Name: Tharindu Rehan Fernando Ponnahennedige

ID:22530478

Course: INFO601 Data Modelling & SQL/2401

TASK 2

Your challenge is to create MSSQL Server scripts to perform the tasks, as outline below. The original Oracle scripts are included for you to use as a guide.

PART A:

1. The first script is responsible for creating the tables and populating them with data.

1.1. Rewriting the Oracle Script in MSSQL Server and make changes to the data types, as required to make them MSSQL Server compatible.

1.2. Creating each of the tables

```
-- ************************************************************************
-- Create database structure
-- ************************************************************************

-- Lookup Table Colour
CREATE TABLE lu_colours (
    c_no INT IDENTITY(1,1) PRIMARY KEY,
    c_colour VARCHAR(15) NOT NULL
);

-- Table Customer
CREATE TABLE customers (
    c_id INT IDENTITY(1,1) PRIMARY KEY,
    c_fname VARCHAR(20) NOT NULL,
    c_lname VARCHAR(20) NOT NULL,
    c_gender CHAR(1) CHECK (c_gender IN ('M','F')),
    c_address1 VARCHAR(20) NOT NULL,
    c_address2 VARCHAR(20) NOT NULL,
    c_address3 VARCHAR(20),
    c_ph VARCHAR(10) NOT NULL
);

-- Table Sales Person
CREATE TABLE sales_persons (
    sp_id VARCHAR(5) PRIMARY KEY,
    sp_fname VARCHAR(20) NOT NULL,
    sp_lname VARCHAR(20) NOT NULL,
    sp_startdate DATE NOT NULL,
    sp_cellph VARCHAR(10) NOT NULL,
    sp_comrate DECIMAL(3,2) NOT NULL,
    sp_sup VARCHAR(5)
);
```

```sql
-- Table Vehicle
CREATE TABLE vehicles (
    v_regno VARCHAR(7) PRIMARY KEY,
    v_make VARCHAR(10) NOT NULL,
    v_model VARCHAR(15) NOT NULL,
    v_year INT NOT NULL CHECK (v_year >= 1990),
    v_numowners TINYINT DEFAULT 0 CHECK (v_numowners <= 5),
    v_price DECIMAL(7,2) NOT NULL,
    v_miledge INT NOT NULL,
    c_no INT,
    FOREIGN KEY (c_no) REFERENCES lu_colours(c_no)
);

-- Table Sales Purchase with invoice# starting at 10000000
CREATE TABLE sales_purchases (
    sp_invoice INT IDENTITY(10000000,1) PRIMARY KEY,
    sp_datesold DATE NOT NULL,
    sp_saleprice DECIMAL(7,2),
    sp_addncost DECIMAL(7,2),
    sp_deposit DECIMAL(7,2),
    sp_total DECIMAL(7,2),
    sp_id VARCHAR(5),
    c_id INT,
    v_regno VARCHAR(7) UNIQUE,
    FOREIGN KEY (sp_id) REFERENCES sales_persons(sp_id),
    FOREIGN KEY (c_id) REFERENCES customers(c_id),
    FOREIGN KEY (v_regno) REFERENCES vehicles(v_regno)
);

-- Table Payment with invoice# starting at 90000000
CREATE TABLE payments (
    p_invoice INT IDENTITY(90000000,1) PRIMARY KEY,
    p_date DATE NOT NULL,
    p_amount DECIMAL(7,2) NOT NULL,
    c_id INT,
    sp_invoice INT,
    FOREIGN KEY (c_id) REFERENCES customers(c_id),
    FOREIGN KEY (sp_invoice) REFERENCES sales_purchases(sp_invoice)
);
```

```sql
-- Table Supplier
CREATE TABLE suppliers (
    s_code VARCHAR(5) PRIMARY KEY,
    s_name VARCHAR(25) NOT NULL,
    s_address1 VARCHAR(20) NOT NULL,
    s_address2 VARCHAR(20) NOT NULL,
    s_address3 VARCHAR(20) NOT NULL,
    s_contactp VARCHAR(20) NOT NULL,
    s_contactph VARCHAR(10) NOT NULL
);

-- Table Item
CREATE TABLE items (
    i_no INT IDENTITY(1,1) PRIMARY KEY,
    i_make VARCHAR(10) NOT NULL,
    i_model VARCHAR(15) NOT NULL,
    i_year INT NOT NULL CHECK (i_year >= 1990),
    i_price DECIMAL(7,2),
    CONSTRAINT uk_items UNIQUE (i_make, i_model, i_year)
);

-- Table Order with order number starting at 80000000
CREATE TABLE orders (
    o_id INT IDENTITY(80000000,1) PRIMARY KEY,
    o_date DATE NOT NULL,
    o_totalqty TINYINT,
    o_total DECIMAL(9,2),
    s_code VARCHAR(5),
    sp_id VARCHAR(5),
    FOREIGN KEY (s_code) REFERENCES suppliers(s_code),
    FOREIGN KEY (sp_id) REFERENCES sales_persons(sp_id)
);

-- Table Order Line
CREATE TABLE order_lines (
    o_id INT,
    i_no INT,
    i_make VARCHAR(10),
    i_model VARCHAR(15),
    i_price DECIMAL(7,2),
    i_year INT,
    ol_qty TINYINT DEFAULT 1,
    ol_subtotal DECIMAL(7,2),
    PRIMARY KEY (o_id, i_no),
    FOREIGN KEY (o_id) REFERENCES orders(o_id),
    FOREIGN KEY (i_no) REFERENCES items(i_no)
);
```

