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

PROJECT PART I

Qianyong	Tang	N11366182	qt344
Tianyu	Gu	N13364566	tg1899

April 10, 2020

Contents

1. Description

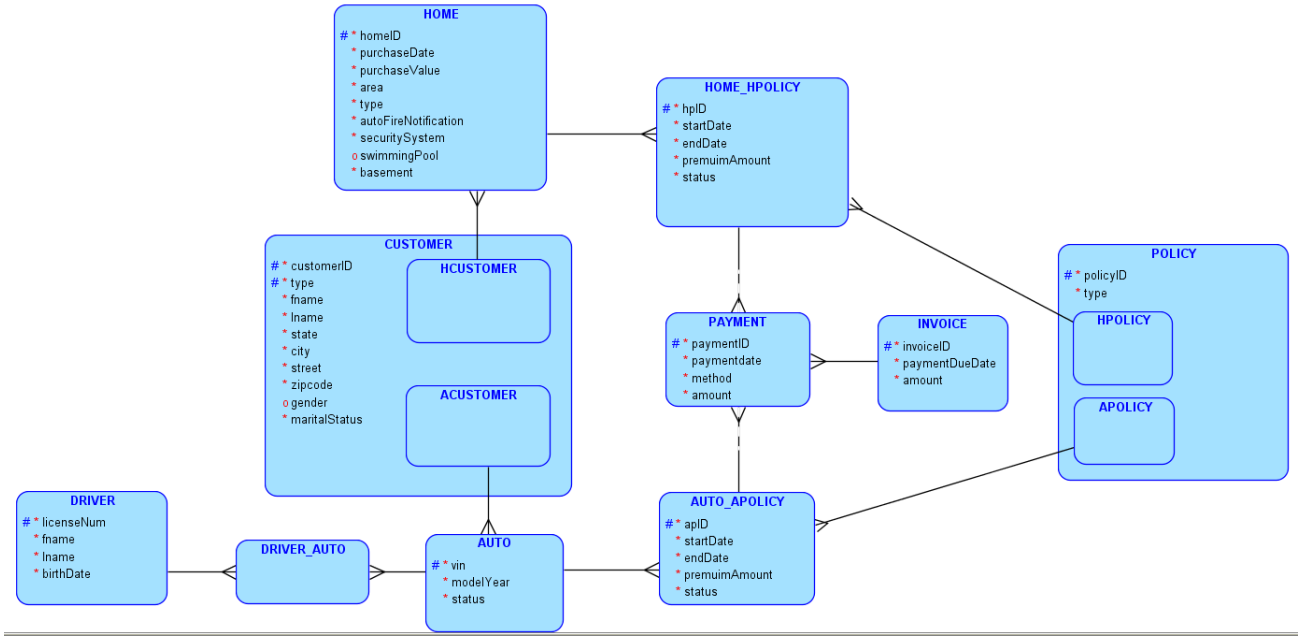
This report describes the database design created for the company We Do Secure (WDS), used to help WDS deal with recently expanded operation on offering Auto and Home insurance to their customers. The database provides the function to store information required by WDS of customer, policy, home, auto, auto driver, customer payment and invoice. Moreover, the database design indicates the relationship between these entities clearly. The detailed design and assumptions are explained in the following pages. The robustness of the database is proved by generating sample data and several practice SQL queries with flexibility. Our model can hold complex situations such as multiple homes and autos for one customer. All the previous records of a customer are stored. Our model also supports multiple drivers for multiple autos. There are more features for our model. For further information, the corresponding codes and results are shown in next sections.

2. Assumptions

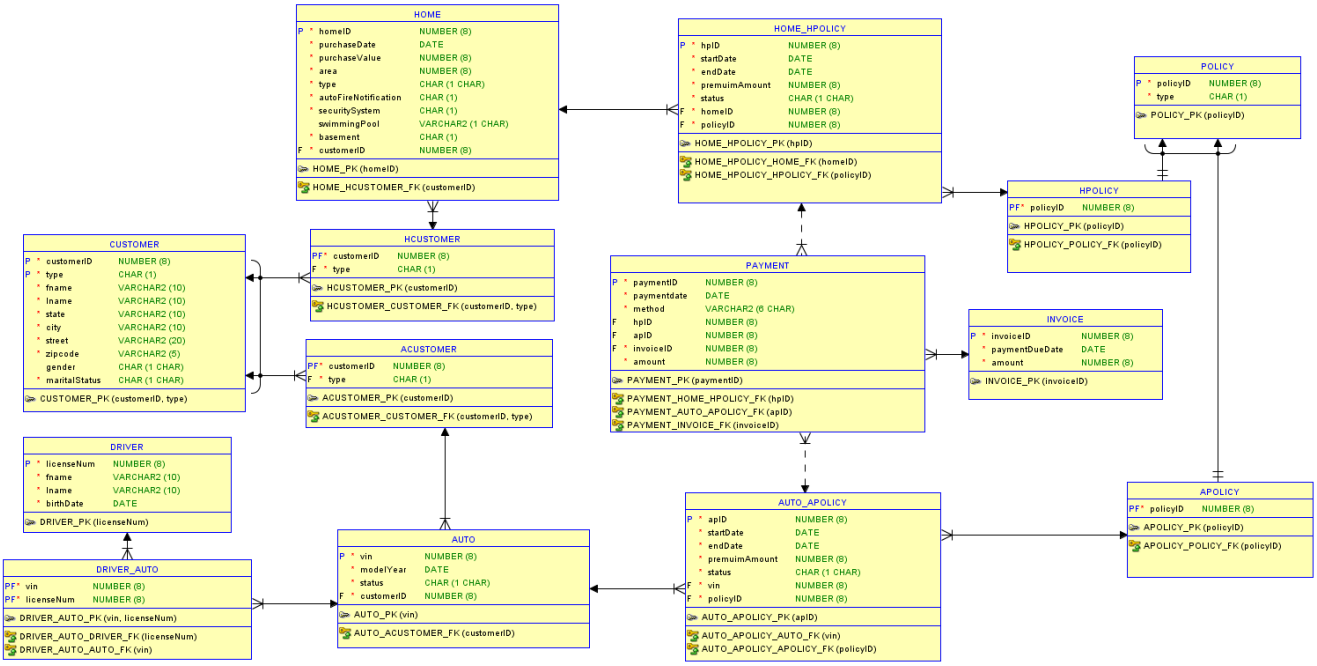
1. A customer has only two subtypes, auto-customer(acustomer) and home-customer (hcustomer).
2. A customer has to be either acustomer or hcustomer, or both.
3. An hcustomer can have multiple homes, and each home can have multiple home insurance policies.
4. An acustomer can have multiple autos, and each auto can have multiple auto insurance policies.
5. An auto may can have multiple drivers and a driver can drive multiple cars.
6. A driver is distinct from a customer in the database.
7. An hpolicy for a home can have multiple payments, i.e., installments.
8. An apolicy for an auto can have multiple payments, i.e., installments.
9. An hpolicy for a home only generate one invoice.
10. An apolicy for an auto only generate one invoice.
11. Multiple payments for a certain hpolicy for a home correspond to one invoice.
12. Multiple payments for a certain apolicy for an auto correspond to one invoice.
13. Each payment corresponds to either home policy or auto policy but cannot be both.
14. A policy has only two subtypes, auto-policy(apolicy) and home-policy(hpolicy).
15. A policy has to be either apolicy or hpolicy, but cannot be both.

