**Final Project Part 1**

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

**A screenshot of a cell phone

Description automatically generated**

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

**A screenshot of a computer

Description automatically generated**

1. **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 );

1. **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));

1. **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;

1. **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",

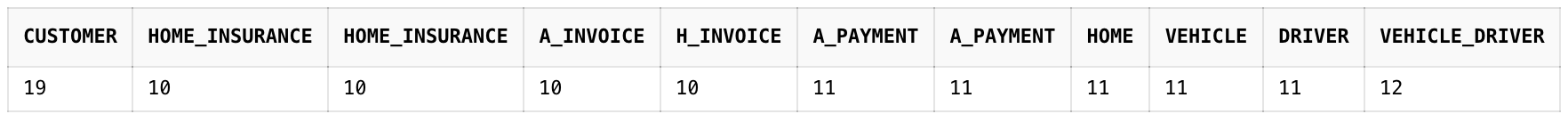
(SELECT COUNT(\*) FROM HOME) as "HOME",

(SELECT COUNT(\*) FROM VEHICLE) as "VEHICLE",

(SELECT COUNT(\*) FROM DRIVER) as "DRIVER",

(SELECT COUNT(\*) FROM VEHICLE\_DRIVER) as "VEHICLE\_DRIVER"

from dual



1. **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;

A close up of text on a white surface

Description automatically generated

1. **Write SQL queries using each of following**
2. **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 screenshot of a cell phone

Description automatically generated

**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.

1. **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:**

A screenshot of a cell phone

Description automatically generated

**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.

1. **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:**

A screenshot of a cell phone

Description automatically generated

**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.