1.3. Modify the scripts to populate the tables.

```
--#############################################################
-- Insert Customer Data
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Katniss', 'Everde', 'F', '45 Dinsdale Rd', 'Frankton', 'Hamilton', '8123456');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Peeta', 'Mallark', 'M', '123 Anglesea St', 'Maeora', 'Hamilton', '8111111');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Gale', 'Hawthorne', 'M', '717 River Rd', 'Melville', 'Hamilton', '8221122');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Haymitch', 'Avern', 'M', '24 Duke St', 'Park View', 'Cambridge', '8881888');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Primrose', 'Suzci', 'F', 'RD1', 'Park View', 'Cambridge', '8112211');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Effie', 'Trinket', 'F', '655 Tristram St', 'Frankton', 'Hamilton', '8181818');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Hamilon', 'Hardna', 'M', '23 Lake Rd', 'Frankton', 'Hamilton', '8151156');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Kermin', 'Shaldon', 'M', '23 Rimu St', 'Maeroa', 'Hamilton', '8113121');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Sukki', 'Rightson', 'F', '252 Fast Rd', 'Silverdale', 'Hamilton', '8222322');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Mathew', 'Allen', 'M', '45 Shelfin St', 'West Side View', 'Cambridge', '8816988');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Frank', 'Smith', 'M', '45 Dinsdale Rd', 'Dinsdale', 'Hamilton', '8183456');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Tody', 'Ever', 'M', '49 Queen St', 'Park View', 'Cambridge', '8823346');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Toby', 'Deen', 'M', '45 Queen St', 'Hamilton Central', 'Hamilton', '8623456');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Pip', 'Jett', 'M', '423 Anglesea St', 'Hamilton Central', 'Hamilton', '8456123');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('John', 'Mack', 'M', '43 Cresent St', 'Melville', 'Hamilton', '8443143');


SELECT c_id, c_fname, c_lname, c_address3 FROM customers;

-- #############################################################
-- Insert Sales Person Data
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('MK201', 'Michael', 'Knapp', '10-Jun-15', '0213390823', 0.25, '');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('KK634', 'Kelly', 'Knapp', '10-Jul-15', '0213390823', 0.15, '');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('BP301', 'Bradley', 'Palmer', '24-Aug-15', '0219878123', 0.15, 'MK201');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('KC312', 'Karen', 'Craften', '21-Aug-15', '0213940903', 0.25, 'KK634');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('KH981', 'Kane', 'Hunter', '12-Mar-16', '0212132231', 0.15, 'MK201');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('JW351', 'John', 'Wrights', '20-Mar-16', '0210998212', 0.15, '');

SELECT sp_id, sp_fname, sp_lname, sp_cellph, sp_sup FROM sales_persons;
```

```
--#############################################################
-- Insert Customer Data
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Katniss', 'Everde', 'F', '45 Dinsdale Rd', 'Frankton', 'Hamilton', '8123456');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Peeta', 'Mallark', 'M', '123 Anglesea St', 'Maeora', 'Hamilton', '8111111');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Gale', 'Hawthorne', 'M', '717 River Rd', 'Melville', 'Hamilton', '8221122');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Haymitch', 'Avern', 'M', '24 Duke St', 'Park View', 'Cambridge', '8881888');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Primrose', 'Suzci', 'F', 'RD1', 'Park View', 'Cambridge', '8112211');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Effie', 'Trinket', 'F', '655 Tristram St', 'Frankton', 'Hamilton', '8181818');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Hamilon', 'Hardna', 'M', '23 Lake Rd', 'Frankton', 'Hamilton', '8151156');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Kermin', 'Shaldon', 'M', '23 Rimu St', 'Maeroa', 'Hamilton', '8113121');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Sukki', 'Rightson', 'F', '252 Fast Rd', 'Silverdale', 'Hamilton', '8222322');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Mathew', 'Allen', 'M', '45 Shelfin St', 'West Side View', 'Cambridge', '8816988');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Frank', 'Smith', 'M', '45 Dinsdale Rd', 'Dinsdale', 'Hamilton', '8183456');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Tody', 'Ever', 'M', '49 Queen St', 'Park View', 'Cambridge', '8823346');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Toby', 'Deen', 'M', '45 Queen St', 'Hamilton Central', 'Hamilton', '8623456');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('Pip', 'Jett', 'M', '423 Anglesea St', 'Hamilton Central', 'Hamilton', '8456123');
INSERT INTO customers (c_fname, c_lname, c_gender, c_address1, c_address2, c_address3, c_ph) VALUES ('John', 'Mack', 'M', '43 Cresent St', 'Melville', 'Hamilton', '8443143');


SELECT c_id, c_fname, c_lname, c_address3 FROM customers;

-- #############################################################
-- Insert Sales Person Data
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('MK201', 'Michael', 'Knapp', '10-Jun-15', '0213390823', 0.25, '');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('KK634', 'Kelly', 'Knapp', '10-Jul-15', '0213390823', 0.15, '');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('BP301', 'Bradley', 'Palmer', '24-Aug-15', '0219878123', 0.15, 'MK201');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('KC312', 'Karen', 'Craften', '21-Aug-15', '0213940903', 0.25, 'KK634');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('KH981', 'Kane', 'Hunter', '12-Mar-16', '0212132231', 0.15, 'MK201');
INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup) VALUES ('JW351', 'John', 'Wrights', '20-Mar-16', '0210998212', 0.15, '');

SELECT sp_id, sp_fname, sp_lname, sp_cellph, sp_sup FROM sales_persons;
```

```sql
-- ############################################################
-- Insert Payment Data
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('27-Jul-2015'), 1000, 1, 10000000);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('27-Aug-2015'), 1000, 1, 10000000);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('27-Sep-2015'), 1000, 1, 10000000);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('17-Aug-2015'), 1000, 2, 10000001);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('17-Sep-2015'), 1000, 2, 10000001);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('17-Oct-2015'), 1000, 2, 10000001);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('17-Nov-2015'), 1000, 2, 10000001);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('7-Jul-2015'), 1000, 3, 10000002);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('11-Jul-2015'), 1000, 4, 10000003);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('17-Jul-2015'), 1000, 5, 10000004);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('19-Aug-2015'), 1000, 6, 10000005);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('19-Sep-2015'), 1000, 6, 10000005);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('19-Oct-2015'), 1000, 6, 10000005);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('7-Aug-2015'), 1000, 7, 10000006);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('7-Jan-2016'), 1000, 7, 10000006);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('7-Feb-2016'), 1000, 7, 10000006);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('10-Sep-2015'), 1000, 8, 10000007);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('10-Oct-2015'), 1000, 8, 10000007);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('10-Nov-2015'), 1000, 8, 10000007);
INSERT INTO payments (p_date, p_amount, c_id, sp_invoice) VALUES (('10-Feb-2016'), 1000, 8, 10000007);


SELECT p_invoice, p_date, p_amount, c_fname, sp_invoice FROM payments, customers WHERE payments.c_id = customers.c_id;
```

```sql
-- Insert Item Data
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'Accord', 2010, 3000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'Accord', 2012, 5000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'Accord', 2013, 7000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'Accord', 2014, 13000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'CRZ', 2012, 4000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'Civic', 2015, 5000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'Jazz', 2011, 2500);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'Jazz', 2012, 3500);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'Crossroad', 2010, 8000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'VTR250F', 2011, 3000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Honda', 'CB900F Hornet', 2010, 2500);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Aurion', 2013, 10000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Hiace', 2014, 8000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Hilux', 2012, 9000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Camry', 2013, 12000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Vitz', 2010, 8000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Prius', 2012, 9000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Corolla', 2015, 8000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Yaris', 2014, 6000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Yaris', 2015, 7000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Rav4', 2014, 12000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Toyota', 'Rav4', 2016, 14000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Mazda', '323', 2012, 4000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Mazda', '323', 2013, 4500);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Mazda', '323', 2014, 5500);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Mazda', '353', 2013, 5000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Mazda', '353', 2014, 6000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Mazda', '626', 2014, 5000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Mazda', 'Familia', 2010, 2500);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Muruno', 2014, 9000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Pulsar', 2016, 8000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Bluebird', 2011, 4000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Bluebird', 2012, 5000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Bluebird', 2013, 6000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'X-Trail', 2016, 12000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Navara', 2016, 16000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Altima', 2013, 8000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Note', 2009, 3000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Skyline', 2014, 15000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Nissan', 'Skyline', 2012, 7000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Ford', 'Falcon', 2012, 7000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Ford', 'Falcon', 2013, 8000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Ford', 'Falcon', 2014, 9000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('BMW', '322', 2010, 15000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('BMW', '727', 2013, 18000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('BMW', 'X5', 2014, 18000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('BMW', '525i', 2009, 9000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('BMW', '525i', 2010, 10000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('BMW', 'M3', 2010, 17000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('BMW', 'Z4', 2012, 12000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Holden', 'Commodore SsV', 2013, 10000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Holden', 'Commodore', 2015, 13000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Holden', 'Commodore XR6', 2011, 15000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Holden', 'Commodore XR6', 2012, 17000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Holden', 'Commodore XR6', 2013, 19000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Holden', 'Clubsport', 2013, 19000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Holden', 'Clubsport', 2014, 21000);
INSERT INTO items (i_make, i_model, i_year, i_price) VALUES ('Holden', 'Clubsport', 2015, 22000);

SELECT * FROM items;
```

```sql
-- ############################################################
-- Insert Supplier Data
INSERT INTO suppliers (s_code, s_name, s_address1, s_address2, s_address3, s_contactp, s_contactph) VALUES ('XTRQC','XTREME Quality Cars Inc','22A Great North Rd','Pimbroke', 'Auckland', 'Murray Knapp', '096666432');

-- ############################################################
-- Insert Order Data
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('01-Jun-2015'),'XTRQC','MK201');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('05-Jun-2015'),'XTRQC','MK201');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('06-Jun-2015'),'XTRQC','MK201');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('10-Jun-2015'),'XTRQC','MK201');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('11-Jun-2015'),'XTRQC','MK201');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('12-Jun-2015'),'XTRQC','MK201');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('21-Jul-2015'),'XTRQC','KK634');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('01-Aug-2015'),'XTRQC','KK634');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('11-Aug-2015'),'XTRQC','KK634');
INSERT INTO orders (o_date, s_code, sp_id) VALUES (('21-Aug-2015'),'XTRQC','KK634');

SELECT * FROM orders;
```

```sql
-- Insert Order Line Data
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 28;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 33;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 18;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 21;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 22;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 1;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 23;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 24;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 38;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000000, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 4;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 3 FROM items WHERE items.i_no = 34;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 35;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 20;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 47;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 49;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 31;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 36;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 2;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 25;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000001, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 27;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000002, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 9;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000002, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 36;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000002, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 2 FROM items WHERE items.i_no = 19;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000002, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 48;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000002, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 51;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000003, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 40;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000003, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 39;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000003, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 3;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000003, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 3 FROM items WHERE items.i_no = 25;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000004, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 34;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000004, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 2 FROM items WHERE items.i_no = 52;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000004, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 32;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000004, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 10;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000004, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 11;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000005, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 2 FROM items WHERE items.i_no = 7;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000005, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 51;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000005, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 42;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000005, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 3 FROM items WHERE items.i_no = 29;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000005, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 2 FROM items WHERE items.i_no = 53;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000006, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 44;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000006, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 2 FROM items WHERE items.i_no = 3;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000006, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 3 FROM items WHERE items.i_no = 58;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000006, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 45;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000006, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 23;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000007, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 42;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000007, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 2 FROM items WHERE items.i_no = 43;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000007, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 55;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000008, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 2 FROM items WHERE items.i_no = 50;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000008, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 51;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000009, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 52;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000009, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 29;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000009, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 2 FROM items WHERE items.i_no = 30;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000009, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 3 FROM items WHERE items.i_no = 55;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000009, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 3;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000009, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 5;
INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty) SELECT 80000009, items.i_no, items.i_make, items.i_model, items.i_price, items.i_year, 1 FROM items WHERE items.i_no = 9;
```