3. Results

3.1 Problem A



3.2 Problem B



3.3 Problem C

```
--   Generated by Oracle SQL Developer Data Modeler 19.4.0.350.1424
--   at:        2020-04-09 19:22:55 EDT
--   site:      Oracle Database 11g
--   type:      Oracle Database 11g



CREATE TABLE acustomer (
    customerid  NUMBER(8) NOT NULL,
    type        CHAR(1) NOT NULL
);

COMMENT ON COLUMN acustomer.customerid IS
    'the unique ID for customer';

COMMENT ON COLUMN acustomer.type IS
    'to show whether the customer is Automobile Insurance customer or
      Home Insurance customer or both, can only be "A" or "H"';

ALTER TABLE acustomer ADD CONSTRAINT acustomer_pk PRIMARY KEY (
   customerid );
```

```
CREATE TABLE apolicy (
    policyid NUMBER(8) NOT NULL
);

COMMENT ON COLUMN apolicy.policyid IS
    'the unique ID of policy';

ALTER TABLE apolicy ADD CONSTRAINT apolicy_pk PRIMARY KEY ( policyid );

CREATE TABLE auto (
    vin          NUMBER(8) NOT NULL,
    modelyear    DATE NOT NULL,
    status       CHAR(1 CHAR) NOT NULL,
    customerid   NUMBER(8) NOT NULL
);

COMMENT ON COLUMN auto.vin IS
    'unqiue vehicle VIN (vehicle identification number)';

COMMENT ON COLUMN auto.modelyear IS
    'vehicle make-model year';

COMMENT ON COLUMN auto.status IS
    'show the status of the vehicle, can be one of L, F, or O
       representing "Leased", "Financed", and "Owned"
';

COMMENT ON COLUMN auto.customerid IS
    'the unique ID for customer';

ALTER TABLE auto ADD CONSTRAINT auto_pk PRIMARY KEY ( vin );

CREATE TABLE auto_apolicy (
    apid            NUMBER(8) NOT NULL,
    startdate       DATE NOT NULL,
    enddate         DATE,
    premuimamount   NUMBER(8) NOT NULL,
    status          CHAR(1 CHAR) NOT NULL,
    vin             NUMBER(8) NOT NULL,
    policyid        NUMBER(8) NOT NULL
);

COMMENT ON COLUMN auto_apolicy.apid IS
    'the unique ID standing for the relationship between a specific auto
       and a specific auto policy';

COMMENT ON COLUMN auto_apolicy.startdate IS
    'auto insurance policy start date';
```

```
COMMENT ON COLUMN auto_apolicy.enddate IS
    'auto insurance policy end date';

COMMENT ON COLUMN auto_apolicy.premuimamount IS
    'auto insurance premium amount';

COMMENT ON COLUMN auto_apolicy.status IS
    'show auto policy insurance status. If auto insurance policy term is
       current, status column should have value "C", and if it is expired
       , it should have value "P". ';

COMMENT ON COLUMN auto_apolicy.vin IS
    'unqiue vehicle VIN (vehicle identification number)';

COMMENT ON COLUMN auto_apolicy.policyid IS
    'the unique ID of policy';

ALTER TABLE auto_apolicy ADD CONSTRAINT auto_apolicy_pk PRIMARY KEY (
   apid );

CREATE TABLE customer (
    customerid      NUMBER(8) NOT NULL,
    type            CHAR(1) NOT NULL,
    fname           VARCHAR2(10) NOT NULL,
    lname           VARCHAR2(10) NOT NULL,
    state           VARCHAR2(10) NOT NULL,
    city            VARCHAR2(10) NOT NULL,
    street          VARCHAR2(20) NOT NULL,
    zipcode         VARCHAR2(5) NOT NULL,
    gender          CHAR(1 CHAR),
    maritalstatus   CHAR(1 CHAR) NOT NULL
);

ALTER TABLE customer
    ADD CONSTRAINT ch_inh_customer CHECK ( type IN (
        'A',
        'H'
    ) );

COMMENT ON COLUMN customer.customerid IS
    'the unique ID for customer';

COMMENT ON COLUMN customer.type IS
    'to show whether the customer is Automobile Insurance customer or
       Home Insurance customer or both, can only be "A" or "H"';

COMMENT ON COLUMN customer.fname IS
```

```
    'the first name of customer, must be in upper case';

COMMENT ON COLUMN customer.lname IS
    'the last name of customer, must be in upper case';

COMMENT ON COLUMN customer.state IS
    'the state where customer lives';

COMMENT ON COLUMN customer.city IS
    'the city where customer lives';

COMMENT ON COLUMN customer.street IS
    'the street where customer lives';

COMMENT ON COLUMN customer.zipcode IS
    'the zipcode that where customer lives uses';

COMMENT ON COLUMN customer.gender IS
    'show whether customer is male or female, can only be "M" or "F",
       customer may choose not to provide gender data';

COMMENT ON COLUMN customer.maritalstatus IS
    'show marital status of customer, must be either "M", "S", or "W",
       representing "Married", "Single", and "Widow/Widower" respectively
       ';

ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY ( customerid,
                                                              type );

CREATE TABLE driver (
    licensenum  NUMBER(8) NOT NULL,
    fname       VARCHAR2(10) NOT NULL,
    lname       VARCHAR2(10) NOT NULL,
    birthdate   DATE NOT NULL
);

COMMENT ON COLUMN driver.licensenum IS
    'the unique license number of driver';

COMMENT ON COLUMN driver.fname IS
    'the first name of driver, must be in upper case';

COMMENT ON COLUMN driver.lname IS
    'the last name of driver, must be in upper case';

COMMENT ON COLUMN driver.birthdate IS
    'the birthdate of driver';
```

```
ALTER TABLE driver ADD CONSTRAINT driver_pk PRIMARY KEY ( licensenum );

CREATE TABLE driver_auto (
    vin         NUMBER(8) NOT NULL,
    licensenum  NUMBER(8) NOT NULL
);

COMMENT ON COLUMN driver_auto.vin IS
    'unqiue vehicle VIN (vehicle identification number)';

COMMENT ON COLUMN driver_auto.licensenum IS
    'the unique license number of driver';

ALTER TABLE driver_auto ADD CONSTRAINT driver_auto_pk PRIMARY KEY ( vin,
                                                                   licensenum
                                                                       )
                                                                      ;

CREATE TABLE hcustomer (
    customerid  NUMBER(8) NOT NULL,
    type        CHAR(1) NOT NULL
);

COMMENT ON COLUMN hcustomer.customerid IS
    'the unique ID for customer';

COMMENT ON COLUMN hcustomer.type IS
    'to show whether the customer is Automobile Insurance customer or
       Home Insurance customer or both, can only be "A" or "H"';

ALTER TABLE hcustomer ADD CONSTRAINT hcustomer_pk PRIMARY KEY (
   customerid );

CREATE TABLE home (
    homeid                 NUMBER(8) NOT NULL,
    purchasedate           DATE NOT NULL,
    purchasevalue          NUMBER(8) NOT NULL,
    area                   NUMBER(8) NOT NULL,
    type                   CHAR(1 CHAR) NOT NULL,
    autofirenotification   CHAR(1) NOT NULL,
    securitysystem         CHAR(1) NOT NULL,
    swimmingpool           VARCHAR2(1 CHAR),
    basement               CHAR(1) NOT NULL,
    customerid             NUMBER(8) NOT NULL
);

COMMENT ON COLUMN home.homeid IS
    'the unique home ID';
```

```
COMMENT ON COLUMN home.purchasedate IS
    'home purchase date';

COMMENT ON COLUMN home.purchasevalue IS
    'home purchase value';

COMMENT ON COLUMN home.area IS
    'home area in Sq. Ft';

COMMENT ON COLUMN home.type IS
    'show the type of home, can be S,M,C,T representing Single family,
       Multi Family, Condominium, Town house respectively';

COMMENT ON COLUMN home.autofirenotification IS
    'show whether there is automatic fire notification to the fire
       department. 1 for there is and 0 for there is not.';

COMMENT ON COLUMN home.securitysystem IS
    'show whether there is the home security system. 1 for there is and 0
        for there is not.';

COMMENT ON COLUMN home.swimmingpool IS
    'show which type of the swimming pool and whether there is a swimming
        pool. Not mendatory attribute, and blank means null showing there
        is no swimming pool. Can be U,O,I,M, representing Underground
       swimming pool, Overground swimming pool, Indoor swimming pool and
       Multiple swimming pool respectively.';

COMMENT ON COLUMN home.basement IS
    'show whether there is basement. 1 for there is and 0 for there is
       not.';

COMMENT ON COLUMN home.customerid IS
    'the unique ID for customer';

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

CREATE TABLE home_hpolicy (
    hpid            NUMBER(8) NOT NULL,
    startdate       DATE NOT NULL,
    enddate         DATE NOT NULL,
    premuimamount   NUMBER(8) NOT NULL,
    status          CHAR(1 CHAR) NOT NULL,
    homeid          NUMBER(8) NOT NULL,
    policyid        NUMBER(8) NOT NULL
