_NO	Policy Number
INVOICE	INVOICE_ID	INVOICE NUMBER FOR HOME POLICY
INVOICE	DUE_DATE	DAY PAYMENT IS DUE
INVOICE	INVOICE_AMT	AMOUNT DUE
PAYMENT	PAY_METHOD	Method of PAYMENT
PAYMENT	PAY_DATE	PAYMENT DATE
PAYMENT	INSTAL_AMT	AMOUNT PAID IN INSTALMENT
PAYMENT	INVOICE_ID	-
PAYMENT	INSTAL_ID	INSTALLMENT ID

Figure 15: Data dictionary query listing the columns and their comments

PAYMENT	INVOICE_ID	-
PAYMENT	INSTAL_ID	INSTALLMENT ID
POLICY	C_ID	Customer ID
POLICY	P_TYPE	Primary Type (A/H)
POLICY	POLICY_NO	Policy Number
POLICY	START_DATE	POLICY START DATE
POLICY	STATUS	POLICY STATUS (Current/Expired)
POLICY	PREMIUM	MONTHLY PREMIUM AMOUNT
POLICY	END_DATE	POLICY END DATE
VEHICLE	VIN	VEHICLE IDENTIFICATION NUMBER
VEHICLE	V_MAKE	VEHICLE MAKE AND MODEL
VEHICLE	V_MODEL	Vehicle Model
VEHICLE	V_YEAR	YEAR THE CAR WAS BOUGHT
VEHICLE	V_STATUS	STATUS IS L,F,O
VEHICLE	POLICY_NO	-
VEHICLE_DRIVER	VIN	Vehicle Id Number
VEHICLE_DRIVER	LICENSE_NO	Driver's license number
VEHICLE_DRIVER	RATING	DRIVER RATINGS

Download CSV

50 rows selected.

Figure 16: Data dictionary query listing constraints

1 REM: List of Table Column Constraints						
2 select table_name,constraint_name,constraint_type,search_condition,						
3 r_owner as referential_owner, r_constraint_name as referential_constraint_name						
4 from user_constraints						
5 order by 1,2;						
: List of Table Column Constraints						
TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	REFERENTIAL_OWNER	REFERENTIAL_CONSTRAINT_NAME	
CUSTOMER	CUSTOMERS_PK	P	-	-	-	
CUSTOMER	C_CUST_CITY	C	city = UPPER(city)	-	-	
CUSTOMER	C_CUST_CTYPE	C	c_type IN ('A','H','AH')	-	-	
CUSTOMER	C_CUST_GENDER	C	gender IN ('M','F')	-	-	
CUSTOMER	C_CUST_ID	C	c_id BETWEEN 100000 AND 999999	-	-	
CUSTOMER	C_CUST_MSTATUS	C	m_status IN ('M','S','W')	-	-	
CUSTOMER	C_CUST_NAME	C	fname = UPPER(fname) AND lname = UPPER(lname)	-	-	
CUSTOMER	C_CUST_STATE	C	state = UPPER(state) AND LENGTH(state)>=2	-	-	
CUSTOMER	C_CUST_ZIPCODE	C	LENGTH(zipcode)>=5	-	-	
CUSTOMER	SYS_C0027924084	C	"C_ID" IS NOT NULL	-	-	
CUSTOMER	SYS_C0027924085	C	"FNAME" IS NOT NULL	-	-	
CUSTOMER	SYS_C0027924086	C	"LNAME" IS NOT NULL	-	-	
CUSTOMER	SYS_C0027924087	C	"ST" IS NOT NULL	-	-	
CUSTOMER	SYS_C0027924088	C	"CITY" IS NOT NULL	-	-	
CUSTOMER	SYS_C0027924089	C	"STATE" IS NOT NULL	-	-	
CUSTOMER	SYS_C0027924090	C	"ZIPCODE" IS NOT NULL	-	-	