NAME : THARINDU REHAN FERNANDO ID: 22530478

```sql
-- Insert Colour Data - 10 rows
INSERT INTO lu_colours (c_colour) VALUES ('Red');
INSERT INTO lu_colours (c_colour) VALUES ('Black');
INSERT INTO lu_colours (c_colour) VALUES ('Silver');
INSERT INTO lu_colours (c_colour) VALUES ('White');
INSERT INTO lu_colours (c_colour) VALUES ('Blue');
INSERT INTO lu_colours (c_colour) VALUES ('Purple');
INSERT INTO lu_colours (c_colour) VALUES ('Yellow');
INSERT INTO lu_colours (c_colour) VALUES ('Paua');
INSERT INTO lu_colours (c_colour) VALUES ('Orange');
INSERT INTO lu_colours (c_colour) VALUES ('Green');

SELECT * FROM lu_colours;
```

```sql
-- Insert Car Data
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('GKN534', 'Mazda', '626', 2014, 2, 14599, 59633,1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('ALP394', 'Nissan', 'Bluebird', 2012, 1, 15995, 59000, 2);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('NT6776', 'Toyota', 'Corolla', 2015, 1, 24990, 20565, 8);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('KGH334', 'Toyota', 'Rav4', 2014, 1, 36990, 6509, 3);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('PHG902', 'Toyota', 'Rav4', 2016, 0, 46990, 14, 6);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('GLM123', 'Honda', 'Accord', 2010, 2, 9995, 119000, 4);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('OM1122', 'Mazda', '323', 2012, 1, 12995, 89000, 5);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('RS3456', 'Mazda', '323', 2013, 1, 13995, 110000, 1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('ZHU123', 'Nissan', 'Note', 2009, 2, 9995, 89000, 3);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('PRH345', 'Honda', 'Accord', 2014, 1, 41885, 5500, 9);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('SUT143', 'Nissan', 'Bluebird', 2013, 1, 17995, 61000, 6);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('GBR553', 'Nissan', 'X-Trail', 2016, 0, 39995, 12, 3);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('LQRT67', 'Toyota', 'Yaris', 2015, 1, 20990, 6825, 2);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('SALES1', 'BMW', '525i', 2009, 1, 27440, 39400, 7);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('LMV541', 'BMW', 'M3', 2010, 2, 54990, 77000, 2);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('NIS123', 'Nissan', 'Pulsar', 2016, 0, 24989, 3, 9);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('TGY683', 'Nissan', 'Navara', 2016, 0, 54989, 13, 1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('GTF098', 'Honda', 'Accord', 2012, 1, 12995, 110000, 9);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('OMG765', 'Mazda', '323', 2014, 1, 17995, 59000, 7);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('LOL201', 'Nissan', 'Bluebird', 2013, 1, 19995, 41000, 1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('YTY561', 'Honda', 'Crossroad', 2010, 2, 25500, 95500, 2);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('MMH291', 'Nissan', 'Altima', 2013, 1, 25995, 5000, 1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('YQTA23', 'Toyota', 'Yaris', 2014, 1, 18990, 36000, 8);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('OUP887', 'BMW', '525i', 2010, 1, 32440, 29400, 9);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('QPL322', 'BMW', 'M3', 2009, 1, 44990, 69000, 3);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('ZQP311', 'Nissan', 'Skyline', 2012, 2, 22980, 65040, 3);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('QPLO31', 'Nissan', 'Skyline', 2014, 1, 38500, 6500, 1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('POOE23', 'Honda', 'Accord', 2013, 1, 13995, 120000, 7);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('PLKI21', 'Mazda', '323', 2014, 1, 16995, 79000, 2);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('HWY231', 'Nissan', 'Bluebird', 2013, 2, 19995, 39000, 1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('SHEEPS', 'Holden', 'Commodore SsV', 2013, 1, 29995, 49000, 3);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('TPGEAR', 'Nissan', 'Bluebird', 2011, 2, 12995, 89000, 4);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('YTT342', 'Honda', 'VTR250F', 2011, 2, 9995, 89000, 1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('GO4T53', 'Honda', 'CB900F Hornet', 2010, 2, 7995, 95000, 5);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('GO4T99', 'Honda', 'Jazz', 2011, 2, 7995, 89000, 4);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('EATPOO', 'BMW', 'Z4', 2012, 1, 39995, 49000, 4);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('KFK122', 'Ford', 'Falcon', 2013, 1, 24995, 59000, 3);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('KER123', 'Mazda', 'Familia', 2010, 2, 8995, 85900, 1);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('PKR111', 'Mazda', 'Familia', 2010, 2, 7995, 95000, 5);
INSERT INTO vehicles (v_regno, v_make, v_model, v_year, v_numowners, v_price, v_mileage, c_no) VALUES ('FJW125', 'Holden', 'Commodore', 2015, 1, 39999, 15000, 4);


SELECT v_regno, v_make, v_model, c_colour FROM vehicles, lu_colours WHERE vehicles.c_no = lu_colours.c_no;
```