);
```

```
COMMENT ON COLUMN home_hpolicy.hpid IS
    'the unique ID standing for the relationship between a specific home
       and a specific home policy';

COMMENT ON COLUMN home_hpolicy.startdate IS
    ' home insurance policy start date';

COMMENT ON COLUMN home_hpolicy.enddate IS
    'home insurance policy end date';

COMMENT ON COLUMN home_hpolicy.premuimamount IS
    'home insurance premium amount';

COMMENT ON COLUMN home_hpolicy.status IS
    'show home policy insurance status. If home insurance policy term is
       current, status column should have value "C", and if it is expired
       , it should have value "P". ';

COMMENT ON COLUMN home_hpolicy.homeid IS
    'the unique home ID';

COMMENT ON COLUMN home_hpolicy.policyid IS
    'the unique ID of policy';

ALTER TABLE home_hpolicy ADD CONSTRAINT home_hpolicy_pk PRIMARY KEY (
   hpid );

CREATE TABLE hpolicy (
    policyid NUMBER(8) NOT NULL
);

COMMENT ON COLUMN hpolicy.policyid IS
    'the unique ID of policy';

ALTER TABLE hpolicy ADD CONSTRAINT hpolicy_pk PRIMARY KEY ( policyid );

CREATE TABLE invoice (
    invoiceid       NUMBER(8) NOT NULL,
    paymentduedate  DATE NOT NULL,
    amount          NUMBER(8) NOT NULL
);

COMMENT ON COLUMN invoice.invoiceid IS
    'the unique of invoice';

COMMENT ON COLUMN invoice.paymentduedate IS
    'show payment due date';
```

```
COMMENT ON COLUMN invoice.amount IS
    'show invoice amount';

ALTER TABLE invoice ADD CONSTRAINT invoice_pk PRIMARY KEY ( invoiceid );

CREATE TABLE payment (
    paymentid  NUMBER(8) NOT NULL,
    "date"     DATE NOT NULL,
    method     VARCHAR2(6 CHAR) NOT NULL,
    hpid       NUMBER(8),
    apid       NUMBER(8),
    invoiceid  NUMBER(8) NOT NULL
);

COMMENT ON COLUMN payment.paymentid IS
    'the unique ID for payment';

COMMENT ON COLUMN payment."date" IS
    'payment date ';

COMMENT ON COLUMN payment.method IS
    'show method of payment, can be either as "PayPal", "Credit", "Debit
       ", "Check" ';

COMMENT ON COLUMN payment.hpid IS
    'the unique ID standing for the relationship between a specific home
       and a specific home policy';

COMMENT ON COLUMN payment.apid IS
    'the unique ID standing for the relationship between a specific auto
       and a specific auto policy';

ALTER TABLE payment ADD CONSTRAINT payment_pk PRIMARY KEY ( paymentid );

CREATE TABLE policy (
    policyid  NUMBER(8) NOT NULL,
    type      CHAR(1) NOT NULL
);

ALTER TABLE policy
    ADD CONSTRAINT ch_inh_policy CHECK ( type IN (
        'A',
        'H'
    ) );

COMMENT ON COLUMN policy.policyid IS
    'the unique ID of policy';
```

```
COMMENT ON COLUMN policy.type IS
    'show the type of policy, a policy can either be Home Insurance
       policy or Auto Insurance policy, but cannot be both, can be "H" or
        "A"';

ALTER TABLE policy ADD CONSTRAINT policy_pk PRIMARY KEY ( policyid );

ALTER TABLE acustomer
    ADD CONSTRAINT acustomer_customer_fk FOREIGN KEY ( customerid,
                                                       type )
        REFERENCES customer ( customerid,
                              type );

ALTER TABLE apolicy
    ADD CONSTRAINT apolicy_policy_fk FOREIGN KEY ( policyid )
        REFERENCES policy ( policyid );

ALTER TABLE auto
    ADD CONSTRAINT auto_acustomer_fk FOREIGN KEY ( customerid )
        REFERENCES acustomer ( customerid );

ALTER TABLE auto_apolicy
    ADD CONSTRAINT auto_apolicy_apolicy_fk FOREIGN KEY ( policyid )
        REFERENCES apolicy ( policyid );

ALTER TABLE auto_apolicy
    ADD CONSTRAINT auto_apolicy_auto_fk FOREIGN KEY ( vin )
        REFERENCES auto ( vin );

ALTER TABLE driver_auto
    ADD CONSTRAINT driver_auto_auto_fk FOREIGN KEY ( vin )
        REFERENCES auto ( vin );

ALTER TABLE driver_auto
    ADD CONSTRAINT driver_auto_driver_fk FOREIGN KEY ( licensenum )
        REFERENCES driver ( licensenum );

ALTER TABLE hcustomer
    ADD CONSTRAINT hcustomer_customer_fk FOREIGN KEY ( customerid,
                                                       type )
        REFERENCES customer ( customerid,
                              type );

ALTER TABLE home
    ADD CONSTRAINT home_hcustomer_fk FOREIGN KEY ( customerid )
        REFERENCES hcustomer ( customerid );

ALTER TABLE home_hpolicy
```

```
    ADD CONSTRAINT home_hpolicy_home_fk FOREIGN KEY ( homeid )
        REFERENCES home ( homeid );

ALTER TABLE home_hpolicy
    ADD CONSTRAINT home_hpolicy_hpolicy_fk FOREIGN KEY ( policyid )
        REFERENCES hpolicy ( policyid );

ALTER TABLE hpolicy
    ADD CONSTRAINT hpolicy_policy_fk FOREIGN KEY ( policyid )
        REFERENCES policy ( policyid );

ALTER TABLE payment
    ADD CONSTRAINT payment_auto_apolicy_fk FOREIGN KEY ( apid )
        REFERENCES auto_apolicy ( apid );

ALTER TABLE payment
    ADD CONSTRAINT payment_home_hpolicy_fk FOREIGN KEY ( hpid )
        REFERENCES home_hpolicy ( hpid );

ALTER TABLE payment
    ADD CONSTRAINT payment_invoice_fk FOREIGN KEY ( invoiceid )
        REFERENCES invoice ( invoiceid );

CREATE OR REPLACE TRIGGER arc_fkarc_21_hcustomer BEFORE
    INSERT OR UPDATE OF customerid, type ON hcustomer
    FOR EACH ROW
DECLARE
    d CHAR(1);
BEGIN
    SELECT
        a.type
    INTO d
    FROM
        customer a
    WHERE
            a.customerid = :new.customerid
        AND a.type = :new.type;

    IF ( d IS NULL OR d <> 'H' ) THEN
        raise_application_error(-20223, 'FK HCUSTOMER_CUSTOMER_FK in
           Table HCUSTOMER violates Arc constraint on Table CUSTOMER -
           discriminator column 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_21_acustomer BEFORE
    INSERT OR UPDATE OF customerid, type ON acustomer
    FOR EACH ROW
DECLARE
    d CHAR(1);
BEGIN
    SELECT
        a.type
    INTO d
    FROM
        customer a
    WHERE
             a.customerid = :new.customerid
        AND a.type = :new.type;

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

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

CREATE OR REPLACE TRIGGER arc_fkarc_22_apolicy BEFORE
    INSERT OR UPDATE OF policyid ON apolicy
    FOR EACH ROW
DECLARE
    d CHAR(1);
BEGIN
    SELECT
        a.type
    INTO d
    FROM
        policy a
    WHERE
        a.policyid = :new.policyid;

    IF ( d IS NULL OR d <> 'A' ) THEN
```

```
        raise_application_error(-20223, 'FK APOLICY_POLICY_FK in Table
           APOLICY violates Arc constraint on Table POLICY -
           discriminator column type doesn''t have value ''A''');
    END IF;

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

CREATE OR REPLACE TRIGGER arc_fkarc_22_hpolicy BEFORE
    INSERT OR UPDATE OF policyid ON hpolicy
    FOR EACH ROW
DECLARE
    d CHAR(1);
BEGIN
    SELECT
        a.type
    INTO d
    FROM
        policy a
    WHERE
        a.policyid = :new.policyid;

    IF ( d IS NULL OR d <> 'H' ) THEN
        raise_application_error(-20223, 'FK HPOLICY_POLICY_FK in Table
           HPOLICY violates Arc constraint on Table POLICY -
           discriminator column type doesn''t have value ''H''');
    END IF;

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

CREATE SEQUENCE auto_apolicy_apid_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER auto_apolicy_apid_trg BEFORE
    INSERT ON auto_apolicy
    FOR EACH ROW
    WHEN ( new.apid IS NULL )
BEGIN
```

```
    :new.apid := auto_apolicy_apid_seq.nextval;
END;
/

CREATE SEQUENCE home_hpolicy_hpid_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER home_hpolicy_hpid_trg BEFORE
    INSERT ON home_hpolicy
    FOR EACH ROW
    WHEN ( new.hpid IS NULL )
BEGIN
    :new.hpid := home_hpolicy_hpid_seq.nextval;
END;
/



-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                            14
-- CREATE INDEX                             0
-- ALTER TABLE                             31
-- CREATE VIEW                              0
-- ALTER VIEW                               0
-- CREATE PACKAGE                           0
-- CREATE PACKAGE BODY                      0
-- CREATE PROCEDURE                         0
-- CREATE FUNCTION                          0
-- CREATE TRIGGER                           6
-- 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                          2
-- 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
```

