This is an open book assessment. You are encouraged to refer to your notes and online resources. ~~Do not confer with other people and do not use AI-assisted solutions.~~

Six questions. Maximum 10 points for each question. Points will be awarded for correctness and for the simplicity and elegance of your code. Do not use variables in your answers.

1. Create the following table and data:

```
CREATE TABLE parkinglot (txt VARCHAR(500) NOT NULL PRIMARY KEY);

INSERT INTO parkinglot (txt) VALUES
 ('354FKH/JKELLER/2022-05-17$64.50')
,('BMR3408/RJEFFREY/2022-05-18$101.90')
,('KFD478/PMURPHY/2022-05-16$35.85')
,('KFD918/JRIMBAUD/2022-05-16$21.50')
,('PA42391/RDYLAN/2022-05-17$21.50')
,('RA8H5G/AROBINSON/2022-05-17$27.50')
,('RX64421/KSMITH/2022-05-16$21.50');
```

   Write a query to extract four columns from the above data based on the following assumptions: **vehicle** will be the string that comes before the first / character; **name** the string between the two /s and immediately before the date; **dt** the date, which is always in the format YYYY-MM-DD; **amount** which is at the end of the string, always preceded by the $ sign.

   In your query, convert the amount column to a DECIMAL and the dt column to a DATE. The result of your query should be:

```
vehicle  name       dt          amount
-------- ---------- ----------- -------
354FKH   JKELLER    2022-05-17  64.50
BMR3408  RJEFFREY   2022-05-18  101.90
KFD478   PMURPHY    2022-05-16  35.85
KFD918   JRIMBAUD   2022-05-16  21.50
PA42391  RDYLAN     2022-05-17  21.50
RA8H5G   AROBINSON  2022-05-17  27.50
RX64421  KSMITH     2022-05-16  21.50
```

2. Create the following tables and data about regions and cases of infections.

```
CREATE TABLE region (region_code varchar(10) NOT NULL PRIMARY KEY, region_name varchar(50)
NOT NULL, population int NOT NULL);

CREATE TABLE infection (region_code varchar(10) NOT NULL REFERENCES region (region_code), dt
date NOT NULL, new_cases DECIMAL(10,0) NOT NULL, PRIMARY KEY (region_code, dt));

INSERT INTO region (region_code, region_name ,population)
VALUES
 ('AA', 'Region 1 (AA)', 457040)
,('BF', 'Region 2 (BF)', 527280)
,('CM', 'Region 3 (CM)', 301680);

INSERT INTO infection (region_code,dt,new_cases)
VALUES
 ('AA', '2021-01-01', 226)
,('AA', '2021-01-15', 280)
,('AA', '2021-02-01', 220)
,('AA', '2021-02-15', 199)
,('BF', '2021-01-01', 140)
,('BF', '2021-01-15', 121)
,('BF', '2021-02-15', 104);
```

Write a query to show regions, dates, new_cases plus a percentage (new_cases as a percentage of the total population of the region), and the cumulative total of cases (total new_cases for the region up until that date). The number of decimal places in the percentage is not important - no need to round it. The result shoul be as follows:

```
region_code region_name      dt         new_cases  pct          cumulative_total
----------- ---------------- ---------- ---------- ------------ -----------------
AA          Region 1 (AA)    2021-01-01 226        0.049        226
AA          Region 1 (AA)    2021-01-15 280        0.061        506
AA          Region 1 (AA)    2021-02-01 220        0.048        726
AA          Region 1 (AA)    2021-02-15 199        0.044        925
BF          Region 2 (BF)    2021-01-01 140        0.027        140
BF          Region 2 (BF)    2021-01-15 121        0.023        261
BF          Region 2 (BF)    2021-02-15 104        0.020        365
```

3. Create the following table of buyers and sellers. Write a query that returns all the buyers and sellers in a single column with the date and a role column that identifies whether the user is a buyer or a seller. The required output is shown below.

```
CREATE TABLE trade (dt DATETIME2(0) NOT NULL PRIMARY KEY, seller_username VARCHAR(50) NOT
NULL, buyer_username VARCHAR(50) NOT NULL);

INSERT INTO trade (dt, seller_username, buyer_username) VALUES
 ('2019-02-01 08:14:31','wcollins','rjones')
,('2019-02-01 09:22:57','esmith','zchandler')
,('2019-02-01 09:31:39','kthomas','jbennet')
,('2019-02-01 10:21:22','rjones','esmith');
```

```
username   dt                          role
---------- --------------------------- ------
wcollins   2019-02-01 08:14:31         SELLER
esmith     2019-02-01 09:22:57         SELLER
kthomas    2019-02-01 09:31:39         SELLER
rjones     2019-02-01 10:21:22         SELLER
rjones     2019-02-01 08:14:31         BUYER
zchandler  2019-02-01 09:22:57         BUYER
jbennet    2019-02-01 09:31:39         BUYER
esmith     2019-02-01 10:21:22         BUYER
```

