NYU – TANDON SCHOOL OF ENGINEERING
CS-GY 6083 - B, SPRING 2020
Principles of Database Systems

Project Part 1

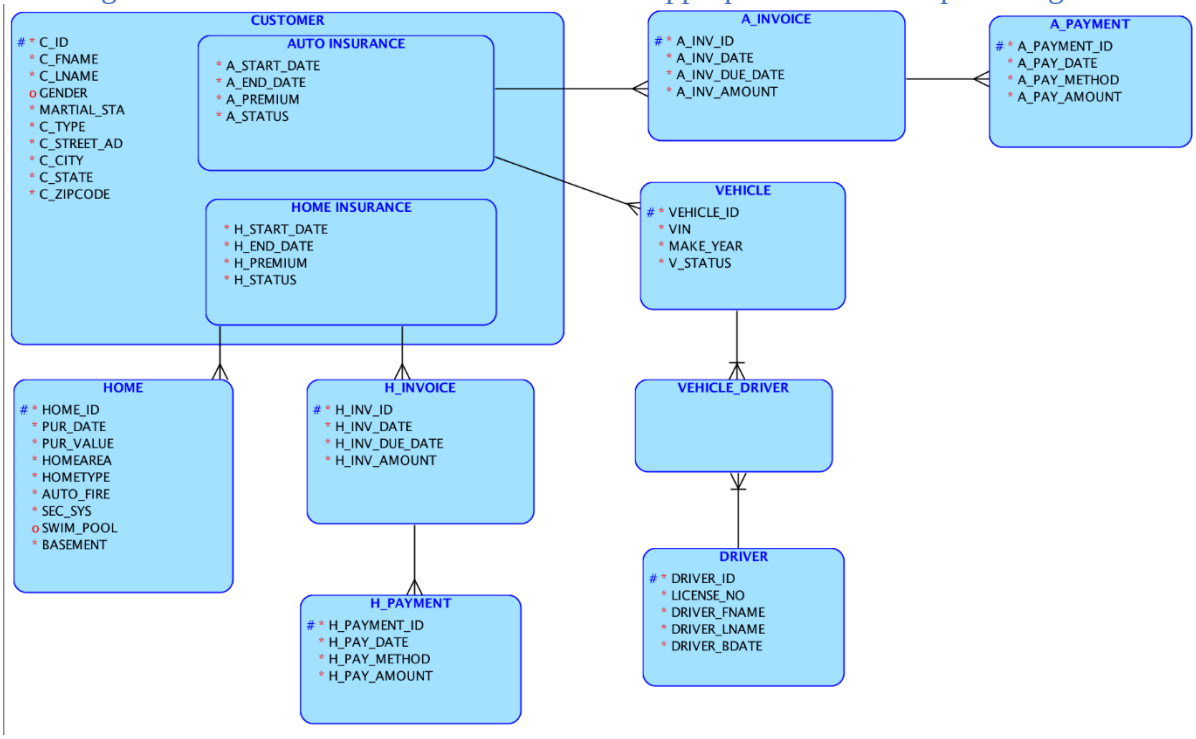
Student Name: Johnny Wang Net ID: yw4504
Shangxi Sun ss12934

Submission Date: 04/11/2020

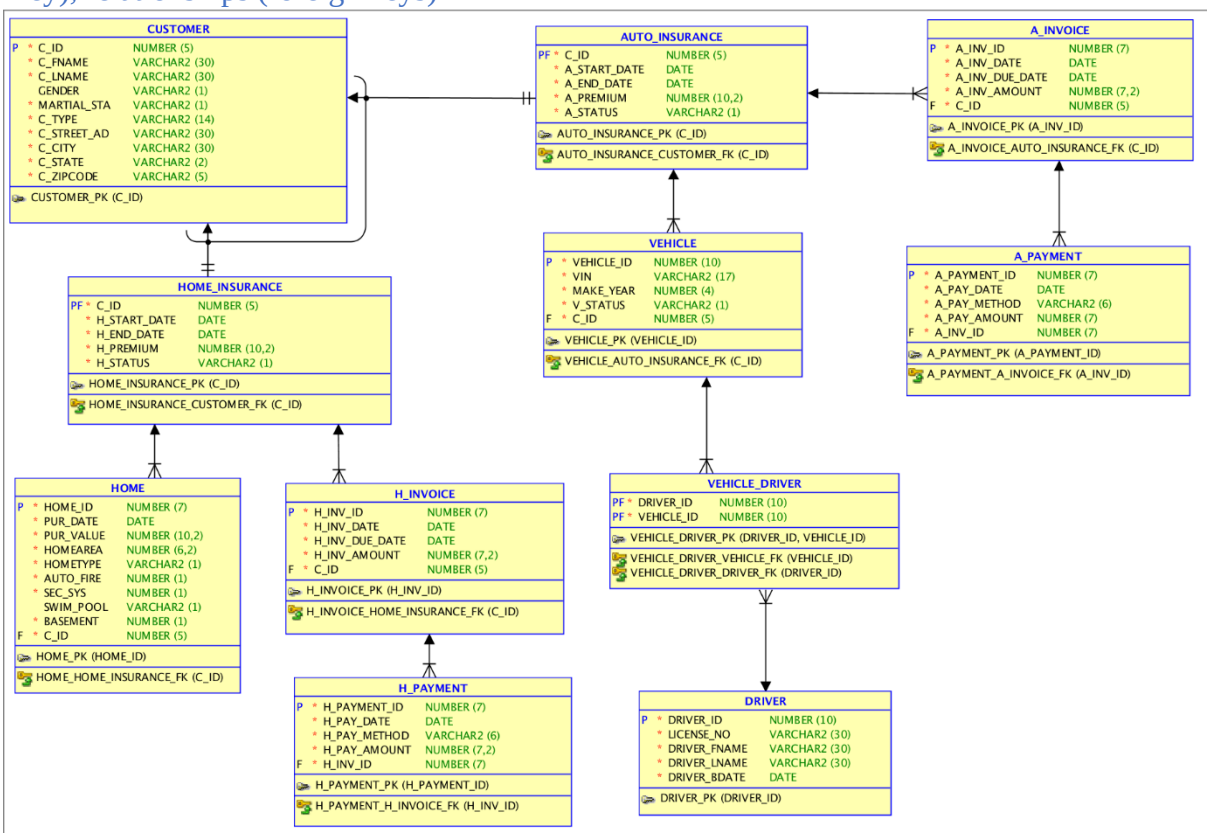
Description

Our design for the business case of We Do Secure insurance provider company follows these basic ideas:
1. Make the ‘CUSTOMER’ as supertype entity, which contains two subtype entities, ‘AUTO INSURANCE’ and ‘HOME INSURANCE’.
2. The disjointness constraints in this supertype/subtype relationship need to follow the overlap rule, and which subtype one customer’s account belongs to is determined by the attribute ‘C-TYPE’. We use extra constraints and triggers to address it.
3. We assume that any two customers won’t insure the same home/ vehicle.
4. We assume that each driver can have multiple vehicles, so relationship between ‘VEHICLE’ and ‘DRIVER’ is many-to-many.