3.4 Problem D

```
--customer
alter table customer add constraints c_customer_id check (customerid >=
    10000000);
alter table customer add constraints c_customer_gender check (gender in
    ('M', 'F'));
alter table customer add constraints c_customer_maritalstatus check (
    maritalstatus in ('M', 'S', 'W'));

--driver
alter table driver add constraints c_driver_num check (licenseNum >=
    10000000);

--home_hpolicy
alter table home_hpolicy add constraints c_home_hpolicy_hpID check (hpID
   >= 10000000);
alter table home_hpolicy add constraints c_home_hpolicy_status check (
    status in ('C', 'P'));

--home
alter table home add constraints c_home_id check (homeid >= 10000000);
alter table home add constraints c_home_type check (type in ('S', 'M', 'C
    ','T'));
alter table home add constraints c_home_autoFireNotification check (
    autoFireNotification in ('1', '0'));
alter table home add constraints c_home_securitySystem check (
    securitySystem in ('1', '0'));
alter table home add constraints c_home_swimmingpool check(swimmingpool
    in ('U', 'O', 'I', 'M'));
alter table home add constraints c_home_basement check (basement in ('1',
     '0'));

--payment
```

```
alter table payment add constraints c_payment_id check (paymentid >=
    10000000);
alter table payment add constraints c_payment_method check (method in ('
    PayPal', 'Credit', 'Debit','Check'));

--auto_apolicy
alter table auto_apolicy add constraints c_auto_apolicy_apID check (apID
    >= 10000000);
alter table auto_apolicy add constraints c_auto_apolicy_status check (
    status in ('C', 'P'));

--auto
alter table auto add constraints c_auto_vin check (vin >= 10000000);
alter table auto add constraints c_auto_status check (status in ('L', 'F
    ', 'O'));

--policy
alter table policy add constraints c_policy_policyid check (policyid >=
    10000000);

--invoice
alter table invoice add constraints c_invoice_id check (invoiceid >=
    10000000);
```