2.Modify the second script (4.1 and 4.2 Look for section - - Question 4.1) to work in MSSQL Server correctly. The original instructions given to create the Oracle script were: Create a procedure uspAddPurchaseSale_XX() which inserts data into the Sales Purchases table.The procedure performs the following act.

```
-- PART B --

--#############################################################################################--
-- Questions 4.1 and 4.2 --

-- Question 4.1--

CREATE PROCEDURE AddPurchaseSale
    @in_sp_datesold DATE,
    @in_sp_deposit DECIMAL(18, 2),
    @in_sp_ID VARCHAR(50),
    @in_c_id INT,
    @in_v_regno VARCHAR(20)
AS
BEGIN
    BEGIN TRANSACTION;

    BEGIN TRY
        DECLARE @max_invoice INT;

        INSERT INTO sales_purchases (sp_datesold, sp_saleprice, sp_deposit, sp_id, c_id, v_regno)
        SELECT @in_sp_datesold, v.v_price, @in_sp_deposit, @in_sp_ID, @in_c_id, @in_v_regno
        FROM vehicles v
        WHERE v.v_regno = @in_v_regno;

        SELECT @max_invoice = MAX(sp_invoice)
        FROM sales_purchases;

        UPDATE sales_purchases
        SET sp_addncost = sp_saleprice * 0.2
        WHERE sp_invoice = @max_invoice;

        UPDATE sales_purchases
        SET sp_total = sp_saleprice + sp_addncost - sp_deposit
        WHERE sp_invoice = @max_invoice;

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        THROW;
    END CATCH;
END;
GO
```

```
-- Execute the procedure and display sales_purchases before and after--
SELECT * FROM sales_purchases;

EXEC AddPurchaseSale '2017-06-02', 2000, 'MK201', 1, 'FJW125';

SELECT * FROM sales_purchases;

BEGIN TRANSACTION;
ROLLBACK TRANSACTION;
GO

-- Find unsold vehicles--

SELECT V_REGNO
FROM VEHICLES
EXCEPT
SELECT V_REGNO
FROM SALES_PURCHASES;
GO
```

```sql
-- Question 4.2 --
CREATE PROCEDURE AddPurchaseOrderItem
    @oid INT,
    @ino INT,
    @qty INT
AS
BEGIN
    BEGIN TRANSACTION;

    BEGIN TRY
        INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty)
        SELECT @oid, @ino, items.i_make, items.i_model, items.i_price, items.i_year, @qty
        FROM items
        WHERE items.i_no = @ino;

        UPDATE order_lines
        SET ol_subtotal = ol_qty * i_price
        WHERE o_id = @oid;

        UPDATE orders
        SET o_total = (
            SELECT SUM(ol_subtotal)
            FROM order_lines
            WHERE order_lines.o_id = @oid);

        UPDATE orders
        SET o_totalqty = (
            SELECT SUM(ol_qty)
            FROM order_lines
            WHERE order_lines.o_id = @oid);

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        THROW;
    END CATCH;
END;
GO


-- Insert into orders--

INSERT INTO orders (o_date, s_code, sp_id)
VALUES ('2021-06-05', 'XTRQC', 'MK201');
GO
```