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



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



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

CREATE TABLE a_invoice (

```
    a_inv_id        NUMBER(7) NOT NULL,
    a_inv_date      DATE NOT NULL,
    a_inv_due_date  DATE NOT NULL,
    a_inv_amount    NUMBER(7, 2) NOT NULL,
    c_id            NUMBER(5) NOT NULL
);

COMMENT ON COLUMN a_invoice.a_inv_id IS
    'Auto insurance"s invoice number';

COMMENT ON COLUMN a_invoice.a_inv_date IS
    'Auto insurance"s invoice date';

COMMENT ON COLUMN a_invoice.a_inv_due_date IS
    'Auto insurance"s payment due date';

COMMENT ON COLUMN a_invoice.a_inv_amount IS
    'Auto insurance"s invoice amount';

ALTER TABLE a_invoice ADD CONSTRAINT a_invoice_pk PRIMARY KEY ( a_inv_id );

CREATE TABLE a_payment (
    a_payment_id  NUMBER(7) NOT NULL,
    a_pay_date    DATE NOT NULL,
    a_pay_method  VARCHAR2(6) NOT NULL,
    a_pay_amount  NUMBER(7) NOT NULL,
    a_inv_id      NUMBER(7) NOT NULL
);

COMMENT ON COLUMN a_payment.a_payment_id IS
    'Auto insurance"s payment number';

COMMENT ON COLUMN a_payment.a_pay_date IS
    'Auto insurance"s payment date';

COMMENT ON COLUMN a_payment.a_pay_method IS
    'Auto insurance"s method of payment. The pamyment method shoud be one of the following:“PayPal”, “Credit”, “Debit”,
“Check”.';

COMMENT ON COLUMN a_payment.a_pay_amount IS
    'Auto insurance"s payment amount';

ALTER TABLE a_payment ADD CONSTRAINT a_payment_pk PRIMARY KEY ( a_payment_id );

CREATE TABLE auto_insurance (
    c_id          NUMBER(5) NOT NULL,
    a_start_date  DATE NOT NULL,
    a_end_date    DATE NOT NULL,
    a_premium     NUMBER(10, 2) NOT NULL,
    a_status      VARCHAR2(1) NOT NULL
);

COMMENT ON COLUMN auto_insurance.c_id IS
    'Customer"s ID';

COMMENT ON COLUMN auto_insurance.a_start_date IS
    'Auto insurance"s start date';

COMMENT ON COLUMN auto_insurance.a_end_date IS
    'Auto insurance"s end date';

COMMENT ON COLUMN auto_insurance.a_premium IS
    'Auto insurance"s premium amount';

COMMENT ON COLUMN auto_insurance.a_status IS
    'Auto policy insurance status. "C" for current and "P" for expired.';

ALTER TABLE auto_insurance ADD CONSTRAINT auto_insurance_pk PRIMARY KEY ( c_id );

CREATE TABLE customer (
```

```
    c_id          NUMBER(5) NOT NULL,
    c_fname       VARCHAR2(30) NOT NULL,
    c_lname       VARCHAR2(30) NOT NULL,
    gender        VARCHAR2(1),
    martial_sta   VARCHAR2(1) NOT NULL,
    c_type        VARCHAR2(14) NOT NULL,
    c_street_ad   VARCHAR2(30) NOT NULL,
    c_city        VARCHAR2(30) NOT NULL,
    c_state       VARCHAR2(2) NOT NULL,
    c_zipcode     VARCHAR2(5) NOT NULL
);

COMMENT ON COLUMN customer.c_id IS
    'Customer"s ID';

COMMENT ON COLUMN customer.c_fname IS
    'Customer"s first name';

COMMENT ON COLUMN customer.c_lname IS
    'Customer"s last name';

COMMENT ON COLUMN customer.gender IS
    'Customer"s gender. “M”, or “F” representing “Male” or “Female” respectively.';

COMMENT ON COLUMN customer.martial_sta IS
    'Customer"s martial status. “M”, “S”, or “W”, representing “Married”, “Single”, and “Widow/Widower” respectively. ';

COMMENT ON COLUMN customer.c_type IS
    'Customer type';

COMMENT ON COLUMN customer.c_street_ad IS
    'Customer"s street  address';

COMMENT ON COLUMN customer.c_city IS
    'The city of the customer"s address';

COMMENT ON COLUMN customer.c_state IS
    'The state abbr of the customer"s address';

COMMENT ON COLUMN customer.c_zipcode IS
    ' The 5-digit zip code of the customer"s address';

ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY ( c_id );

CREATE TABLE driver (
    driver_id     NUMBER(10) NOT NULL,
    license_no    VARCHAR2(30) NOT NULL,
    driver_fname  VARCHAR2(30) NOT NULL,
    driver_lname  VARCHAR2(30) NOT NULL,
    driver_bdate  DATE NOT NULL
);

COMMENT ON COLUMN driver.driver_id IS
    'Driver"s unique identifier';

COMMENT ON COLUMN driver.license_no IS
    'Driver"s license number';

COMMENT ON COLUMN driver.driver_fname IS
    'Driver"s first name';

COMMENT ON COLUMN driver.driver_lname IS
    'Driver"s last name';

COMMENT ON COLUMN driver.driver_bdate IS
    'Driver"s birthdate';

ALTER TABLE driver ADD CONSTRAINT driver_pk PRIMARY KEY ( driver_id );

CREATE TABLE h_invoice (
```

```
    h_inv_id        NUMBER(7) NOT NULL,
    h_inv_date      DATE NOT NULL,
    h_inv_due_date  DATE NOT NULL,
    h_inv_amount    NUMBER(7, 2) NOT NULL,
    c_id            NUMBER(5) NOT NULL
);

COMMENT ON COLUMN h_invoice.h_inv_id IS
    'Home insurance"s invoice number';

COMMENT ON COLUMN h_invoice.h_inv_date IS
    'Home insurance"s invoice date';

COMMENT ON COLUMN h_invoice.h_inv_due_date IS
    'Home insurance"s payment due date';

COMMENT ON COLUMN h_invoice.h_inv_amount IS
    'Home insurance"s invoice amount';

ALTER TABLE h_invoice ADD CONSTRAINT h_invoice_pk PRIMARY KEY ( h_inv_id );

CREATE TABLE h_payment (
    h_payment_id  NUMBER(7) NOT NULL,
    h_pay_date    DATE NOT NULL,
    h_pay_method  VARCHAR2(6) NOT NULL,
    h_pay_amount  NUMBER(7, 2) NOT NULL,
    h_inv_id      NUMBER(7) NOT NULL
);

COMMENT ON COLUMN h_payment.h_payment_id IS
    'Home insurance"s payment number';

COMMENT ON COLUMN h_payment.h_pay_date IS
    'Home insurance"s payment date';

COMMENT ON COLUMN h_payment.h_pay_method IS
    'Home insurace"s method of payment. The pamyment method shoud be one of the following:“PayPal”, “Credit”, “Debit”,
“Check”.';

COMMENT ON COLUMN h_payment.h_pay_amount IS
    'Home insurance"s payment amount';

ALTER TABLE h_payment ADD CONSTRAINT h_payment_pk PRIMARY KEY ( h_payment_id );

CREATE TABLE home (
    home_id     NUMBER(7) NOT NULL,
    pur_date    DATE NOT NULL,
    pur_value   NUMBER(10, 2) NOT NULL,
    homearea    NUMBER(6, 2) NOT NULL,
    hometype    VARCHAR2(1) NOT NULL,
    auto_fire   NUMBER(1) NOT NULL,
    sec_sys     NUMBER(1) NOT NULL,
    swim_pool   VARCHAR2(1),
    basement    NUMBER(1) NOT NULL,
    c_id        NUMBER(5) NOT NULL
);

COMMENT ON COLUMN home.home_id IS
    'Home"s ID';

COMMENT ON COLUMN home.pur_date IS
    'Home"s purchase date';

COMMENT ON COLUMN home.pur_value IS
    'Home"s purchase value';

COMMENT ON COLUMN home.homearea IS
    'Home area in sq.ft.';

COMMENT ON COLUMN home.hometype IS
```

```
   'Type of home. S,M,C,T representing Single family, Multi Family, Condominium, Town house respectively';

COMMENT ON COLUMN home.auto_fire IS
   'Indicate whether there is a Auto fire notification';

COMMENT ON COLUMN home.sec_sys IS
   'Indicate whether there is a home security system';

COMMENT ON COLUMN home.swim_pool IS
   'Swimming pool. "U", "O", "I", "M", null representing underground swimming pool, overground swimming pool, indoor swimming
pool, multiple swimming pool and no swimming pool respectively';

COMMENT ON COLUMN home.basement IS
   'Indicate whether there is a basement';

ALTER TABLE home ADD CONSTRAINT home_pk PRIMARY KEY ( home_id );

CREATE TABLE home_insurance (
   c_id          NUMBER(5) NOT NULL,
   h_start_date  DATE NOT NULL,
   h_end_date    DATE NOT NULL,
   h_premium     NUMBER(10, 2) NOT NULL,
   h_status      VARCHAR2(1) NOT NULL
);

COMMENT ON COLUMN home_insurance.c_id IS
   'Customer"s ID';

COMMENT ON COLUMN home_insurance.h_start_date IS
   'Home insurance"s start date';

COMMENT ON COLUMN home_insurance.h_end_date IS
   'Home insurance"s end date';

COMMENT ON COLUMN home_insurance.h_premium IS
   'Home insurance"s premium amount';

COMMENT ON COLUMN home_insurance.h_status IS
   'Home policy insurance status. "C" for current and "P" for expired.';

ALTER TABLE home_insurance ADD CONSTRAINT home_insurance_pk PRIMARY KEY ( c_id );

CREATE TABLE vehicle (
   vehicle_id  NUMBER(10) NOT NULL,
   vin         VARCHAR2(17) NOT NULL,
   make_year   NUMBER(4) NOT NULL,
   v_status    VARCHAR2(1) NOT NULL,
   c_id        NUMBER(5) NOT NULL
);

COMMENT ON COLUMN vehicle.vehicle_id IS
   'The unique numerical identifier of the vehice';

COMMENT ON COLUMN vehicle.vin IS
   'Vehicle identification number.';

COMMENT ON COLUMN vehicle.make_year IS
   'Vehicle"s make-model-year';

COMMENT ON COLUMN vehicle.v_status IS
   'Status of vehicle. “L”, “F”, or “O” representing “Leased”, Financed”, “and Owned’.';

ALTER TABLE vehicle ADD CONSTRAINT vehicle_pk PRIMARY KEY ( vehicle_id );

CREATE TABLE vehicle_driver (
   driver_id   NUMBER(10) NOT NULL,
   vehicle_id  NUMBER(10) NOT NULL
);

ALTER TABLE vehicle_driver ADD CONSTRAINT vehicle_driver_pk PRIMARY KEY ( driver_id,
```

```
                                                    vehicle_id );

ALTER TABLE a_invoice
    ADD CONSTRAINT a_invoice_auto_insurance_fk FOREIGN KEY ( c_id )
        REFERENCES auto_insurance ( c_id );

ALTER TABLE a_payment
    ADD CONSTRAINT a_payment_a_invoice_fk FOREIGN KEY ( a_inv_id )
        REFERENCES a_invoice ( a_inv_id );

ALTER TABLE auto_insurance
    ADD CONSTRAINT auto_insurance_customer_fk FOREIGN KEY ( c_id )
        REFERENCES customer ( c_id );

ALTER TABLE h_invoice
    ADD CONSTRAINT h_invoice_home_insurance_fk FOREIGN KEY ( c_id )
        REFERENCES home_insurance ( c_id );

ALTER TABLE h_payment
    ADD CONSTRAINT h_payment_h_invoice_fk FOREIGN KEY ( h_inv_id )
        REFERENCES h_invoice ( h_inv_id );

ALTER TABLE home
    ADD CONSTRAINT home_home_insurance_fk FOREIGN KEY ( c_id )
        REFERENCES home_insurance ( c_id );

ALTER TABLE home_insurance
    ADD CONSTRAINT home_insurance_customer_fk FOREIGN KEY ( c_id )
        REFERENCES customer ( c_id );

ALTER TABLE vehicle
    ADD CONSTRAINT vehicle_auto_insurance_fk FOREIGN KEY ( c_id )
        REFERENCES auto_insurance ( c_id );

ALTER TABLE vehicle_driver
    ADD CONSTRAINT vehicle_driver_driver_fk FOREIGN KEY ( driver_id )
        REFERENCES driver ( driver_id );

ALTER TABLE vehicle_driver
    ADD CONSTRAINT vehicle_driver_vehicle_fk FOREIGN KEY ( vehicle_id )
        REFERENCES vehicle ( vehicle_id );
```