4. The **loc** table defined below defines some geographical locations and various sub-locations contained within them. Each location except USA has a “parent” location that contains it. Write a query that returns every location with a count of how many sublocations it has directly within it. Your answer should be as follows:

```
code       name                cnt
---------- ------------------- -----------
ABY        Albany              0
BK         Brooklyn            0
BX         Bronx               0
MN         Manhattan           0
NY         New York State      2
NYC        New York City       5
QN         Queens              0
SI         Staten Island       0
USA        United States       1
```

```
CREATE TABLE loc (code VARCHAR(10) NOT NULL PRIMARY KEY, name VARCHAR(20) NOT NULL UNIQUE,
parent_loc VARCHAR(10) NULL REFERENCES loc (code));

INSERT INTO loc (code, name, parent_loc) VALUES
 ('USA', 'United States',NULL)
,('NY', 'New York State','USA')
,('NYC', 'New York City','NY')
,('MN', 'Manhattan','NYC')
,('BX', 'Bronx','NYC')
,('BK', 'Brooklyn','NYC')
,('QN', 'Queens','NYC')
,('SI', 'Staten Island','NYC')
,('ABY', 'Albany','NY');
```

5. Create the following tables and data:

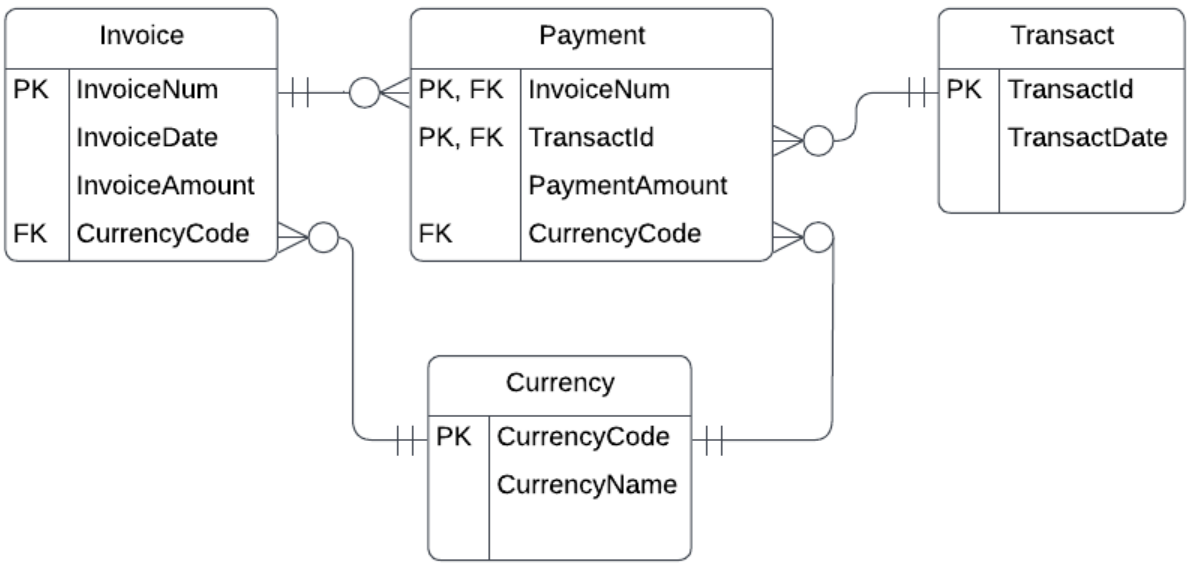
```
CREATE TABLE asset (asset_num INTEGER NOT NULL PRIMARY KEY, account_num VARCHAR(10) NOT NULL,
currency_code CHAR(3) NOT NULL, amount DECIMAL(10,2));

INSERT INTO asset (asset_num, account_num, currency_code, amount)
VALUES
 (11,'A32814','GBP', 82470)
,(12,'A70155','EUR', 92230)
,(13,'A83866','USD',268105)
,(14,'A32814','USD',191400)
,(15,'A70155','EUR',129000)
,(16,'A16786','HKD',300400)
,(17,'A70155','GBP',601000)
,(18,'A32814','EUR', 45500)
,(19,'A83866','EUR', 23850)
,(20,'A83866','EUR',118090);
```

Write a query that returns one row per account_num and currency code, but only include the result the rows for the account numbers (identified by account_num) that have rows for both EUR and GBP currency_codes. Include the total amount and a count of the number of rows in the original table. Your result should have 5 rows as follows:

```
account_num currency_code amount       cnt
----------- ------------- ------------ -----
A32814      EUR           45500.00     1
A32814      GBP           82470.00     1
A32814      USD           191400.00    1
A70155      EUR           221230.00    2
A70155      GBP           601000.00    1
```

6. Create four tables as shown in the following ER diagram. Your tables should include primary and foreign keys and you must specify suitable data types for each column. All columns should be non-nullable.



THE END 😊