2.1. Accepts parameters to insert data into the Sales Purchase table. Data that is pulled from another table should not be listed as a parameter.

```
--2.1 Accepts parameters to insert data into the Sales Purchase table.  Data that is pulled from another table should not be listed as a parameter  --

IF OBJECT_ID('uspAddPurchaseSale_XX', 'P') IS NOT NULL
    DROP PROCEDURE uspAddPurchaseSale_XX;
GO
CREATE PROCEDURE uspAddPurchaseSale_XX
    @sp_datesold DATE,
    @sp_saleprice DECIMAL(7,2),
    @sp_deposit DECIMAL(7,2),
    @sp_id VARCHAR(5),
    @c_id INT,
    @v_regno VARCHAR(7)
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO sales_purchases (sp_datesold, sp_saleprice, sp_deposit, sp_id, c_id, v_regno)
    VALUES (@sp_datesold, @sp_saleprice, @sp_deposit, @sp_id, @c_id, @v_regno);


    UPDATE sales_purchases
    SET sp_addncost = sp_saleprice * 0.2,
        sp_total = (sp_saleprice + (sp_saleprice * 0.2)) - sp_deposit
    WHERE sp_id = @sp_id AND c_id = @c_id AND v_regno = @v_regno;
END;
GO
```

2.2. Substitute the parameters for the data values in the INSERT ….. SELECT statement

```
  --2.2. Substitute the parameters for the data values in the INSERT ….. SELECT statement--


IF OBJECT_ID('uspAddPurchaseSale_XX', 'P') IS NOT NULL
    DROP PROCEDURE uspAddPurchaseSale_XX;
GO

CREATE PROCEDURE uspAddPurchaseSale_XX
    @sp_datesold DATE,
    @sp_saleprice DECIMAL(7,2),
    @sp_deposit DECIMAL(7,2),
    @sp_id VARCHAR(5),
    @c_id INT,
    @v_regno VARCHAR(7)
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO sales_purchases (sp_datesold, sp_saleprice, sp_deposit, sp_id, c_id, v_regno)
    SELECT @sp_datesold, @sp_saleprice, @sp_deposit, @sp_id, @c_id, @v_regno;

    UPDATE sales_purchases
    SET sp_addncost = @sp_saleprice * 0.2,
        sp_total = (@sp_saleprice + (@sp_saleprice * 0.2)) - @sp_deposit
    WHERE sp_id = @sp_id AND c_id = @c_id AND v_regno = @v_regno;
END;
GO
```

2.3. Add the UPDATE formulae for the Additional Cost and Total fields from the assignment script

```
--2.3. Add the UPDATE formulae for the Additional Cost and Total fields from the assignment script --


IF OBJECT_ID('uspAddPurchaseSale_XX', 'P') IS NOT NULL
    DROP PROCEDURE uspAddPurchaseSale_XX;
GO


CREATE PROCEDURE uspAddPurchaseSale_XX
    @sp_datesold DATE,
    @sp_saleprice DECIMAL(7,2),
    @sp_deposit DECIMAL(7,2),
    @sp_id VARCHAR(5),
    @c_id INT,
    @v_regno VARCHAR(7)
AS
BEGIN
    SET NOCOUNT ON;


    INSERT INTO sales_purchases (sp_datesold, sp_saleprice, sp_deposit, sp_id, c_id, v_regno)
    SELECT @sp_datesold, @sp_saleprice, @sp_deposit, @sp_id, @c_id, @v_regno;


    UPDATE sales_purchases
    SET sp_addncost = @sp_saleprice * 0.2,
        sp_total = (@sp_saleprice + (@sp_saleprice * 0.2)) - @sp_deposit
    WHERE sp_id = @sp_id AND c_id = @c_id AND v_regno = @v_regno;
END;
GO
```

2.4. IF the purchase is not successful, for any reason, then any errors should display an appropriate error message and ROLLBACK the transaction.

```
-- 2.4. IF the purchase is not successful, for any reason, then any errors should display an appropriate error message and ROLLBACK the transaction. --


IF OBJECT_ID('uspAddPurchaseSale_XX', 'P') IS NOT NULL
    DROP PROCEDURE uspAddPurchaseSale_XX;
GO

CREATE PROCEDURE uspAddPurchaseSale_XX
    @sp_datesold DATE,
    @sp_saleprice DECIMAL(7,2),
    @sp_deposit DECIMAL(7,2),
    @sp_id VARCHAR(5),
    @c_id INT,
    @v_regno VARCHAR(7)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY

        BEGIN TRANSACTION;

        INSERT INTO sales_purchases (sp_datesold, sp_saleprice, sp_deposit, sp_id, c_id, v_regno)
        VALUES (@sp_datesold, @sp_saleprice, @sp_deposit, @sp_id, @c_id, @v_regno);

        UPDATE sales_purchases
        SET sp_addncost = @sp_saleprice * 0.2,
            sp_total = (@sp_saleprice + (@sp_saleprice * 0.2)) - @sp_deposit
        WHERE sp_id = @sp_id AND c_id = @c_id AND v_regno = @v_regno;

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;

        DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;
        SELECT
            @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(),
            @ErrorState = ERROR_STATE();

        RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
    END CATCH;
END;
GO
```