3.5 Problem E

```
--customer
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000000, 'A', 'A', 'A', 'NY
    ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000001, 'A', 'A', 'B', 'NY
    ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000002, 'A', 'A', 'C', 'NY
    ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000003, 'A', 'A', 'D', 'NY
    ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000004, 'A', 'A', 'E', 'NY
    ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000005, 'A', 'A', 'F', 'NY
    ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000006, 'A', 'A', 'G', 'NY
```

```
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000007, 'A', 'A', 'H', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000008, 'A', 'A', 'I', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000009, 'A', 'A', 'J', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');

insert into acustomer (customerid, type) values (10000000, 'A');
insert into acustomer (customerid, type) values (10000001, 'A');
insert into acustomer (customerid, type) values (10000002, 'A');
insert into acustomer (customerid, type) values (10000003, 'A');
insert into acustomer (customerid, type) values (10000004, 'A');
insert into acustomer (customerid, type) values (10000005, 'A');
insert into acustomer (customerid, type) values (10000006, 'A');
insert into acustomer (customerid, type) values (10000007, 'A');
insert into acustomer (customerid, type) values (10000008, 'A');
insert into acustomer (customerid, type) values (10000009, 'A');


insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000000, 'H', 'A', 'A', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000001, 'H', 'A', 'B', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000002, 'H', 'A', 'C', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000003, 'H', 'A', 'D', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000004, 'H', 'A', 'E', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000005, 'H', 'A', 'F', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000006, 'H', 'A', 'G', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
   , zipcode, gender, maritalstatus) values (10000007, 'H', 'A', 'H', 'NY
   ', 'NY', '5th Av', '88888', 'M', 'W');
```

```
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000008, 'H', 'A', 'I', 'NY
    ', 'NY', '5th Av', '88888', 'M', 'W');
insert into customer (customerid, type, fname, lname, state, city, street
    , zipcode, gender, maritalstatus) values (10000009, 'H', 'A', 'J', 'NY
    ', 'NY', '5th Av', '88888', 'M', 'W');


insert into hcustomer (customerid, type) values (10000000, 'H');
insert into hcustomer (customerid, type) values (10000001, 'H');
insert into hcustomer (customerid, type) values (10000002, 'H');
insert into hcustomer (customerid, type) values (10000003, 'H');
insert into hcustomer (customerid, type) values (10000004, 'H');
insert into hcustomer (customerid, type) values (10000005, 'H');
insert into hcustomer (customerid, type) values (10000006, 'H');
insert into hcustomer (customerid, type) values (10000007, 'H');
insert into hcustomer (customerid, type) values (10000008, 'H');
insert into hcustomer (customerid, type) values (10000009, 'H');


--home
insert into home (homeid, purchasedate, purchasevalue, area, type,
    autofirenotification, securitysystem, swimmingpool, basement,
    customerid) values (10000000, date'2020-01-01', 100, 5, 'S', '0', '0',
     'U', '1', 10000000);
insert into home (homeid, purchasedate, purchasevalue, area, type,
    autofirenotification, securitysystem, swimmingpool, basement,
    customerid) values (10000001, date'2020-01-02', 100, 5, 'M', '0', '1',
     'O', '0', 10000001);
insert into home (homeid, purchasedate, purchasevalue, area, type,
    autofirenotification, securitysystem, swimmingpool, basement,
    customerid) values (10000002, date'2020-01-03', 100, 5, 'C', '1', '0',
     'I', '1', 10000002);
insert into home (homeid, purchasedate, purchasevalue, area, type,
    autofirenotification, securitysystem, swimmingpool, basement,
    customerid) values (10000003, date'2020-02-01', 100, 5, 'T', '1', '1',
     'M', '0', 10000003);
insert into home (homeid, purchasedate, purchasevalue, area, type,
    autofirenotification, securitysystem, swimmingpool, basement,
    customerid) values (10000004, date'2020-03-01', 100, 5, 'S', '0', '0',
     'U', '0', 10000004);
insert into home (homeid, purchasedate, purchasevalue, area, type,
    autofirenotification, securitysystem, swimmingpool, basement,
    customerid) values (10000005, date'2020-03-02', 100, 5, 'M', '1', '1',
     'O', '0', 10000005);
insert into home (homeid, purchasedate, purchasevalue, area, type,
    autofirenotification, securitysystem, swimmingpool, basement,
    customerid) values (10000006, date'2020-01-01', 100, 5, 'C', '0', '0',
```

```
    'I', '1', 10000006);
insert into home (homeid, purchasedate, purchasevalue, area, type,
   autofirenotification, securitysystem, swimmingpool, basement,
   customerid) values (10000007, date'2008-01-01', 100, 5, 'T', '1', '0',
    'M', '1', 10000007);
insert into home (homeid, purchasedate, purchasevalue, area, type,
   autofirenotification, securitysystem, swimmingpool, basement,
   customerid) values (10000008, date'2014-01-01', 100, 5, 'S', '0', '0',
    'U', '1', 10000008);
insert into home (homeid, purchasedate, purchasevalue, area, type,
   autofirenotification, securitysystem, swimmingpool, basement,
   customerid) values (10000009, date'2012-03-01', 100, 5, 'M', '1', '0',
    'O', '1', 10000009);



--driver
insert into driver (licensenum, fname, lname, birthdate) values
   (10000000, 'd', 'a', date'1995-06-02');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000001, 'd', 'b', date'1995-08-02');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000002, 'd', 'c', date'1995-03-02');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000003, 'd', 'd', date'1995-06-21');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000004, 'd', 'e', date'1993-06-02');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000005, 'd', 'f', date'1995-06-02');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000006, 'd', 'g', date'1995-06-02');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000007, 'd', 'h', date'1995-06-02');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000008, 'd', 'i', date'1995-06-02');
insert into driver (licensenum, fname, lname, birthdate) values
   (10000009, 'd', 'j', date'1995-06-02');



--auto
insert into auto (vin, modelyear, status, customerid) values (10000000,
   date'1995-06-02', 'L', 10000000);
insert into auto (vin, modelyear, status, customerid) values (10000001,
   date'1995-06-02', 'F', 10000001);
insert into auto (vin, modelyear, status, customerid) values (10000002,
   date'1995-06-02', 'O', 10000002);
insert into auto (vin, modelyear, status, customerid) values (10000003,
   date'1995-06-02', 'L', 10000003);
```

```
insert into auto (vin, modelyear, status, customerid) values (10000004,
   date'1995-06-02', 'F', 10000004);
insert into auto (vin, modelyear, status, customerid) values (10000005,
   date'1995-06-02', 'O', 10000005);
insert into auto (vin, modelyear, status, customerid) values (10000006,
   date'1995-06-02', 'L', 10000006);
insert into auto (vin, modelyear, status, customerid) values (10000007,
   date'1995-06-02', 'F', 10000007);
insert into auto (vin, modelyear, status, customerid) values (10000008,
   date'1995-06-02', 'O', 10000008);
insert into auto (vin, modelyear, status, customerid) values (10000009,
   date'1995-06-02', 'L', 10000009);



--driver_auto
insert into driver_auto (vin, licensenum) values (10000000, 10000000);
insert into driver_auto (vin, licensenum) values (10000001, 10000001);
insert into driver_auto (vin, licensenum) values (10000002, 10000002);
insert into driver_auto (vin, licensenum) values (10000003, 10000003);
insert into driver_auto (vin, licensenum) values (10000004, 10000004);
insert into driver_auto (vin, licensenum) values (10000005, 10000005);
insert into driver_auto (vin, licensenum) values (10000006, 10000006);
insert into driver_auto (vin, licensenum) values (10000007, 10000007);
insert into driver_auto (vin, licensenum) values (10000008, 10000008);
insert into driver_auto (vin, licensenum) values (10000009, 10000009);

--policy
insert into policy (policyid, type) values (10000000, 'A');
insert into policy (policyid, type) values (10000001, 'A');
insert into policy (policyid, type) values (10000002, 'A');
insert into policy (policyid, type) values (10000003, 'A');
insert into policy (policyid, type) values (10000004, 'A');

insert into apolicy (policyid) values (10000000);
insert into apolicy (policyid) values (10000001);
insert into apolicy (policyid) values (10000002);
insert into apolicy (policyid) values (10000003);
insert into apolicy (policyid) values (10000004);

insert into policy (policyid, type) values (10000005, 'H');
insert into policy (policyid, type) values (10000006, 'H');
insert into policy (policyid, type) values (10000007, 'H');
insert into policy (policyid, type) values (10000008, 'H');
insert into policy (policyid, type) values (10000009, 'H');

insert into hpolicy (policyid) values (10000005);
insert into hpolicy (policyid) values (10000006);
```

```
insert into hpolicy (policyid) values (10000007);
insert into hpolicy (policyid) values (10000008);
insert into hpolicy (policyid) values (10000009);

--home_hpolicy
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000000, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000000, 10000005);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000001, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000001, 10000005);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000002, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000002, 10000005);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000003, date'2019-01-01', date
    '2020-01-01', 10000, 'P', 10000001, 10000006);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000004, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000003, 10000005);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000005, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000004, 10000007);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000006, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000004, 10000008);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000007, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000005, 10000009);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000008, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000006, 10000009);
insert into home_hpolicy (hpid, startdate, enddate, premuimamount, status
    , homeid, policyid) values (10000009, date'2020-01-01', date
    '2021-01-01', 10000, 'C', 10000007, 10000006);

--invoice
insert into invoice (invoiceid, paymentduedate, amount) values (10000000,
    date'2020-5-5', 10000);
insert into invoice (invoiceid, paymentduedate, amount) values (10000001,
    date'2020-5-5', 10000);
insert into invoice (invoiceid, paymentduedate, amount) values (10000002,
    date'2020-5-5', 10000);
insert into invoice (invoiceid, paymentduedate, amount) values (10000003,
    date'2020-5-5', 10000);
insert into invoice (invoiceid, paymentduedate, amount) values (10000004,
    date'2020-5-5', 10000);
```

```
insert into invoice (invoiceid, paymentduedate, amount) values (10000005,
    date'2020-5-5', 10000);
insert into invoice (invoiceid, paymentduedate, amount) values (10000006,
    date'2020-5-5', 10000);
insert into invoice (invoiceid, paymentduedate, amount) values (10000007,
    date'2020-5-5', 10000);
insert into invoice (invoiceid, paymentduedate, amount) values (10000008,
    date'2020-5-5', 10000);
insert into invoice (invoiceid, paymentduedate, amount) values (10000009,
    date'2020-5-5', 10000);

--auto_apolicy
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000000, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000000, 10000000);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000001, date'2019-01-01', date'2020-01-01',
    10000, 'P', 10000001, 10000000);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000002, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000002, 10000001);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000003, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000003, 10000001);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000004, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000004, 10000002);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000005, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000005, 10000002);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000006, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000006, 10000003);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000007, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000007, 10000003);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000008, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000008, 10000004);
insert into auto_apolicy (apid, startdate, enddate, premuimamount, status
    , vin, policyid) values (10000009, date'2020-01-01', date'2021-01-01',
    10000, 'C', 10000009, 10000004);

--payment
insert into payment (paymentid, paymentdate, method, hpid, invoiceID,
   amount) values (10000000, date'2020-01-01', 'PayPal', 10000000,
   10000000, 5000);
```

```
insert into payment (paymentid, paymentdate, method, hpid, invoiceID,
    amount) values (10000001, date'2020-01-01', 'Credit', 10000000,
    10000000, 5000);
insert into payment (paymentid, paymentdate, method, hpid, invoiceID,
    amount) values (10000002, date'2020-01-01', 'Debit', 10000001,
    10000001, 5000);
insert into payment (paymentid, paymentdate, method, hpid, invoiceID,
    amount) values (10000003, date'2020-01-01', 'Check', 10000001,
    10000001, 5000);
insert into payment (paymentid, paymentdate, method, hpid, invoiceID,
    amount) values (10000004, date'2020-01-01', 'PayPal', 10000002,
    10000002, 10000);
insert into payment (paymentid, paymentdate, method, hpid, invoiceID,
    amount) values (10000005, date'2020-01-01', 'PayPal', 10000003,
    10000003, 10000);
insert into payment (paymentid, paymentdate, method, apid, invoiceID,
    amount) values (10000006, date'2020-01-01', 'Credit', 10000000,
    10000004, 10000);
insert into payment (paymentid, paymentdate, method, apid, invoiceID,
    amount) values (10000007, date'2020-01-01', 'Check', 10000001,
    10000005, 10000);
insert into payment (paymentid, paymentdate, method, apid, invoiceID,
    amount) values (10000008, date'2020-01-01', 'PayPal', 10000002,
    10000006, 10000);
insert into payment (paymentid, paymentdate, method, apid, invoiceID,
    amount) values (10000009, date'2020-01-01', 'PayPal', 10000003,
    10000007, 10000);
```