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

```
ALTER TABLE customer
    ADD CONSTRAINT ch_inh_customer CHECK ( c_type IN (
        'A',
        'H',
        'AH'
    ) );

CREATE OR REPLACE TRIGGER arc_fkarc_1_home_insurance BEFORE
    INSERT OR UPDATE OF c_id ON home_insurance
    FOR EACH ROW
DECLARE
    d VARCHAR2(2);
BEGIN
    SELECT
        a.c_type
    INTO d
    FROM
        customer a
    WHERE
        a.c_id = :new.c_id;

    IF ( d IS NULL OR (d <> 'H' AND d <> 'AH') ) THEN
        raise_application_error(-20223, 'FK HOME_INSURANCE_CUSTOMER_FK in Table HOME_INSURANCE violates Arc constraint
on Table CUSTOMER - discriminator column C_TYPE doesn''t have value ''H''');
```

```
   END IF;

EXCEPTION
   WHEN no_data_found THEN
      NULL;
   WHEN OTHERS THEN
      RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_1_auto_insurance BEFORE
   INSERT OR UPDATE OF c_id ON auto_insurance
   FOR EACH ROW
DECLARE
   d VARCHAR2(2);
BEGIN
   SELECT
      a.c_type
   INTO d
   FROM
      customer a
   WHERE
      a.c_id = :new.c_id;

   IF ( d IS NULL OR (d <> 'A' AND d<>'AH') ) THEN
      raise_application_error(-20223, 'FK AUTO_INSURANCE_CUSTOMER_FK in Table AUTO_INSURANCE violates Arc constraint
on Table CUSTOMER - discriminator column C_TYPE doesn''t have value ''A''');
   END IF;

EXCEPTION
   WHEN no_data_found THEN
      NULL;
   WHEN OTHERS THEN
      RAISE;
END;
/

ALTER TABLE customer
   ADD CONSTRAINT c_customer_gender CHECK ( gender IN ('M', 'F') );

ALTER TABLE home_insurance
   ADD CONSTRAINT c_home_insurance_h_status CHECK ( h_status IN ( 'C', 'P') );

ALTER TABLE home
   ADD CONSTRAINT c_home_hometype CHECK ( hometype IN ('S', 'M', 'C', 'T'));

ALTER TABLE home
ADD CONSTRAINT c_home_swim_pool CHECK ( swim_pool IN ('U', 'O', 'I', 'M' ));

ALTER TABLE home
ADD CONSTRAINT c_home_auto_fire CHECK ( AUTO_FIRE IN ( 0, 1));

ALTER TABLE home
ADD CONSTRAINT c_home_sec_sys CHECK ( SEC_SYS IN ( 0, 1));

ALTER TABLE home
ADD CONSTRAINT c_home_basement CHECK ( BASEMENT IN ( 0, 1));

ALTER TABLE h_payment
  ADD CONSTRAINT c_h_payment_h_pay_method CHECK( h_pay_method IN ('Paypal', 'Credit', 'Debit', 'Check'));

ALTER TABLE auto_insurance
   ADD CONSTRAINT c_auto_insurance_a_status CHECK ( a_status IN ( 'C', 'P') );

ALTER TABLE a_payment
  ADD CONSTRAINT c_a_payment_a_pay_method CHECK( a_pay_method IN ('PayPal', 'Credit', 'Debit', 'Check'));

ALTER TABLE VEHICLE
   ADD CONSTRAINT c_vehicle_v_status CHECK ( v_status IN ('L', 'F', 'O'));
```

```
ALTER TABLE CUSTOMER
   ADD CONSTRAINT c_CUSTOMER_ID CHECK (C_ID between 10000 and 99999);

ALTER TABLE HOME
   ADD CONSTRAINT c_HOME_ID CHECK (HOME_ID between 1000000 and 9999999);

ALTER TABLE H_INVOICE
   ADD CONSTRAINT c_H_INV_ID CHECK (H_INV_ID between 1000000 and 9999999);

ALTER TABLE H_PAYMENT
ADD CONSTRAINT c_H_PAYMENT_ID CHECK (H_PAYMENT_ID between 1000000 and 9999999);

ALTER TABLE VEHICLE
   ADD CONSTRAINT c_VEHICLE_ID CHECK ( VEHICLE_ID between 1000000000 and 9999999999);

ALTER TABLE DRIVER
   ADD CONSTRAINT c_DRIVER_ID CHECK ( DRIVER_ID between 1000000000 and 9999999999);

ALTER TABLE A_INVOICE
   ADD CONSTRAINT c_A_INV_ID CHECK (A_INV_ID between 1000000 and 9999999);

ALTER TABLE A_PAYMENT
ADD CONSTRAINT c_A_PAYMENT_ID CHECK (A_PAYMENT_ID between 1000000 and 9999999);

ALTER TABLE CUSTOMER
ADD constraint C_CUSTOMER_FNAME CHECK (C_FNAME=upper (C_FNAME));

ALTER TABLE CUSTOMER
ADD constraint C_CUSTOMER_LNAME CHECK (C_LNAME=upper (C_LNAME));
```