2.5. Test the procedure with test data to show it works.

2.5.1. Perform a SELECT query on Sales_Purchases

```
-- 2.5.1. Perform a SELECT query on Sales_Purchases--

SELECT * FROM sales_purchases;
```

	sp_invoice	sp_datesold	sp_saleprice	sp_addncost	sp_deposit	sp_total	sp_id	c_id	v_regno
1	10000000	2015-06-17	14599.00	2919.80	2000.00	15518.80	MK201	1	GKN534
2	10000001	2015-06-27	15995.00	3199.00	2000.00	17194.00	MK201	2	ALP394
3	10000002	2015-07-07	24990.00	4998.00	5000.00	24988.00	MK201	3	NT6776
4	10000003	2015-07-11	36990.00	7398.00	1500.00	42888.00	MK201	4	KGH334
5	10000004	2015-07-17	46990.00	9398.00	0.00	56388.00	KK634	5	PHG902
6	10000005	2015-07-19	9995.00	1999.00	0.00	11994.00	MK201	6	GLM123
7	10000006	2015-08-07	12995.00	2599.00	1500.00	14094.00	KK634	7	OM1122
8	10000007	2015-08-10	13995.00	2799.00	3000.00	13794.00	KK634	8	RS3456
9	10000008	2015-08-27	9995.00	1999.00	0.00	11994.00	BP301	9	ZHU123
10	10000009	2015-10-01	41885.00	8377.00	0.00	50262.00	BP301	10	PRH345
11	10000010	2015-10-11	17995.00	3599.00	0.00	21594.00	MK201	11	SUT143
12	10000011	2015-11-10	39995.00	7999.00	2000.00	45994.00	BP301	12	GBR553
13	10000012	2015-12-11	20990.00	4198.00	3000.00	22188.00	KC312	13	LQRT67
14	10000013	2015-12-12	27440.00	5488.00	3000.00	29928.00	KC312	14	SALES1
15	10000014	2016-01-12	54990.00	10998.00	2500.00	63488.00	BP301	15	LMV541
16	10000015	2016-01-15	24989.00	4997.80	2000.00	27986.80	KC312	1	NIS123
17	10000016	2016-03-17	54989.00	10997.80	3000.00	62986.80	KH981	2	TGY683
18	10000017	2016-03-20	12995.00	2599.00	2000.00	13594.00	KK634	3	GTF098
19	10000018	2016-04-20	17995.00	3599.00	0.00	21594.00	KH981	4	OMG765
20	10001019	2017-06-02	39999.00	7999.80	2000.00	45998.80	MK201	1	FJW125

2.5.2. Call the uspAddPurchaseSale_XX() procedure with suitable test data

```
--2.5.2. Call the uspAddPurchaseSale_XX() procedure with suitable test data--

EXEC uspAddPurchaseSale_XX '2024-07-27', 19000.00, 4000.00, 'MK209', 1, 'LHZ626';
```

2.5.3. Perform a SELECT query on Sales_Purchases again

```
--2.5.3. Perform a SELECT query on Sales_Purchases again--

SELECT * FROM sales_purchases;
```

	sp_invoice	sp_datesold	sp_saleprice	sp_addncost	sp_deposit	sp_total	sp_id	c_id	v_regno
2	10000002	2015-06-27	15995.00	3199.00	2000.00	17194.00	MK201	2	ALP394
3	10000003	2015-07-07	24990.00	4998.00	5000.00	24988.00	MK201	3	NT6776
4	10000004	2015-07-11	36990.00	7398.00	1500.00	42888.00	MK201	4	KGH334
5	10000005	2015-07-17	46990.00	9398.00	0.00	56388.00	KK634	5	PHG902
6	10000006	2015-07-19	9995.00	1999.00	0.00	11994.00	MK201	6	GLM123
7	10000007	2015-08-07	12995.00	2599.00	1500.00	14094.00	KK634	7	OM1122
8	10000008	2015-08-10	13995.00	2799.00	3000.00	13794.00	KK634	8	RS3456
9	10000009	2015-08-27	9995.00	1999.00	0.00	11994.00	BP301	9	ZHU123
10	10000010	2015-10-01	41885.00	8377.00	0.00	50262.00	BP301	10	PRH345
11	10000011	2015-10-11	17995.00	3599.00	0.00	21594.00	MK201	11	SUT143
12	10000012	2015-11-10	39995.00	7999.00	2000.00	45994.00	BP301	12	GBR553
13	10000013	2015-12-11	20990.00	4198.00	3000.00	22188.00	KC312	13	LQRT67
14	10000014	2015-12-12	27440.00	5488.00	3000.00	29928.00	KC312	14	SALES1
15	10000015	2016-01-12	54990.00	10998.00	2500.00	63488.00	BP301	15	LMV541
16	10000016	2016-01-15	24989.00	4997.80	2000.00	27986.80	KC312	1	NIS123
17	10000017	2016-03-17	54989.00	10997.80	3000.00	62986.80	KH981	2	TGY683
18	10000018	2016-03-20	12995.00	2599.00	2000.00	13594.00	KK634	3	GTF098
19	10000019	2016-04-20	17995.00	3599.00	0.00	21594.00	KH981	4	OMG765
20	10001039	2017-06-02	39999.00	7999.80	2000.00	45998.80	MK201	1	FJW125