CUSTOMER	SYS_C0027924091	C	"DOB" IS NOT NULL	-	-
CUSTOMER	SYS_C0027924092	C	"M_STATUS" IS NOT NULL	-	-
CUSTOMER	SYS_C0027924093	C	"C_TYPE" IS NOT NULL	-	-
DRIVERS	C_LICENSE_NO	C	LENGTH(license_no)=10	-	-
DRIVERS	DRIVERS_PK	P	-	-	-
DRIVERS	SYS_C0027924158	C	"LICENSE_NO" IS NOT NULL	-	-
DRIVERS	SYS_C0027924159	C	"D_FNAME" IS NOT NULL	-	-
DRIVERS	SYS_C0027924160	C	"D_LNAME" IS NOT NULL	-	-
DRIVERS	SYS_C0027924161	C	"D_DOB" IS NOT NULL	-	-
HOME	C_HOME_BASEMENT	C	basement IN (1,0)	-	-
HOME	C_HOME_FIRE	C	auto_fire IN (1,0)	-	-
HOME	C_HOME_ID	C	home_id BETWEEN 1000000 AND 9999999	-	-
HOME	C_HOME_POOL	C	pool IN ('U','O','I','M')	-	-
HOME	C_HOME_SEC	C	home_sec IN (1,0)	-	-
HOME	C_HOME_TYPE	C	home_type IN ('S','M','C','T')	-	-
HOME	HOME_PK	P	-	-	-
HOME	HOME_POLICY_FK	R	-	SQL_LHGKYCX50WOXACTRJVQVAOF	POLICY_PK
HOME	SYS_C0027924180	C	"HOME_ID" IS NOT NULL	-	-
HOME	SYS_C0027924181	C	"PURCHASE_DATE" IS NOT NULL	-	-
HOME	SYS_C0027924182	C	"HOME_VALUE" IS NOT NULL	-	-
HOME	SYS_C0027924183	C	"AREA" IS NOT NULL	-	-
HOME	SYS_C0027924184	C	"HOME_TYPE" IS NOT NULL	-	-
HOME	SYS_C0027924185	C	"AUTO_FIRE" IS NOT NULL	-	-
HOME	SYS_C0027924186	C	"HOME_SEC" IS NOT NULL	-	-
HOME	SYS_C0027924187	C	"BASEMENT" IS NOT NULL	-	-
INVOICE	C_INVOICE_ID	C	invoice_id BETWEEN 100000 AND 999999	-	-
INVOICE	INVOICE_PK	P	-	-	-
INVOICE	INVOICE_POLICY_FK	R	-	SQL_LHGKYCX50WOXACTRJVQVAOF	POLICY_PK
INVOICE	SYS_C0027924234	C	"INVOICE_ID" IS NOT NULL	-	-
INVOICE	SYS_C0027924235	C	"DUE_DATE" IS NOT NULL	-	-
INVOICE	SYS_C0027924236	C	"INVOICE_AMT" IS NOT NULL	-	-
INVOICE	SYS_C0027924237	C	"POLICY_NO" IS NOT NULL	-	-
PAYMENT	C_PAYMENT_INSTAL_ID	C	instal_id BETWEEN 1000000 AND 9999999	-	-
PAYMENT	C_PAYMENT_METHOD	C	pay_method IN ('CREDIT','DEBIT','PAYPAL','CHEQUE')	-	-
PAYMENT	PAYMENT_INVOICE_FK	R	-	SQL_RGNRCWOFWNIMTKOKFXWTYYORH	INVOICE_PK
PAYMENT	PAYMENT_PK	P	-	-	-
PAYMENT	SYS_C0028016300	C	"INSTAL_ID" IS NOT NULL	-	-
PAYMENT	SYS_C0028016301	C	"INSTAL_AMT" IS NOT NULL	-	-
PAYMENT	SYS_C0028016302	C	"PAY_DATE" IS NOT NULL	-	-
PAYMENT	SYS_C0028016303	C	"PAY_METHOD" IS NOT NULL	-	-
PAYMENT	SYS_C0028016304	C	"INVOICE_ID" IS NOT NULL	-	-
POLICY	C_POLICY_DATE	C	end_date > start_date	-	-
POLICY	C_POLICY_NO	C	policy_no BETWEEN 1000000000 AND 9999999999	-	-
POLICY	C_POLICY_PTYPE	C	p_type IN ('A','H')	-	-
POLICY	C_POLICY_STATUS	C	status IN ('C','P')	-	-
POLICY	PK_CUSTOMER_POLICY	R	-	SQL_RGNRCWOFWNIMTKOKFXWTYYORH	CUSTOMERS_PK
POLICY	POLICY_PK	P	-	-	-
POLICY	SYS_C0028016310	C	"POLICY_NO" IS NOT NULL	-	-
POLICY	SYS_C0028016311	C	"P_TYPE" IS NOT NULL	-	-
POLICY	SYS_C0028016312	C	"C_ID" IS NOT NULL	-	-
POLICY	SYS_C0028016313	C	"START_DATE" IS NOT NULL	-	-
POLICY	SYS_C0028016314	C	"END_DATE" IS NOT NULL	-	-
POLICY	SYS_C0028016315	C	"PREMIUM" IS NOT NULL	-	-
POLICY	SYS_C0028016316	C	"STATUS" IS NOT NULL	-	-

VEHICLE	C_VEHICLE_VIN	C	LENGTH(vin)=10	-	-
VEHICLE	C_VEHICLE_VSTATUS	C	V_STATUS IN ('L','F','O')	-	-
VEHICLE	SYS_C0028016328	C	"VIN" IS NOT NULL	-	-
VEHICLE	SYS_C0028016329	C	"V_MAKE" IS NOT NULL	-	-
VEHICLE	SYS_C0028016330	C	"V_MODEL" IS NOT NULL	-	-
VEHICLE	SYS_C0028016331	C	"V_YEAR" IS NOT NULL	-	-
VEHICLE	SYS_C0028016332	C	"V_STATUS" IS NOT NULL	-	-
VEHICLE	VEHICLE_PK	P	-	-	-
VEHICLE	VEHICLE_POLICY_FK	R	-	SQL_RGNRCWOFWNIMTKOKFXWTYYORH	POLICY_PK
VEHICLE_DRIVER	SYS_C0028016336	C	"VIN" IS NOT NULL	-	-
VEHICLE_DRIVER	SYS_C0028016337	C	"LICENSE_NO" IS NOT NULL	-	-
VEHICLE_DRIVER	SYS_C0028016338	C	"RATING" IS NOT NULL	-	-
VEHICLE_DRIVER	VEHICLE_DRIVER_DRIVERS_FK	R	-	SQL_RGNRCWOFWNIMTKOKFXWTYYORH	DRIVERS_PK
VEHICLE_DRIVER	VEHICLE_DRIVER_VEHICLE_FK	R	-	SQL_RGNRCWOFWNIMTKOKFXWTYYORH	VEHICLE_PK