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

```
-- CUSTOMER
INSERT INTO CUSTOMER (C_ID, C_FNAME, C_LNAME, GENDER, MARTIAL_STA
, C_TYPE, C_STREET_AD, C_CITY, C_STATE, C_ZIPCODE)
WITH names AS (
SELECT 10000, 'JOHNNY', 'WANG', 'M', 'S', 'A', '415 RED HOOK LN', 'BROOKLYN', 'NY', '11201' FROM dual UNION ALL
SELECT 10001, 'SUSAN', 'WILLIAM', 'F', 'S', 'H', '20 BENTON ST', 'NEW YORK', 'NY', '12301' FROM dual UNION ALL
SELECT 10002, 'MIKE', 'JACKSON', 'M', 'M', 'H', '434 GOVENOR ST', 'BROOKLYN', 'NY', '11204' FROM dual UNION ALL
SELECT 10003, 'JOHN', 'NICKSON', 'M', 'S', 'AH', '314 LINN ST', 'BROOKLYN', 'NY', '11231' FROM dual UNION ALL
SELECT 10004, 'JESSI', 'JOHNSON', 'F', 'S', 'A', '323 FULTON ST', 'IOWA CITY', 'IA', '52245' FROM dual UNION ALL
SELECT 10005, 'MIA', 'AVRIAL', 'F', 'M', 'H', '545 LINN ST', 'NEW YORK', 'NY', '12342' FROM dual UNION ALL
SELECT 10006, 'ROGER', 'STEVENS', 'M', 'S', 'A','98 BOARDWAY', 'NEW YORK', 'NY','23424' FROM dual UNION ALL
SELECT 10007, 'JACK', 'CLERK', 'M','M', 'A', '435 BENTON ST', 'IOWA CITY', 'IA','52240' FROM dual UNION ALL
SELECT 10008, 'DERON', 'WILLIAMS', 'M', 'S', 'A', '2313 GILBERT ST', 'CIDAR RAPIDS', 'IA', '54345' FROM dual UNION ALL
SELECT 10009, 'JACK', 'MA', 'M','M','H','21 WILL ST', 'TWIN CITY', 'IA', '53573' FROM dual UNION ALL
SELECT 10010, 'JOHNNY', 'PETERSON', 'M', 'S', 'A', '653 RED HOOK LN', 'BROOKLYN', 'NY', '11201' FROM dual UNION ALL
SELECT 10011, 'SID', 'WILLIAM', 'F', 'S', 'A', '76 BENTON ST', 'NEW YORK', 'NY', '12301' FROM dual UNION ALL
SELECT 10012, 'GEORGE', 'JACKSON', 'M', 'M', 'A', '444 GOVENOR ST', 'BROOKLYN', 'NY', '11204' FROM dual UNION ALL
SELECT 10013, 'SAM', 'NICKSON', 'M', 'S', 'A', '865 LINN ST', 'BROOKLYN', 'NY', '11231' FROM dual UNION ALL
SELECT 10014, 'JESSI', 'JOHNSON', 'F', 'S', 'H', '8766 FULTON ST', 'IOWA CITY', 'IA', '52245' FROM dual UNION ALL
SELECT 10015, 'KIO', 'AVRIAL', 'F', 'M', 'H', '1122 LINN ST', 'NEW YORK', 'NY', '12342' FROM dual UNION ALL
SELECT 10016, 'KIM', 'STEVENS', 'M', 'S', 'H','456 BOARDWAY', 'NEW YORK', 'NY','23424' FROM dual UNION ALL
SELECT 10017, 'PETER', 'CLERK', 'M','M', 'H', '56 BENTON ST', 'IOWA CITY', 'IA','52240' FROM dual UNION ALL
SELECT 10018, 'PETERSON', 'WILLIAMS', 'M', 'H', 'H', '66 GILBERT ST', 'CIDAR RAPIDS', 'IA', '54345' FROM dual
)
SELECT * FROM names;

-- HOME_INSURANCE
INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10001, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 10000, 'P');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10002, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 10000, 'P');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
```

```
VALUES (10003, TO_DATE('2019/3/14', 'YYYY/MM/DD'), TO_DATE('2021/3/14', 'YYYY/MM/DD'), 10000, 'C');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10005, TO_DATE('2019/3/14', 'YYYY/MM/DD'), TO_DATE('2021/3/14', 'YYYY/MM/DD'), 23000, 'C');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10009, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2021/3/14', 'YYYY/MM/DD'), 10000, 'C');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10014, TO_DATE('2016/1/14', 'YYYY/MM/DD'), TO_DATE('2021/1/14', 'YYYY/MM/DD'), 50000, 'C');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10015, TO_DATE('2016/7/14', 'YYYY/MM/DD'), TO_DATE('2021/7/14', 'YYYY/MM/DD'), 40000, 'C');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10016, TO_DATE('2016/5/14', 'YYYY/MM/DD'), TO_DATE('2021/5/14', 'YYYY/MM/DD'), 45000, 'C');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10017, TO_DATE('2016/12/14', 'YYYY/MM/DD'), TO_DATE('2021/12/14', 'YYYY/MM/DD'), 45000, 'C');

INSERT INTO HOME_INSURANCE(C_ID, H_START_DATE, H_END_DATE, H_PREMIUM, H_STATUS)
VALUES (10018, TO_DATE('2016/1/14', 'YYYY/MM/DD'), TO_DATE('2021/1/14', 'YYYY/MM/DD'), 50000, 'C');


-- HOME
INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000000, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'S', '1', '1', 'M', '1', 10001);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000001, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'S', '1', '1', 'M', '1', 10001);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000002, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'S', '1', '1', 'M', '1', 10002);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000003, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'M', '0', '0', 'I', '1', 10003);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000004, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'C', '0', '1', 'M', '0', 10005);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000005, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'T', '1', '0', 'O', '1', 10009);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000006, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'T', '0', '0', null, '0', 10014);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000007, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'S', '0', '1', 'U', '1', 10015);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000008, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'C', '1', '0', 'M', '1', 10016);
```

```
INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000009, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'M', '1', '0', null, '1', 10017);


INSERT INTO HOME (HOME_ID, PUR_DATE, PUR_VALUE, HOMEAREA, HOMETYPE, AUTO_FIRE, SEC_SYS, SWIM_POOL,
BASEMENT, C_ID)
VALUES (1000010, TO_DATE('2016/3/14', 'YYYY/MM/DD'), 340000, 2200, 'S', '0', '1', null, '0', 10018);



-- H_INVOICE
INSERT INTO H_INVOICE (H_INV_ID, H_INV_DATE, H_INV_DUE_DATE, H_INV_AMOUNT, C_ID)
WITH names AS (
SELECT 1000000, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 1000, 10001 FROM dual
UNION ALL
SELECT 1000001, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 2000, 10002 FROM dual
UNION ALL
SELECT 1000002, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 3000, 10003 FROM dual
UNION ALL
SELECT 1000003, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 5000, 10005 FROM dual
UNION ALL
SELECT 1000004, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 3000, 10009 FROM dual
UNION ALL
SELECT 1000005, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 3000, 10014 FROM dual
UNION ALL
SELECT 1000006, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 3000, 10015 FROM dual
UNION ALL
SELECT 1000007, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 2000, 10016 FROM dual
UNION ALL
SELECT 1000008, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 1800, 10017 FROM dual
UNION ALL
SELECT 1000009, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 2100, 10018 FROM dual
)
SELECT * FROM names;

--H_PAYMENT
INSERT INTO H_PAYMENT (H_PAYMENT_ID, H_PAY_DATE, H_PAY_METHOD, H_PAY_AMOUNT, H_INV_ID )
WITH names AS (
SELECT 2000000, TO_DATE('2016/6/18', 'YYYY/MM/DD'), 'Debit', 1000, 1000000 FROM dual UNION ALL
SELECT 2000001, TO_DATE('2016/12/14', 'YYYY/MM/DD'), 'Debit', 2000, 1000001 FROM dual UNION ALL
SELECT 2000002, TO_DATE('2016/6/25', 'YYYY/MM/DD'), 'Debit', 3000, 1000002 FROM dual UNION ALL
SELECT 2000003, TO_DATE('2016/8/14', 'YYYY/MM/DD'), 'Check', 5000, 1000003 FROM dual UNION ALL
SELECT 2000004, TO_DATE('2017/6/14', 'YYYY/MM/DD'), 'Debit', 3000, 1000004 FROM dual UNION ALL
SELECT 2000005, TO_DATE('2016/5/20', 'YYYY/MM/DD'), 'Check', 3000, 1000005 FROM dual UNION ALL
SELECT 2000006, TO_DATE('2016/6/13', 'YYYY/MM/DD'), 'Debit', 3000, 1000006 FROM dual UNION ALL
