Question 1

Write a regular expression to validate addresses. You only have to accommodate the addresses in the list.

Example to validate	**Valid?**
15 Riversdale Rd, Hawthorn 3122 Vic	Yes
15 Riversdale Rd Hawthorn 3122 Vic	No
18 Ocean St, Woollahra 2025 NSW	No
18 Ocean St, Woollahra 2025 NSW	Yes
1025 Bass Hw, Parklands 7320 Tas	Yes
1025 Bass Hw, Parklands 73200 Tas	No
20 Darwin Gv, Highgate 6003 WA	Yes
20 Darwin Gv, Highgate WA	No
20 Darwin Gv,Highgate 6003 WA	No
20 Kent St, Rockhampton 4700 Qld	Yes
20 Kent St, Rockhampton 4700 Qdd	No
20 Kent St, Rockhampton4700 Qld	No

Implement the following rules

1. House number is mandatory and has one to four numbers and must be followed by a space.
2. A street name starts with a capital letter followed by one or more lower case letters followed by a space. No need to accommodate names like ‘McLaren St’ (but you can if you like).
3. A street type of either Rd, St, Gv or Hw (no need to add other possibilities), followed by a comma, followed by a space.
4. A post code that consists of four numbers (no need to check if it corresponds to the state) followed by a space.
5. A state abbreviation of either Vic, NSW, Tas, WA or Qld.

Your answer:

\d{1-4} [A-Z][a-z]+ [Rd|St|Gv|Hw], [A-Z][a-z]+ \d{4} [Vic|NSW|Tas|WA|Qld]

Question 2

Consider the following XQuery:

It produces the following output:

Write the smallest XML document that can produce this outcome. You don't have to add any elements not referenced in the query or output, meaning it has only the two messages shown and matching data.

Your answer:

<messages>

<message>

<date>

<day>1</day>

<month>January</month>

<year>2021</year>

</date>

<from>Kim</from>

<to>Thanh</to>

<msg>See you in 5 minutes</msg>

</message>

<message>

<date>

<day>1</day>

<month>January</month>

<year>2021</year>

</date>

<from>Joyce</from>

<to>Steve</to>

<msg>No time today</msg>

</message>

</messages>

Question 3

Consider the following XQuery:

It produces the following output:

Add another entry (message element) to your answer at 2a that would not be picked up by the XQuery. Just invent the data for the correctly named elements.

Your answer:

<messages>

<message>

<date>

<day>10</day>

<month>Auguest</month>

<year>2020</year>

</date>

<from>Tom</from>

<to>John</to>

<msg>Goodbye</msg>

</message>

</messages>

Question 4

Consider the following information given to you to build a database system that keeps records of the IT assets issued to staff and discipline areas:

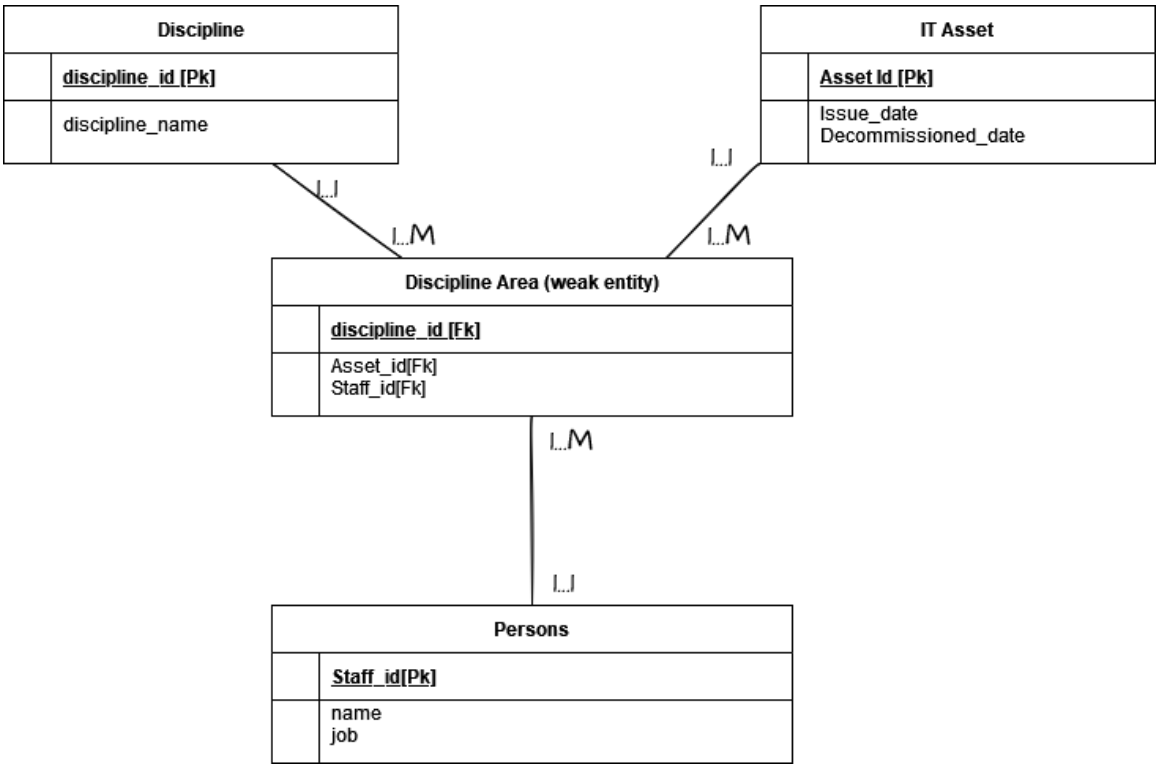
- Each IT asset has an id, an issue date and a decommissioned date. The decommissioned date is null for assets still in use.
- Each asset is used in a discipline area. There are disciplines like Business, Health and Science but also library and administration. Each discipline has an id and a name.
- Some, but not all, of the assets are issued to specific persons. Each person has a name, a staff id and a job, such as a librarian or lecturer. Each of the persons is attached to exactly one discipline area.