3.6 Problem F

```
select 'ACUSTOMER',count(*) from ACUSTOMER;
select 'APOLICY',count(*) from APOLICY;
select 'AUTO',count(*) from AUTO;
select 'AUTO_APOLICY',count(*) from AUTO_APOLICY;
select 'CUSTOMER',count(*) from CUSTOMER;
select 'DRIVER',count(*) from DRIVER;
select 'DRIVER_AUTO',count(*) from DRIVER_AUTO;
select 'HCUSTOMER',count(*) from HCUSTOMER;
select 'HOME',count(*) from HOME;
select 'HOME_HPOLICY',count(*) from HOME_HPOLICY;
select 'HPOLICY',count(*) from HPOLICY;
select 'INVOICE',count(*) from INVOICE;
select 'PAYMENT',count(*) from PAYMENT;
select 'POLICY',count(*) from POLICY;
```

'ACUSTOMER'	COUNT(*)
ACUSTOMER	10

Download CSV

'APOLICY'	COUNT(*)
APOLICY	5

Download CSV

'AUTO'	COUNT(*)
AUTO	10

Download CSV

'AUTO_APOLICY'	COUNT(*)
AUTO_APOLICY	10

Download CSV

'CUSTOMER'	COUNT(*)
CUSTOMER	20

Download CSV

'DRIVER'	COUNT(*)
DRIVER	10

Download CSV

'DRIVER_AUTO'	COUNT(*)
DRIVER_AUTO	10

Download CSV

'HCUSTOMER'	COUNT(*)
HCUSTOMER	10

Download CSV

'HOME'	COUNT(*)
HOME	10

Download CSV

'HOME_HPOLICY'	COUNT(*)
HOME_HPOLICY	10

Download CSV

'HPOLICY'	COUNT(*)
HPOLICY	5

Download CSV

'INVOICE'	COUNT(*)
INVOICE	10

Download CSV

'PAYMENT'	COUNT(*)
PAYMENT	10

Download CSV

'POLICY'	COUNT(*)
POLICY	10

Download CSV

3.7 Problem G

```
--data dictionary queries that details all tables, columns-datatype-size-
    mandatory/optional, and
--attribute comments of schema objects
select table_name, column_name, data_type, data_precision, char_length,
    nullable, comments
from user_tab_columns natural join user_col_comments

--data dictionary queries that details all constraints
select table_name, constraint_name, constraint_type, search_condition,
    r_owner, r_constraint_name
from user_constraints
```

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_PRECISION	CHAR_LENGTH	NULLABLE	COMMENTS
APOLICY	POLICYID	NUMBER	8	0	N	the unique ID of policy
HOME	SECURITYSYSTEM	CHAR	-	1	N	show whether there is the home security system. 1 for there is and 0 for there is not.
PAYMENT	PAYMENTID	NUMBER	8	0	N	the unique ID for payment
AUTO_APOLICY	PREMIUMAMOUNT	NUMBER	8	0	N	auto insurance premium amount
CUSTOMER	GENDER	CHAR	-	1	Y	show whether customer is male or female, can only be "M" or "F", customer may choose not to provide gender data
HOME	SWIMMINGPOOL	VARCHAR2	-	1	Y	show which type of the swimming pool and whether there is a swimming pool. Not mandatory attribute, and blank means null showing there is no swimming pool. Can be U,O,I,M, representing Underground Swimming pool, Overground swimming pool, Indoor swimming pool and Multiple swimming pool respectively.
PAYMENT	METHOD	VARCHAR2	-	6	N	show method of payment, can be either as "PayPal", "Credit", "Debit", "Check"
CUSTOMER	CITY	VARCHAR2	-	10	N	the city where customer lives
HOME	CUSTOMERID	NUMBER	8	0	N	the unique ID for customer
AUTO	VIN	NUMBER	8	0	N	unqiue vehicle VIN (vehicle identification number)
CUSTOMER	LNAME	VARCHAR2	-	10	N	the last name of customer
DRIVER	FNAME	VARCHAR2	-	10	N	the first name of driver
HOME	PURCHASEVALUE	NUMBER	8	0	N	home purchase value
HOME	AREA	NUMBER	8	0	N	home area in Sq. Ft
HOME	TYPE	CHAR	-	1	N	show the type of home, can be S,M,C,T representing Single family, Multi Family, Condominium, Town house respectively
HOME_HPOLICY	PREMIUMAMOUNT	NUMBER	8	0	N	home insurance premium amount
INVOICE	PAYMENTDUEDATE	DATE	-	0	N	show payment due date
DRIVER	BIRTHDATE	DATE	-	0	N	the birthdate of driver
PAYMENT	PAYMENTDATE	DATE	-	0	N	payment date
AUTO_APOLICY	APID	NUMBER	8	0	N	the unique ID standing for the relationship between a specific auto and a specific auto policy
HOME_HPOLICY	HOMEID	NUMBER	8	0	N	the unique home ID
HPOLICY	POLICYID	NUMBER	8	0	N	the unique ID of policy
AUTO	CUSTOMERID	NUMBER	8	0	N	the unique ID for customer
HOME	PURCHASEDATE	DATE	-	0	N	home purchase date
POLICY	POLICYID	NUMBER	8	0	N	the unique ID of policy
AUTO_APOLICY	POLICYID	NUMBER	8	0	N	the unique ID of policy
CUSTOMER	CUSTOMERID	NUMBER	8	0	N	the unique ID for customer
CUSTOMER	MARITALSTATUS	CHAR	-	1	N	show marital status of customer, must be either "M", "S", or "W", representing "Married", "Single", and "Widow/Widower" respectively
HOME_HPOLICY	POLICYID	NUMBER	8	0	N	the unique ID of policy
AUTO_APOLICY	ENDDATE	DATE	-	0	N	auto insurance policy end date
AUTO_APOLICY	VIN	NUMBER	8	0	N	unqiue vehicle VIN (vehicle identification number)
CUSTOMER	FNAME	VARCHAR2	-	10	N	the first name of customer
ACUSTOMER	TYPE	CHAR	-	1	N	to show whether the customer is Automobile Insurance customer or Home Insurance customer or both, can only be "A" or "H"
AUTO	MODELYEAR	DATE	-	0	N	vehicle make-model year
CUSTOMER	ZIPCODE	VARCHAR2	-	5	N	the zipcode that where customer lives uses
HOME_HPOLICY	HPID	NUMBER	8	0	N	the unique ID standing for the relationship between a specific home and a specific home policy
POLICY	TYPE	CHAR	-	1	N	show the type of policy, a policy can either be Home Insurance policy or Auto Insurance policy, but cannot be both, can be "H" or "A"
AUTO	STATUS	CHAR	-	1	N	show the status of the vehicle, can be one of "L", "F", or "O" representing "Leased", "Financed", and "Owned"
AUTO_APOLICY	STARTDATE	DATE	-	0	N	auto insurance policy start date
DRIVER	LNAME	VARCHAR2	-	10	N	the last name of driver
HOME	HOMEID	NUMBER	8	0	N	the unique home ID
DRIVER	LICENSENUM	NUMBER	8	0	N	the unique license number of driver
HCUSTOMER	TYPE	CHAR	-	1	N	to show whether the customer is Automobile Insurance customer or Home Insurance customer or both, can only be "A" or "H"
PAYMENT	HPID	NUMBER	8	0	Y	the unique ID standing for the relationship between a specific home and a specific home policy
ACUSTOMER	CUSTOMERID	NUMBER	8	0	N	the unique ID for customer
AUTO_APOLICY	STATUS	CHAR	-	1	N	show auto policy insurance status. If auto insurance policy term is current, status column should have value "C", and if it is expired, it should have value "P".
HCUSTOMER	CUSTOMERID	NUMBER	8	0	N	the unique ID for customer
INVOICE	AMOUNT	NUMBER	8	0	N	show invoice amount
PAYMENT	APID	NUMBER	8	0	Y	the unique ID standing for the relationship between a specific auto and a specific auto policy
PAYMENT	AMOUNT	NUMBER	8	0	N	the amount of money of payment
CUSTOMER	TYPE	CHAR	-	1	N	to show whether the customer is Automobile Insurance customer or Home Insurance customer or both, can only be "A" or "H"
HOME	AUTOFIRENOTIFICATION	CHAR	-	1	N	show whether there is automatic fire notification to the fire department. 1 for there is and 0 for there is not.
INVOICE	INVOICEID	NUMBER	8	0	N	the unique of invoice
PAYMENT	INVOICEID	NUMBER	8	0	N	-
CUSTOMER	STATE	VARCHAR2	-	10	N	the state where customer lives
CUSTOMER	STREET	VARCHAR2	-	20	N	the street where customer lives
DRIVER_AUTO	VIN	NUMBER	8	0	N	unqiue vehicle VIN (vehicle identification number)
DRIVER_AUTO	LICENSENUM	NUMBER	8	0	N	the unique license number of driver
HOME	BASEMENT	CHAR	-	1	N	show whether there is basement. 1 for there is and 0 for there is not.
HOME_HPOLICY	STARTDATE	DATE	-	0	N	home insurance policy start date
HOME_HPOLICY	ENDDATE	DATE	-	0	N	home insurance policy end date
HOME_HPOLICY	STATUS	CHAR	-	1	N	show home policy insurance status. If home insurance policy term is current, status column should have value "C", and if it is expired, it should have value "P".