2.5.4. Perform a ROLLBACK so that the database returns to its original state

```
--2.5.4. Perform a ROLLBACK so that the database returns to its original state--

BEGIN TRANSACTION;

EXEC uspAddPurchaseSale_XX '2024-07-27', 19000.00, 4000.00, 'MK209', 1, 'LHZ626';

SELECT * FROM sales_purchases;

ROLLBACK TRANSACTION;

SELECT * FROM sales_purchases;
GO
```

3. Modify the second script (4.1 and 4.2 Look for section - - Question 4.2) to work in MSSQL Server correctly. The original instructions given to create the Oracle script were: Create a procedure called uspPurchaseOrderItem_XX().This procedure inserts data into the ORDER_LINES table. The procedure performs the following:

3.1. Accepts parameters to insert data into the ORDER_LINES table – make sure to choose appropriate parameters again

```
--3.1. Accepts parameters to insert data into the ORDER_LINES table – make sure to choose appropriate parameters again--

IF OBJECT_ID('uspPurchaseOrderItem_XX', 'P') IS NOT NULL
    DROP PROCEDURE uspPurchaseOrderItem_XX;
GO

CREATE PROCEDURE uspPurchaseOrderItem_XX
    @o_id INT,
    @i_no INT,
    @i_make VARCHAR(10),
    @i_model VARCHAR(15),
    @i_price DECIMAL(7,2),
    @i_year INT,
    @ol_qty TINYINT
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY

        BEGIN TRANSACTION;

        INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty)
        VALUES (@o_id, @i_no, @i_make, @i_model, @i_price, @i_year, @ol_qty);

        UPDATE order_lines
        SET ol_subtotal = ol_qty * i_price
        WHERE o_id = @o_id AND i_no = @i_no;

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;

        DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;
        SELECT
            @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(),
            @ErrorState = ERROR_STATE();

        RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
    END CATCH;
END;
GO
```

3.2. Substitute the parameters for the data values in the INSERT …. SELECT statement

```
--3.2. Substitute the parameters for the data values in the INSERT …. SELECT statement --

IF OBJECT_ID('uspPurchaseOrderItem_XX', 'P') IS NOT NULL
    DROP PROCEDURE uspPurchaseOrderItem_XX;
GO

CREATE PROCEDURE uspPurchaseOrderItem_XX
    @o_id INT,
    @i_no INT
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty)
        SELECT @o_id, i_no, i_make, i_model, i_price, i_year, 1
        FROM items
        WHERE i_no = @i_no;

        UPDATE order_lines
        SET ol_subtotal = ol_qty * i_price
        WHERE o_id = @o_id AND i_no = @i_no;

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;

        DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;
        SELECT
            @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(),
            @ErrorState = ERROR_STATE();

        RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
    END CATCH;
END;
GO
```

3.3. Add the UPDATE formulae for the Subtotal and UPDATE formulae for the ORDERS table from the assignment script.

```
--3.3. Add the UPDATE formulae for the Subtotal and UPDATE formulae for the ORDERS table from the assignment script--

IF OBJECT_ID('uspPurchaseOrderItem_XX', 'P') IS NOT NULL
    DROP PROCEDURE uspPurchaseOrderItem_XX;
GO

CREATE PROCEDURE uspPurchaseOrderItem_XX
    @o_id INT,
    @i_no INT,
    @i_make VARCHAR(10),
    @i_model VARCHAR(15),
    @i_price DECIMAL(7,2),
    @i_year INT,
    @ol_qty TINYINT = 1
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        INSERT INTO order_lines (o_id, i_no, i_make, i_model, i_price, i_year, ol_qty)
        VALUES (@o_id, @i_no, @i_make, @i_model, @i_price, @i_year, @ol_qty);

        UPDATE order_lines
        SET ol_subtotal = @ol_qty * @i_price
        WHERE o_id = @o_id AND i_no = @i_no;

        UPDATE orders
        SET o_total = (
            SELECT SUM(ol_subtotal)
            FROM order_lines
            WHERE orders.o_id = order_lines.o_id
        ),
        o_totalqty = (
            SELECT SUM(ol_qty)
            FROM order_lines
            WHERE orders.o_id = order_lines.o_id
        )
        WHERE o_id = @o_id;


        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;

        DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;
        SELECT
            @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(),
            @ErrorState = ERROR_STATE();

        RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
    END CATCH;
END;
GO
```

3.4. Test the procedure with test data to show it works – create an order first, and then test the procedure as such:

3.4.1. Perform a SELECT query on orders

```
SELECT * FROM orders;
```

	o_id	o_date	o_totalqty	o_total	s_code	sp_id
1	80000000	2015-06-01	10	71500.00	XTRQC	MK201
2	80000001	2015-06-05	12	103500...	XTRQC	MK201
3	80000002	2015-06-06	6	56000.00	XTRQC	MK201
4	80000003	2015-06-10	6	45500.00	XTRQC	MK201
5	80000004	2015-06-11	6	41500.00	XTRQC	MK201
6	80000005	2015-06-12	9	60500.00	XTRQC	MK201
7	80000006	2015-07-21	8	117000...	XTRQC	KK634
8	80000007	2015-08-01	4	45000.00	XTRQC	KK634
9	80000008	2015-08-11	3	34000.00	XTRQC	KK634
10	80000009	2015-08-21	10	109500...	XTRQC	KK634
11	80000010	2024-06-07	NULL	NULL	XTRQC	MK201
12	80000011	2024-06-07	NULL	NULL	XTRQC	MK201
13	80000012	2024-06-07	NULL	NULL	XTRQC	MK201
14	80000013	2024-06-07	NULL	NULL	XTRQC	MK201
15	80000014	2024-06-07	NULL	NULL	XTRQC	MK201
16	80000015	2024-06-07	NULL	NULL	XTRQC	MK201
17	80000016	2024-06-07	NULL	NULL	XTRQC	MK201
18	80000017	2024-06-07	2	6000.00	XTRQC	MK201
19	80000018	2024-06-07	2	6000.00	XTRQC	MK201
20	80000019	2024-06-07	2	6000.00	XTRQC	MK201
21	80001010	2021-06-05	NULL	NULL	XTRQC	MK201
22	80001011	2021-06-05	NULL	NULL	XTRQC	MK201
23	80001012	2024-06-07	2	6000.00	XTRQC	MK201