SELECT 2000007, TO_DATE('2016/6/14', 'YYYY/MM/DD'), 'Debit', 2000, 1000007 FROM dual UNION ALL
SELECT 2000008, TO_DATE('2016/6/22', 'YYYY/MM/DD'), 'Debit', 1800, 1000008 FROM dual UNION ALL
SELECT 2000009, TO_DATE('2016/9/4', 'YYYY/MM/DD'), 'Credit', 1000, 1000009 FROM dual UNION ALL
SELECT 2000010, TO_DATE('2016/8/14', 'YYYY/MM/DD'), 'Credit', 1100, 1000009 FROM dual
)
SELECT * FROM names;


-- AUTO_INSURANCE
INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10000, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 10000, 'P');

INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10003, TO_DATE('2018/3/14', 'YYYY/MM/DD'), TO_DATE('2020/3/14', 'YYYY/MM/DD'), 10000, 'P');

INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10004, TO_DATE('2019/3/14', 'YYYY/MM/DD'), TO_DATE('2021/3/14', 'YYYY/MM/DD'), 10000, 'C');

INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10006, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 10000, 'P');

INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10007, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 10000, 'P');
```

```
INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10008, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2021/3/14', 'YYYY/MM/DD'), 10000, 'C');

INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10010, TO_DATE('2019/2/14', 'YYYY/MM/DD'), TO_DATE('2021/2/14', 'YYYY/MM/DD'), 21000, 'C');

INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10011, TO_DATE('2019/3/14', 'YYYY/MM/DD'), TO_DATE('2021/3/14', 'YYYY/MM/DD'), 99000, 'C');

INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10012, TO_DATE('2019/7/14', 'YYYY/MM/DD'), TO_DATE('2021/7/14', 'YYYY/MM/DD'), 70000, 'C');

INSERT INTO AUTO_INSURANCE (C_ID, A_START_DATE, A_END_DATE, A_PREMIUM, A_STATUS)
VALUES (10013, TO_DATE('2019/8/14', 'YYYY/MM/DD'), TO_DATE('2021/8/14', 'YYYY/MM/DD'), 33000, 'C');


-- VEHICLE
INSERT INTO VEHICLE (VEHICLE_ID, VIN, MAKE_YEAR, V_STATUS, C_ID)
WITH names AS (
        SELECT 1000000000, '3F92F0SF9S2F9DSF0', 2015, 'O', 10000 FROM dual UNION ALL
        SELECT 1000000001, 'DAF98SDF8SF9S8F9D', 2015, 'O', 10000 FROM dual UNION ALL
        SELECT 1000000002, 'DSF9S8FS98FSDF8V9', 2011, 'F', 10003 FROM dual UNION ALL
        SELECT 1000000003, 'SF0SDF9FOVS9DCDSF', 2016, 'F', 10004 FROM dual UNION ALL
        SELECT 1000000004, 'DFS7A5DASD8ASDA9S', 2015, 'O', 10006 FROM dual UNION ALL
        SELECT 1000000005, 'DSSDFSDF9SDF9S89F', 2017, 'F', 10007 FROM dual UNION ALL
        SELECT 1000000006, 'FFBDS98F9SD8GS98F', 2014, 'O', 10008 FROM dual UNION ALL
        SELECT 1000000007, 'B8SF79D8V98V7S98F', 2019, 'L', 10010 FROM dual UNION ALL
        SELECT 1000000008, 'FSD9F87S7F6S8F7FS', 2018, 'O', 10011 FROM dual UNION ALL
        SELECT 1000000009, 'FS9FS8F7S6F7SF6SF', 2015, 'L', 10012 FROM dual UNION ALL
        SELECT 1000000010, 'SFSF8S7DF6SD4FFDS', 2016, 'O', 10013 FROM dual
)
SELECT * FROM names;


-- DRIVER
INSERT INTO DRIVER (DRIVER_ID, LICENSE_NO, DRIVER_FNAME, DRIVER_LNAME, DRIVER_BDATE)
WITH names AS (
        SELECT 2000000000, 'AFFA892937', 'JOHNNY', 'WANG', TO_DATE('1994/3/24', 'YYYY/MM/DD') FROM dual UNION
ALL
        SELECT 2000000001, 'JH67676676', 'DOGULS', 'RIVERS', TO_DATE('1996/7/21', 'YYYY/MM/DD') FROM dual UNION
ALL
        SELECT 2000000002, '868686868C', 'JOHN', 'NICKSON', TO_DATE('1994/6/30', 'YYYY/MM/DD') FROM dual UNION
ALL
        SELECT 2000000003, '88XCVXVXVV', 'JESSI', 'JOHNSON', TO_DATE('1996/8/21', 'YYYY/MM/DD') FROM dual UNION
ALL
        SELECT 2000000004, '57668787VC', 'ROGER', 'STEVENS', TO_DATE('1994/9/6', 'YYYY/MM/DD') FROM dual UNION
ALL
        SELECT 2000000005, '7449749GHY', 'JACK', 'CLERK', TO_DATE('1944/3/9', 'YYYY/MM/DD') FROM dual UNION ALL
        SELECT 2000000006, 'K564646645', 'DERON', 'WILLIAMS', TO_DATE('1974/3/7', 'YYYY/MM/DD') FROM dual UNION
ALL
        SELECT 2000000007, '3LJLKJOIOI', 'JOHNNY', 'PETERSON', TO_DATE('1966/3/4', 'YYYY/MM/DD') FROM dual UNION
ALL
        SELECT 2000000008, '45366363K5', 'SID', 'WILLIAM', TO_DATE('1943/3/15', 'YYYY/MM/DD') FROM dual UNION ALL
        SELECT 2000000009, '353535NMMM', 'GEORGE', 'JACKSON', TO_DATE('1987/3/23', 'YYYY/MM/DD') FROM dual
UNION ALL
        SELECT 2000000010, '353535NMMM', 'SAM', 'NICKSON', TO_DATE('1988/5/4', 'YYYY/MM/DD') FROM dual
)
SELECT * FROM names;


-- VEHICLE_DRIVER
INSERT INTO VEHICLE_DRIVER (DRIVER_ID, VEHICLE_ID)
WITH names AS (
        SELECT 2000000000, 1000000000 FROM dual UNION ALL
        SELECT 2000000000, 1000000001 FROM dual UNION ALL
        SELECT 2000000001, 1000000001 FROM dual UNION ALL
        SELECT 2000000002, 1000000002 FROM dual UNION ALL
        SELECT 2000000003, 1000000003 FROM dual UNION ALL
        SELECT 2000000004, 1000000004 FROM dual UNION ALL
        SELECT 2000000005, 1000000005 FROM dual UNION ALL
```

```
        SELECT 2000000006, 1000000006 FROM dual UNION ALL
        SELECT 2000000007, 1000000007 FROM dual UNION ALL
        SELECT 2000000008, 1000000008 FROM dual UNION ALL
        SELECT 2000000009, 1000000009 FROM dual UNION ALL
        SELECT 2000000010, 1000000010 FROM dual
)
SELECT * FROM names;

-- A_INVOICE
INSERT INTO A_INVOICE (A_INV_ID, A_INV_DATE, A_INV_DUE_DATE, A_INV_AMOUNT, C_ID)
WITH names AS (
SELECT 1000000, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 1000, 10000 FROM dual
UNION ALL
SELECT 1000001, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 2000, 10003 FROM dual
UNION ALL
SELECT 1000002, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 3000, 10004 FROM dual
UNION ALL
SELECT 1000003, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 5000, 10006 FROM dual
UNION ALL
SELECT 1000004, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 3000, 10007 FROM dual
UNION ALL
SELECT 1000005, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 3000, 10008 FROM dual
UNION ALL
SELECT 1000006, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 3000, 10010 FROM dual
UNION ALL
SELECT 1000007, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 2000, 10011 FROM dual
UNION ALL
SELECT 1000008, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 1800, 10012 FROM dual
UNION ALL
SELECT 1000009, TO_DATE('2016/3/14', 'YYYY/MM/DD'), TO_DATE('2019/3/14', 'YYYY/MM/DD'), 2100, 10013 FROM dual
)
SELECT * FROM names;

--A_PAYMENT
INSERT INTO A_PAYMENT (A_PAYMENT_ID, A_PAY_DATE, A_PAY_METHOD, A_PAY_AMOUNT, A_INV_ID )
WITH names AS (
SELECT 2000000, TO_DATE('2016/6/18', 'YYYY/MM/DD'), 'Debit', 1000, 1000000 FROM dual UNION ALL
SELECT 2000001, TO_DATE('2016/12/14', 'YYYY/MM/DD'), 'Debit', 2000, 1000001 FROM dual UNION ALL
SELECT 2000002, TO_DATE('2016/6/25', 'YYYY/MM/DD'), 'Debit', 3000, 1000002 FROM dual UNION ALL
SELECT 2000003, TO_DATE('2016/8/14', 'YYYY/MM/DD'), 'Check', 5000, 1000003 FROM dual UNION ALL
SELECT 2000004, TO_DATE('2017/6/14', 'YYYY/MM/DD'), 'Debit', 3000, 1000004 FROM dual UNION ALL
SELECT 2000005, TO_DATE('2016/5/20', 'YYYY/MM/DD'), 'Check', 3000, 1000005 FROM dual UNION ALL
SELECT 2000006, TO_DATE('2016/6/13', 'YYYY/MM/DD'), 'Debit', 3000, 1000006 FROM dual UNION ALL
SELECT 2000007, TO_DATE('2016/6/14', 'YYYY/MM/DD'), 'Debit', 2000, 1000007 FROM dual UNION ALL
SELECT 2000008, TO_DATE('2016/6/22', 'YYYY/MM/DD'), 'Debit', 1800, 1000008 FROM dual UNION ALL
SELECT 2000009, TO_DATE('2016/9/4', 'YYYY/MM/DD'), 'Credit', 1000, 1000009 FROM dual UNION ALL
SELECT 2000010, TO_DATE('2016/8/14', 'YYYY/MM/DD'), 'Credit', 1100, 1000009 FROM dual
)
SELECT * FROM names;
```