Download CSV
62 rows selected.

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_CONSTRAINT_NAME
ACUSTOMER	ACUSTOMER_CUSTOMER_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	CUSTOMER_PK
APOLICY	APOLICY_POLICY_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	POLICY_PK
AUTO	AUTO_ACUSTOMER_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	ACUSTOMER_PK
AUTO_APOLICY	AUTO_APOLICY_APOLICY_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	APOLICY_PK
AUTO_APOLICY	AUTO_APOLICY_AUTO_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	AUTO_PK
DRIVER_AUTO	DRIVER_AUTO_AUTO_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	AUTO_PK
DRIVER_AUTO	DRIVER_AUTO_DRIVER_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	DRIVER_PK
HCUSTOMER	HCUSTOMER_CUSTOMER_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	CUSTOMER_PK
HOME	HOME_HCUSTOMER_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	HCUSTOMER_PK
HOME_HPOLICY	HOME_HPOLICY_HOME_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	HOME_PK
HOME_HPOLICY	HOME_HPOLICY_HPOLICY_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	HPOLICY_PK
HPOLICY	HPOLICY_POLICY_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	POLICY_PK
PAYMENT	PAYMENT_AUTO_APOLICY_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	AUTO_APOLICY_PK
PAYMENT	PAYMENT_HOME_HPOLICY_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	HOME_HPOLICY_PK
PAYMENT	PAYMENT_INVOICE_FK	R	-	SQL_DVQOVIDONBLSZLYPSVHQLKUZM	INVOICE_PK
CUSTOMER	C_CUSTOMER_ID	C	customerid >= 10000000	-	-
CUSTOMER	C_CUSTOMER_GENDER	C	gender in ('M', 'F')	-	-
CUSTOMER	C_CUSTOMER_MARITALSTATUS	C	maritalstatus in ('M', 'S', 'W')	-	-
DRIVER	C_DRIVER_NUM	C	licenseNum >= 10000000	-	-
HOME_HPOLICY	C_HOME_HPOLICY_HPID	C	hpID >= 10000000	-	-
HOME_HPOLICY	C_HOME_HPOLICY_STATUS	C	status in ('C', 'P')	-	-
HOME	C_HOME_ID	C	homeid >= 10000000	-	-
HOME	C_HOME_TYPE	C	type in ('S', 'M', 'C','T')	-	-
HOME	C_HOME_AUTOFIRENOTIFICATION	C	autoFireNotification in ('1', '0')	-	-
HOME	C_HOME_SECURITYSYSTEM	C	securitySystem in ('1', '0')	-	-
HOME	C_HOME_SWIMMINGPOOL	C	swimmingpool in ('U', 'O', 'I', 'M')	-	-
HOME	C_HOME_BASEMENT	C	basement in ('1', '0')	-	-
PAYMENT	C_PAYMENT_ID	C	paymentid >= 10000000	-	-
PAYMENT	C_PAYMENT_METHOD	C	method in ('PayPal', 'Credit', 'Debit','Check')	-	-
AUTO_APOLICY	C_AUTO_APOLICY_APID	C	apID >= 10000000	-	-
AUTO_APOLICY	C_AUTO_APOLICY_STATUS	C	status in ('C', 'P')	-	-
CUSTOMER	SYS_C0027994432	C	"STATE" IS NOT NULL	-	-
CUSTOMER	SYS_C0027994433	C	"CITY" IS NOT NULL	-	-
CUSTOMER	SYS_C0027994434	C	"STREET" IS NOT NULL	-	-
CUSTOMER	SYS_C0027994435	C	"ZIPCODE" IS NOT NULL	-	-
CUSTOMER	SYS_C0027994436	C	"MARITALSTATUS" IS NOT NULL	-	-
CUSTOMER	CH_INH_CUSTOMER	C	type IN ('C', 'H')	-	-
DRIVER	SYS_C0027994444	C	"LICENSENUM" IS NOT NULL	-	-
DRIVER	SYS_C0027994445	C	"FNAME" IS NOT NULL	-	-
DRIVER	SYS_C0027994446	C	"LNAME" IS NOT NULL	-	-
DRIVER	SYS_C0027994447	C	"BIRTHDATE" IS NOT NULL	-	-
DRIVER_AUTO	SYS_C0027994454	C	"VIN" IS NOT NULL	-	-
DRIVER_AUTO	SYS_C0027994455	C	"LICENSENUM" IS NOT NULL	-	-
HCUSTOMER	SYS_C0027994457	C	"CUSTOMERID" IS NOT NULL	-	-
HCUSTOMER	SYS_C0027994458	C	"TYPE" IS NOT NULL	-	-
HOME	SYS_C0027994460	C	"HOMEID" IS NOT NULL	-	-
HOME	SYS_C0027994461	C	"PURCHASEDATE" IS NOT NULL	-	-
HOME	SYS_C0027994462	C	"PURCHASEVALUE" IS NOT NULL	-	-
HOME	SYS_C0027994463	C	"AREA" IS NOT NULL	-	-
HOME	SYS_C0027994464	C	"TYPE" IS NOT NULL	-	-
HOME	SYS_C0027994465	C	"AUTOFIRENOTIFICATION" IS NOT NULL	-	-
HOME	SYS_C0027994466	C	"SECURITYSYSTEM" IS NOT NULL	-	-
HOME	SYS_C0027994467	C	"BASEMENT" IS NOT NULL	-	-
HOME	SYS_C0027994468	C	"CUSTOMERID" IS NOT NULL	-	-
HOME_HPOLICY	SYS_C0027994478	C	"HPID" IS NOT NULL	-	-
HOME_HPOLICY	SYS_C0027994479	C	"STARTDATE" IS NOT NULL	-	-
HOME_HPOLICY	SYS_C0027994480	C	"ENDDATE" IS NOT NULL	-	-
HOME_HPOLICY	SYS_C0027994481	C	"PREMUIMAMOUNT" IS NOT NULL	-	-
HOME_HPOLICY	SYS_C0027994482	C	"STATUS" IS NOT NULL	-	-
HOME_HPOLICY	SYS_C0027994483	C	"HOMEID" IS NOT NULL	-	-
HOME_HPOLICY	SYS_C0027994484	C	"POLICYID" IS NOT NULL	-	-
HPOLICY	SYS_C0027994486	C	"POLICYID" IS NOT NULL	-	-
INVOICE	SYS_C0027994490	C	"INVOICEID" IS NOT NULL	-	-
INVOICE	SYS_C0027994491	C	"PAYMENTDUEDATE" IS NOT NULL	-	-
INVOICE	SYS_C0027994492	C	"AMOUNT" IS NOT NULL	-	-
PAYMENT	SYS_C0027994499	C	"PAYMENTID" IS NOT NULL	-	-
PAYMENT	SYS_C0027994500	C	"PAYMENTDATE" IS NOT NULL	-	-
PAYMENT	SYS_C0027994501	C	"METHOD" IS NOT NULL	-	-
PAYMENT	SYS_C0027994502	C	"INVOICEID" IS NOT NULL	-	-
PAYMENT	SYS_C0027994503	C	"AMOUNT" IS NOT NULL	-	-