h) Write SQL queries using each of following,

Q1) Table joins with at least 3 tables in join

List Customer IDs, Customer first name, Policy numbers, Policy type, Invoice ID and Invoice Amount where Invoice Amount is greater than 1000

```

1 /*List Customer ID, Customer Name, their policies and associated invoices*/
2
3 SELECT a.C_ID, a.FNAME, b.POLICY_NO, b.P_TYPE, c.INVOICE_ID, c.INVOICE_AMT FROM CUSTOMER a JOIN POLICY b ON a.C_ID = b.C_ID
4 JOIN INVOICE c ON b.POLICY_NO=c.POLICY_NO WHERE INVOICE_AMT > 1000 ORDER BY 4,1;
```

C_ID	FNAME	POLICY_NO	P_TYPE	INVOICE_ID	INVOICE_AMT
100006	OSCAR	10000000102	A	1001107	2400
100007	QUINTIN	10000000104	A	1001110	2200
100007	QUINTIN	10000000103	A	1001109	1400
100009	ELON	10000000106	A	1001114	1200

Q2) Multi-row subquery

List those auto policies whose premiums are higher than the average

```
1  /* List those AUTO policies whose premiums are higher than the average */
2  SELECT POLICY_NO, P_TYPE, PREMIUM FROM POLICY
3  WHERE
4  PREMIUM > ALL(SELECT AVG(PREMIUM) FROM POLICY WHERE P_TYPE = 'A') AND P_TYPE = 'A' ;
5
```

POLICY_NO	P_TYPE	PREMIUM
100000000102	A	2400
100000000103	A	1400
100000000104	A	2200
100000000106	A	1200

[Download CSV](#)

4 rows selected.

Q3) Co-related subquery

List those policies whose premiums are higher then their respective policy type's average

```
1  /* Q3) Co-related subquery
2  # List those policies whose premiums are higher then their respective policy types average*/
3
4  SELECT a.POLICY_NO, a.P_TYPE, a.PREMIUM FROM POLICY a
5  WHERE
6  a.PREMIUM > (SELECT AVG(PREMIUM) avgpremium FROM POLICY b WHERE a.P_TYPE= b.P_TYPE) ORDER BY 2;
```

POLICY_NO	P_TYPE	PREMIUM
100000000104	A	2200
100000000106	A	1200
100000000103	A	1400
100000000102	A	2400
100000000209	H	10000
100000000208	H	12000
100000000206	H	9000
100000000207	H	29000
100000000205	H	11000

[Download CSV](#)

9 rows selected.

Q4) SET operator query

Finding people with multiple policies as opposed to single policy holders

```
1 /*Q4) SET operator query
2 # FINDING PEOPLE WITH MULTIPLE POLICIES AS OPPOSED TO SINGLE POLICY HOLDERS*/
3
4 SELECT POLICY_NO, P_TYPE, C_ID, STATUS FROM POLICY WHERE STATUS = 'C'
5 INTERSECT
6 SELECT POLICY_NO, P_TYPE, C_ID, STATUS FROM POLICY WHERE P_TYPE = 'H';
7
```

POLICY_NO	P_TYPE	C_ID	STATUS
100000000201	H	100002	C
100000000203	H	100004	C
100000000204	H	100005	C
100000000205	H	100006	C
100000000207	H	100008	C
100000000208	H	100010	C
100000000210	H	100013	C
100000000211	H	100013	C

[Download CSV](#)

8 rows selected.

Q5) Query with any analytical function or in line view or WITH clause

Find a safe home and then find out if it is a multi family home

```

1  /* Q5) Query with any analytical function or in line view or WITH clause
2  FIRST FIND A SAFE HOME AND THEN FIND OUT IF IT IS A MULTI FAMILY HOME*/
3
4  CREATE OR REPLACE VIEW SAFE_HOME
5  AS
6  SELECT * FROM HOME WHERE AUTO_FIRE = 1 AND HOME_SEC =1 WITH READ ONLY;
7
8  SELECT * FROM SAFE_HOME WHERE HOME_TYPE = 'M';
9
10
11

```

View created.



HOME_ID	PURCHASE_DATE	HOME_VALUE	AREA	HOME_TYPE	AUTO_FIRE	HOME_SEC	POOL	BASEMENT	POLICY_NO
10003002	01-JUL-17	250000	700	M	1	1	-	1	100000000202
10003004	05-JAN-19	190000	800	M	1	1	0	1	100000000204
10003011	21-APR-19	600000	12400	M	1	1	-	1	100000000210

[Download CSV](#)

3 rows selected.