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

```
SELECT
   (SELECT COUNT(*)  FROM CUSTOMER) as "CUSTOMER",
   (SELECT COUNT(*)  FROM HOME_INSURANCE) as "HOME_INSURANCE",
   (SELECT COUNT(*)  FROM AUTO_INSURANCE) as "HOME_INSURANCE",
   (SELECT COUNT(*)  FROM A_INVOICE) as "A_INVOICE",
   (SELECT COUNT(*)  FROM H_INVOICE) as "H_INVOICE",
   (SELECT COUNT(*)  FROM A_PAYMENT) as "A_PAYMENT",
   (SELECT COUNT(*)  FROM A_PAYMENT) as "A_PAYMENT",
   (SELECT COUNT(*)  FROM HOME) as "HOME",
   (SELECT COUNT(*)  FROM VEHICLE) as "VEHICLE",
   (SELECT COUNT(*)  FROM DRIVER) as "DRIVER",
   (SELECT COUNT(*)  FROM VEHICLE_DRIVER) as "VEHICLE_DRIVER"
from dual
```

CUSTOMER	HOME_INSURANCE	HOME_INSURANCE	A_INVOICE	H_INVOICE	A_PAYMENT	A_PAYMENT	HOME	VEHICLE	DRIVER	VEHICLE_DRIVER
19	10	10	10	10	11	11	11	11	11	12