POLICY	SYS_C0027994505	C	"POLICYID" IS NOT NULL	-	-
POLICY	SYS_C0027994506	C	"TYPE" IS NOT NULL	-	-
POLICY	CH_INH_POLICY	C	type IN ('A', 'H')	-	-
ACUSTOMER	SYS_C0027994408	C	"CUSTOMERID" IS NOT NULL	-	-
ACUSTOMER	SYS_C0027994409	C	"TYPE" IS NOT NULL	-	-
APOLICY	SYS_C0027994411	C	"POLICYID" IS NOT NULL	-	-
AUTO	SYS_C0027994413	C	"VIN" IS NOT NULL	-	-
AUTO	SYS_C0027994414	C	"MODELYEAR" IS NOT NULL	-	-
AUTO	SYS_C0027994415	C	"STATUS" IS NOT NULL	-	-
AUTO	SYS_C0027994416	C	"CUSTOMERID" IS NOT NULL	-	-
AUTO_APOLICY	SYS_C0027994420	C	"APID" IS NOT NULL	-	-
AUTO_APOLICY	SYS_C0027994421	C	"STARTDATE" IS NOT NULL	-	-
AUTO_APOLICY	SYS_C0027994422	C	"ENDDATE" IS NOT NULL	-	-
AUTO_APOLICY	SYS_C0027994423	C	"PREMUIMAMOUNT" IS NOT NULL	-	-
AUTO_APOLICY	SYS_C0027994424	C	"STATUS" IS NOT NULL	-	-
AUTO_APOLICY	SYS_C0027994425	C	"VIN" IS NOT NULL	-	-
AUTO_APOLICY	SYS_C0027994426	C	"POLICYID" IS NOT NULL	-	-
CUSTOMER	SYS_C0027994428	C	"CUSTOMERID" IS NOT NULL	-	-
CUSTOMER	SYS_C0027994429	C	"TYPE" IS NOT NULL	-	-
CUSTOMER	SYS_C0027994430	C	"FNAME" IS NOT NULL	-	-
CUSTOMER	SYS_C0027994431	C	"LNAME" IS NOT NULL	-	-
AUTO	C_AUTO_VIN	C	vin >= 10000000	-	-
AUTO	C_AUTO_STATUS	C	status in ('L', 'F', 'O')	-	-
POLICY	C_POLICY_POLICYID	C	policyid >= 10000000	-	-
INVOICE	C_INVOICE_ID	C	invoiceid >= 10000000	-	-
HOME	HOME_PK	P	-	-	-
HOME_HPOLICY	HOME_HPOLICY_PK	P	-	-	-
HPOLICY	HPOLICY_PK	P	-	-	-
INVOICE	INVOICE_PK	P	-	-	-
PAYMENT	PAYMENT_PK	P	-	-	-
POLICY	POLICY_PK	P	-	-	-
ACUSTOMER	ACUSTOMER_PK	P	-	-	-
APOLICY	APOLICY_PK	P	-	-	-
AUTO	AUTO_PK	P	-	-	-
AUTO_APOLICY	AUTO_APOLICY_PK	P	-	-	-
CUSTOMER	CUSTOMER_PK	P	-	-	-
DRIVER	DRIVER_PK	P	-	-	-
DRIVER_AUTO	DRIVER_AUTO_PK	P	-	-	-
HCUSTOMER	HCUSTOMER_PK	P	-	-	-

Download CSV
109 rows selected.

3.8 Problem H

3.8.1 Q1: Table joins with at least 3 tables in join

Select query:

```
SELECT A.vin, C.lname
FROM AUTO A INNER JOIN DRIVER_AUTO B ON
A.vin = B.vin
INNER JOIN DRIVER C ON
B.licenseNum = C.licenseNum
```

Result of the query:

VIN	LNAME
10000000	a
10000001	b
10000002	c
10000003	d
10000004	e
10000005	f
10000006	g
10000007	h
10000008	i
10000009	j

Download CSV

10 rows selected.

Purpose of query: select the vin of the car and its corresponding drivers' last name.

3.8.2 Q2: Multi-row subquery

Select query:

```
SELECT homeID, purchasedate, purchasevalue, area
FROM home
WHERE homeID IN
(
SELECT homeID
FROM home_hpolicy
WHERE status = 'C'
);
```

Result of the query:

HOMEID	PURCHASEDATE	PURCHASEVALUE	AREA
10000000	01-JAN-20	100	5
10000001	02-JAN-20	100	5
10000002	03-JAN-20	100	5
10000003	01-FEB-20	100	5
10000004	01-MAR-20	100	5
10000005	02-MAR-20	100	5
10000006	01-JAN-20	100	5
10000007	01-JAN-08	100	5

Download CSV

8 rows selected.

Purpose of query: select homeID, purchasedate, purchasevalue and area of home whose home policy status is "Current".

3.8.3 Q3: Co-related subquery

Select query:

```
SELECT *
FROM payment pa
WHERE amount>
(
SELECT AVG(amount)
FROM payment
WHERE method = pa.method
);
```

Result of the query:

PAYMENTID	PAYMENTDATE	METHOD	HPID	APID	INVOICEID	AMOUNT
10000004	01-JAN-20	PayPal	10000002	-	10000002	10000
10000005	01-JAN-20	PayPal	10000003	-	10000003	10000
10000006	01-JAN-20	Credit	-	10000000	10000004	10000
10000007	01-JAN-20	Check	-	10000001	10000005	10000
10000008	01-JAN-20	PayPal	-	10000002	10000006	10000
10000009	01-JAN-20	PayPal	-	10000003	10000007	10000

Download CSV

6 rows selected.

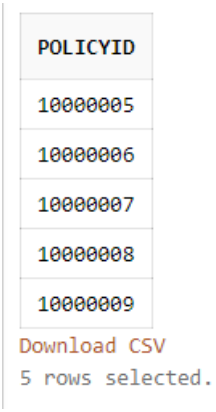
Purpose of query: select the information of payment whose amount is larger than the average amount of the payment with the same payment method.

3.8.4 Q4: SET operator query

Select query:

```
SELECT policyID
FROM HPOLICY
INTERSECT
SELECT policyID
FROM HOME_HPOLICY;
```

Result of the query:



POLICYID
10000005
10000006
10000007
10000008
10000009

Download CSV
5 rows selected.

Purpose of query: select the home policy ID that the policy is chosen and being used by home customer.

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

Select query:

```
SELECT
  *
FROM
(
    SELECT
        homeid,
        purchasedate,
        purchasevalue
    FROM
        home
    ORDER BY
        purchasedate DESC
)
WHERE
ROWNUM <= 10;
```

Result of the query:

HOMEID	PURCHASEDATE	PURCHASEVALUE
10000005	02-MAR-20	100
10000004	01-MAR-20	100
10000003	01-FEB-20	100
10000002	03-JAN-20	100
10000001	02-JAN-20	100
10000000	01-JAN-20	100
10000006	01-JAN-20	100
10000008	01-JAN-14	100
10000009	01-MAR-12	100
10000007	01-JAN-08	100

Download CSV

10 rows selected.

Purpose of query: select no more than 10 rows of homeID, purchaseDate and purchseValue from home with the descendent ordering of purchase value.