3.4.2. Create an ORDER so that you can get an order number

```
DECLARE @NewOrderID INT;

INSERT INTO orders (o_date, s_code, sp_id)
VALUES ('2024-06-07', 'XTRQC', 'MK201');


SET @NewOrderID = SCOPE_IDENTITY();
```

3.4.3. Perform a SELECT query on your order to get the order number

```
SELECT * FROM orders WHERE o_id = @NewOrderID;
```

	o_id	o_date	o_totalqty	o_total	s_code	sp_id
1	80001015	2024-06-07	NULL	NULL	XTRQC	MK201

3.4.4. Call the uspAddPurchaseOrderItem_XX() with your order number and other required parameters

```
EXEC uspPurchaseOrderItem_XX
    @o_id = @NewOrderID,
    @i_no = 1,
    @i_make = 'Honda',
    @i_model = 'Accord',
    @i_price = 3000.00,
    @i_year = 2010,
    @ol_qty = 2;
```

3.4.5. Perform a SELECT query on ORDER_LINES and ORDERS based on your order_number

```
SELECT * FROM order_lines WHERE o_id = @NewOrderID;
SELECT * FROM orders WHERE o_id = @NewOrderID;
```

	o_id	i_no	i_make	i_model	i_price	i_year	ol_qty	ol_subtotal
1	80001016	1	Honda	Accord	3000.00	2010	2	6000.00

	o_id	o_date	o_totalqty	o_total	s_code	sp_id
1	80001016	2024-06-07	2	6000.00	XTRQC	MK201

3.4.6. Perform a ROLLBACK so that the database returns to its original state

```
-------------------------
ROLLBACK TRANSACTION;
GO
```

4. 5. Create the following business rules

4.1. A supervisor can supervise no more than 2 people. This trigger should occur on INSERT and UPDATE from the sales persons table.

4.2. Modify the script to work in MSSQL Server correctly.

```
-- Question 5 Trigger--
-- Create the trigger--
--4.1 and 4.2 Create and Modify the Trigger for MSSQL Server--

CREATE TRIGGER trg_sales
ON sales_persons
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @count_sale INT;
    DECLARE @sp_sup VARCHAR(50);

    SELECT @sp_sup = sp_sup FROM INSERTED;

    SELECT @count_sale = COUNT(sp_id)
    FROM sales_persons
    WHERE sp_sup = @sp_sup;

    IF @count_sale > 2
    BEGIN
    ROLLBACK TRANSACTION; -- Rollback the transaction
        THROW 50000, 'INSERT DENIED: A supervisor cannot supervise more than 2 people', 1;

        RETURN; -- Exit the trigger
    END;
END;
GO


-- Demonstrating the trigger--

BEGIN TRY

    INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup)
    VALUES ('PW501', 'Mavis', 'Halmer', '2015-08-24', '0219338123', 0.15, 'MK201');
END TRY
BEGIN CATCH
    PRINT ERROR_MESSAGE();
END CATCH;
GO
```

4.3. Test that the trigger is working by inserting rows into Sales_Persons with the same supervisor. Use ROLLBACK to return the database to its original state when testing is complete.

```
--4.3. Test that the trigger is working by inserting rows into Sales_Persons with the same supervisor.
--Use ROLLBACK to return the database to its original state when testing is complete.--

SELECT * FROM sales_persons;

------------------------------------------------

BEGIN TRANSACTION;
BEGIN TRY
    -- Insert statement to test the trigger
    INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup)
    VALUES ('PW501', 'George', 'Russell', '2015-08-24', '0219338123', 0.15, 'MK201');

    INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup)
    VALUES ('PW502', 'John', 'Cena', '2015-08-24', '0219338124', 0.15, 'MK201');

    -- This should trigger the
    INSERT INTO sales_persons (sp_id, sp_fname, sp_lname, sp_startdate, sp_cellph, sp_comrate, sp_sup)
    VALUES ('PW503', 'Jane', 'Jackson', '2015-08-24', '0219338125', 0.15, 'MK201');
END TRY
BEGIN CATCH
    -- Display error message
    PRINT ERROR_MESSAGE();
END CATCH;

ROLLBACK TRANSACTION;

------------------------------------------------

SELECT * FROM sales_persons;

------------------------------------------------

------------------------------------------------

SELECT * FROM sales_persons;
```

	sp_id	sp_fname	sp_lname	sp_startdate	sp_cellph	sp_comrate	sp_sup
1	BP301	Bradley	Palmer	2015-08-24	0219878123	0.15	MK201
2	JW351	John	Wrights	2016-03-20	0210998212	0.15	
3	KC312	Karen	Craften	2015-08-21	0213940903	0.25	KK634
4	KH981	Kane	Hunter	2016-03-12	0212132231	0.15	MK201
5	KK634	Kelly	Knapp	2015-07-10	0213390823	0.15	
6	MK201	Michael	Knapp	2015-06-10	0213390823	0.25	

5.

Explanation for how the stored procedures work together for PurchaseSale:

Firstly the parameters procedure takes the parameters for the date of sale, amount, sale person ID customerID and rego number.

Then the insert operation inserts a new record into the Sales purchases table and using the provided parameters. And the Commit command is there so that the transaction is committed and if there is an error the transaction is rolled back.

Explanation for PurchaseorderItem:

Like the PurchaseSale procedure the PurchaseOrderItem procedure takes the parameters to accept the oderID, item number and quantity.

After that the insert operation takes the new records into the order_Line table using the given parameters. The items are then fetched into the items table based on the item number. Like the purchasesale procedure the PurchaseorderItem takes the transaction committed if there is an error then the transaction is rolled back.

The update operation is there to update the items with the subtotal, total and total quantity of the purchase items.

APA referencing

- Chatgpt
- https://www.youtube.com/watch?v=f6VWSlnHGCE&ab_channel=edureka%21
- https://www.youtube.com/watch?v=PkrjuQ4dNnk&ab_channel=InternetAuthoring
- https://www.mssqltips.com/sqlservertip/6132/create-alter-drop-and-execute-sql-server-stored-procedures/