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

```
SELECT a.table_name
    , a.column_name
    , a.data_type
    , nvl(a.data_precision, a.data_length) as "DATA_SIZE"
    , REPLACE(REPLACE(REPLACE(a.nullable,'N','M'),'Y','OPTIONAL'),'M','MANDATORY') AS "MANDATORY/OPTIONAL"
    , cc.constraint_name
    , uc.search_condition
    , co.comments
  FROM user_tab_columns a
      LEFT JOIN user_cons_columns cc
         ON (cc.table_name = a.table_name AND
           cc.column_name = a.column_name)
      LEFT JOIN user_constraints uc
         ON (a.table_name = uc.table_name AND
             uc.constraint_name = cc.constraint_name)
      LEFT JOIN user_col_comments co
         ON (a.table_name = co.table_name AND
             a.column_name = co.column_name)
   ORDER BY 1;
```

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_SIZE	MANDATORY/OPTIONAL	CONSTRAINT_NAME	SEARCH_CONDITION	COMMENTS
AUTO_INSURANCE	C_ID	NUMBER	5	MANDATORY	SYS_C0027890097	C_ID" IS NOT NULL	Customer's ID
AUTO_INSURANCE	A_START_DATE	DATE	7	MANDATORY	SYS_C0027890098	A_START_DATE" IS NOT NULL	Auto insurance's start date
AUTO_INSURANCE	A_END_DATE	DATE	7	MANDATORY	SYS_C0027890099	A_END_DATE" IS NOT NULL	Auto insurance's end date
AUTO_INSURANCE	A_PREMIUM	NUMBER	10	MANDATORY	SYS_C0027890100	A_PREMIUM" IS NOT NULL	Auto insurance's premium amount
AUTO_INSURANCE	C_ID	NUMBER	5	MANDATORY	AUTO_INSURANCE_CUSTOMER_FK	-	Customer's ID
AUTO_INSURANCE	C_ID	NUMBER	5	MANDATORY	AUTO_INSURANCE_PK	-	Customer's ID
AUTO_INSURANCE	A_STATUS	VARCHAR2	1	MANDATORY	C_AUTO_INSURANCE_A_STATUS	a_status IN ('C', 'P')	Auto policy insurance status. "C" for current and "P" for expired.
AUTO_INSURANCE	A_STATUS	VARCHAR2	1	MANDATORY	SYS_C0027890101	A_STATUS" IS NOT NULL	Auto policy insurance status. "C" for current and "P" for expired.
A_INVOICE	A_INV_ID	NUMBER	7	MANDATORY	A_INVOICE_PK	-	Auto insurance's invoice number
A_INVOICE	C_ID	NUMBER	5	MANDATORY	SYS_C0027890081	C_ID" IS NOT NULL	-
A_INVOICE	A_INV_AMOUNT	NUMBER	7	MANDATORY	SYS_C0027890080	A_INV_AMOUNT" IS NOT NULL	Auto insurance's invoice amount
A_INVOICE	A_INV_DUE_DATE	DATE	7	MANDATORY	SYS_C0027890079	A_INV_DUE_DATE" IS NOT NULL	Auto insurance's payment due date
A_INVOICE	A_INV_DATE	DATE	7	MANDATORY	SYS_C0027890078	A_INV_DATE" IS NOT NULL	Auto insurance's invoice date
A_INVOICE	A_INV_ID	NUMBER	7	MANDATORY	SYS_C0027890077	A_INV_ID" IS NOT NULL	Auto insurance's invoice number
A_INVOICE	A_INV_ID	NUMBER	7	MANDATORY	C_A_INV_ID	A_INV_ID between 1000000 and 9999999	Auto insurance's invoice number
A_INVOICE	C_ID	NUMBER	5	MANDATORY	A_INVOICE_AUTO_INSURANCE_FK	-	-
A_PAYMENT	A_PAY_AMOUNT	NUMBER	7	MANDATORY	SYS_C0027890087	A_PAY_AMOUNT" IS NOT NULL	Auto insurance's payment amount
A_PAYMENT	A_PAY_METHOD	VARCHAR2	6	MANDATORY	SYS_C0027890086	A_PAY_METHOD" IS NOT NULL	Auto insurance's method of payment. The pamyment method shoud be one of the following:"PayPal", "Credit", "Debit", "Check".
A_PAYMENT	A_PAY_DATE	DATE	7	MANDATORY	SYS_C0027890085	A_PAY_DATE" IS NOT NULL	Auto insurance's payment date
A_PAYMENT	A_PAYMENT_ID	NUMBER	7	MANDATORY	SYS_C0027890084	A_PAYMENT_ID" IS NOT NULL	Auto insurance's payment number
A_PAYMENT	A_INV_ID	NUMBER	7	MANDATORY	SYS_C0027890088	A_INV_ID" IS NOT NULL	-
A_PAYMENT	A_INV_ID	NUMBER	7	MANDATORY	A_PAYMENT_A_INVOICE_FK	-	-
A_PAYMENT	A_PAYMENT_ID	NUMBER	7	MANDATORY	C_A_PAYMENT_ID	A_PAYMENT_ID between 1000000 and 9999999	Auto insurance's payment number
A_PAYMENT	A_PAYMENT_ID	NUMBER	7	MANDATORY	A_PAYMENT_PK	-	Auto insurance's payment number
A_PAYMENT	A_PAY_METHOD	VARCHAR2	6	MANDATORY	C_A_PAYMENT_A_PAY_METHOD	a_pay_method IN ('PayPal', 'Credit', 'Debit', 'Check')	Auto insurance's method of payment. The pamyment method shoud be one of the following:"PayPal", "Credit", "Debit", "Check".
CUSTOMER	C_ID	NUMBER	5	MANDATORY	SYS_C0027890103	C_ID" IS NOT NULL	Customer's ID
CUSTOMER	C_FNAME	VARCHAR2	30	MANDATORY	SYS_C0027890104	C_FNAME" IS NOT NULL	Customer's first name
CUSTOMER	C_LNAME	VARCHAR2	30	MANDATORY	SYS_C0027890105	C_LNAME" IS NOT NULL	Customer's last name
CUSTOMER	MARTIAL_STA	VARCHAR2	1	MANDATORY	SYS_C0027890106	MARTIAL_STA" IS NOT NULL	Customer's martial status. "M", "S", or "W", representing "Married", "Single", and "Widow/Widower" respectively.
CUSTOMER	C_TYPE	VARCHAR2	14	MANDATORY	SYS_C0027890107	C_TYPE" IS NOT NULL	Customer type
CUSTOMER	C_STREET_AD	VARCHAR2	30	MANDATORY	SYS_C0027890108	C_STREET_AD" IS NOT NULL	Customer's street address
CUSTOMER	C_CITY	VARCHAR2	30	MANDATORY	SYS_C0027890109	C_CITY" IS NOT NULL	The city of the customer's address
CUSTOMER	C_STATE	VARCHAR2	2	MANDATORY	SYS_C0027890110	C_STATE" IS NOT NULL	The state abbr of the customer's address
CUSTOMER	C_ZIPCODE	VARCHAR2	5	MANDATORY	SYS_C0027890111	C_ZIPCODE" IS NOT NULL	The 5-digit zip code of the customer's address
CUSTOMER	C_ID	NUMBER	5	MANDATORY	CUSTOMER_PK	-	Customer's ID
CUSTOMER	C_TYPE	VARCHAR2	14	MANDATORY	CH_INH_CUSTOMER	c_type IN ('A', 'H', 'AH')	Customer type
CUSTOMER	GENDER	VARCHAR2	1	OPTIONAL	C_CUSTOMER_GENDER	gender IN ('M', 'F')	Customer's gender. "M", or "F" representing "Male" or "Female" respectively.
CUSTOMER	C_ID	NUMBER	5	MANDATORY	C_CUSTOMER_ID	C_ID between 10000 and 99999	Customer's ID
CUSTOMER	C_FNAME	VARCHAR2	30	MANDATORY	C_CUSTOMER_FNAME	C_FNAME=upper (C_FNAME)	Customer's first name
CUSTOMER	C_LNAME	VARCHAR2	30	MANDATORY	C_CUSTOMER_LNAME	C_LNAME=upper (C_LNAME)	Customer's last name
DRIVER	DRIVER_BDATE	DATE	7	MANDATORY	SYS_C0027890119	DRIVER_BDATE" IS NOT NULL	Driver's birthdate
DRIVER	DRIVER_ID	NUMBER	10	MANDATORY	DRIVER_PK	-	Driver's unique identifier
DRIVER	DRIVER_ID	NUMBER	10	MANDATORY	SYS_C0027890115	DRIVER_ID" IS NOT NULL	Driver's unique identifier
DRIVER	DRIVER_ID	NUMBER	10	MANDATORY	C_DRIVER_ID	DRIVER_ID between 1000000000 and 9999999999	Driver's unique identifier
DRIVER	LICENSE_NO	VARCHAR2	30	MANDATORY	SYS_C0027890116	LICENSE_NO" IS NOT NULL	Driver's license number
DRIVER	DRIVER_FNAME	VARCHAR2	30	MANDATORY	SYS_C0027890117	DRIVER_FNAME" IS NOT NULL	Driver's first name
DRIVER	DRIVER_LNAME	VARCHAR2	30	MANDATORY	SYS_C0027890118	DRIVER_LNAME" IS NOT NULL	Driver's last name
HOME	BASEMENT	NUMBER	1	MANDATORY	SYS_C0027890198	BASEMENT" IS NOT NULL	Indicate whether there is a basement
HOME	SEC_SYS	NUMBER	1	MANDATORY	SYS_C0027890197	SEC_SYS" IS NOT NULL	Indicate whether there is a home security system
HOME	HOME_ID	NUMBER	7	MANDATORY	C_HOME_ID	HOME_ID between 1000000 and 9999999	Home's ID

h)

Write SQL queries using each of following:

Q1. Table joins with at least 3 tables in join

Select query:

```
SELECT a.A_PAYMENT_ID, a.A_PAY_AMOUNT, b.A_INV_DATE, c.A_STATUS
FROM A_PAYMENT a JOIN A_INVOICE b ON a.A_INV_ID = b.A_INV_ID JOIN AUTO_INSURANCE c ON b.C_ID = c.C_ID;
```

Result of the query:

A_PAYMENT_ID	A_PAY_AMOUNT	A_INV_DATE	A_STATUS
2000000	1000	14-MAR-16	P
2000001	2000	14-MAR-16	P
2000002	3000	14-MAR-16	C
2000003	5000	14-MAR-16	P
2000004	3000	14-MAR-16	P
2000005	3000	14-MAR-16	C
2000006	3000	14-MAR-16	C
2000007	2000	14-MAR-16	C
2000008	1800	14-MAR-16	C
2000009	1000	14-MAR-16	C
2000010	1100	14-MAR-16	C

Information intended to achieve:

List the payment id of all auto insurance payment records alone with the payment amount, the start date of the invoice associated with this payment record and the status of the insurance policy associated with this invoice.

Q2. Multi-row subquery

Select query:

```
SELECT C_ID, A_PREMIUM
FROM AUTO_INSURANCE
WHERE A_PREMIUM > ANY (SELECT AVG(a.A_PREMIUM) FROM AUTO_INSURANCE a JOIN CUSTOMER b ON a.C_ID = b.C_ID
GROUP BY b.C_CITY);
```

Result of the query:

C_ID	A_PREMIUM
10010	21000
10011	99000
10012	70000
10013	33000

Information intended to achieve:

List the customer ID and premium amount of each customer who has a higher premium amount than the average premium amount of the customers live in that city.

Q3. Co-related subquery

Select query:

```
SELECT a.C_ID, a.H_PREMIUM
FROM HOME_INSURANCE a
WHERE a.H_PREMIUM < (SELECT AVG(b.H_PREMIUM) FROM HOME_INSURANCE b WHERE to_char(a.H_START_DATE, 'yyyy') =
to_char(b.H_START_DATE,'yyyy'));
```

Result of the query:

C_ID	H_PREMIUM
10001	10000
10002	10000
10003	10000
10009	10000

Information intended to achieve:
List customer Identifier and his/her home insurance premium amount whose home premium amount is lower than the average premium amount of all home insurance start in that year.

Q4. SET operator query

Select query:

```
SELECT UNIQUE b.C_ID,b.C_FNAME, b.C_LNAME
FROM HOME a JOIN CUSTOMER b ON a.C_ID = b.C_ID
WHERE SEC_SYS = '1'
UNION
SELECT UNIQUE b.C_ID,b.C_FNAME, b.C_LNAME
FROM HOME a JOIN CUSTOMER b ON a.C_ID = b.C_ID
WHERE AUTO_FIRE = '1';
```

Result of the query:

C_ID	C_FNAME	C_LNAME
10001	SUSAN	WILLIAM
10002	MIKE	JACKSON
10005	MIA	AVRIAL
10009	JACK	MA
10015	KIO	AVRIAL
10016	KIM	STEVENS
10017	PETER	CLERK
10018	PETERSON	WILLIAMS

Information intended to achieve:
List customer ID and customer's name who has a home with security system or basement.

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

Select query:

```
WITH CITY_AVERAGE AS
   (SELECT a.C_CITY, AVG(b.H_PREMIUM) city_ave
    FROM CUSTOMER a JOIN HOME_INSURANCE b ON a.C_ID = b.C_ID
    GROUP BY a.C_CITY)
SELECT c.C_ID,c.C_FNAME, c.C_LNAME, d.city_ave
FROM CUSTOMER c JOIN CITY_AVERAGE d ON c.C_CITY = d.C_CITY;
```

Result of the query:

C_ID	C_FNAME	C_LNAME	CITY_AVE
10000	JOHNNY	WANG	10000
10001	SUSAN	WILLIAM	29500
10002	MIKE	JACKSON	10000
10003	JOHN	NICKSON	10000
10004	JESSI	JOHNSON	47500
10005	MIA	AVRIAL	29500
10006	ROGER	STEVENS	29500
10007	JACK	CLERK	47500
10008	DERON	WILLIAMS	50000
10009	JACK	MA	10000
10010	JOHNNY	PETERSON	10000
10011	SID	WILLIAM	29500
10012	GEORGE	JACKSON	10000
10013	SAM	NICKSON	10000
10014	JESSI	JOHNSON	47500
10015	KIO	AVRIAL	29500
10016	KIM	STEVENS	29500
10017	PETER	CLERK	47500
10018	PETERSON	WILLIAMS	50000

Information intended to achieve:
List each customer's ID and name and the average home insurance premium of the city he/she live in.