Based on the above, answer the following:

Draw the ER or UML diagram for the above. Mark the primary [PK] and foreign key columns [FK]. Show the cardinalities between the tables (relations).

You can use Powerpoint for the drawing, then save it as a picture and upload.

Your answer:



Question 5

Based on the diagram in Question 3a, write the DDL to implement all tables including data types, primary, foreign key constraints and specify if the attribute is nullable.

Your answer:

```
Create Database IT;
Use IT;


Create table Discipline
(
discipline_id int(5) UNSIGNED NOT NULL,

discipline_name varchar(10) NOT NULL,
PRIMARY KEY (discipline_id)

);
```

```
Create table Persons
(
staff_id int(5) UNSIGNED NOT NULL,

name varchar(10) NOT NULL,

job varchar(14) NOT NULL,
PRIMARY KEY (staff_id)

);

Create table IT_assets
(
Asset_id int(5) UNSIGNED NOT NULL,

Issue_date date NOT NULL,

Decommisioned_date date NOT NULL,
PRIMARY KEY (Asset_id)

);

Create table Discipline_area

(

discipline_id int(5) UNSIGNED NOT NULL,

Asset_id int(5) UNSIGNED NOT NULL,

staff_id int(5) UNSIGNED NOT NULL,

PRIMARY KEY (discipline_id, Asset_id, staff_id),

FOREIGN KEY (discipline_id) references Discipline(discipline_id),

FOREIGN KEY (Asset_id) references IT_assets (Asset_id),

FOREIGN KEY (staff_id) references Persons(staff_id),



)
```

Question 6

For each table created in questions 3a and 3b, write the SQL statements to insert 1 tuple (row) of data. **Specify the attribute for each item of data.**

Your answer:

```
INSERT INTO Asset (Asset_id, Issue_date, Decommisioned_date )

VALUES (12345,  "2021-10-14",  2025-10-10);


INSERT INTO Discipline (discipline_id, discipline_name)

VALUES (14567," Business");


INSERT INTO Person (staff_id,  name,  job)
 VALUES (15478,  "Bob",  "librarian");


INSERT INTO Discipline_area (discipline_id, Asset_id, staff_id)

 VALUES (14567, 12345, 15478);
```

Question 7

A university subject has enrolled students who are divided into tutorial groups for their practical work. Each tutorial has a code like 'TU-01', 'TU-02', etc. Each student, identified by their student ids, is allocated to exactly one tutorial. The tutorials have weekly start times. Each tutorial has one tutor who runs it. At the moment, we have a rather messy table that captures this information:

Are there any changes you might want to make to bring this table into first normal form?

You can describe the changes, you do not have to recreate the table.

Your answer:

There is repeating group of data in time and tutor name, so it will be better to make time into day and time and for tutor name it will be better to make it into tutor first name and tutor last name

Question 8

A university subject has enrolled students who are divided into tutorial groups for their practical work. Each tutorial has a code like 'TU-01', 'TU-02', etc. Each student, identified by their student ids, is allocated to exactly one tutorial. The tutorials have weekly start times. Each tutorial has one tutor who runs it. At the moment, we have a rather messy table that captures this information:

Divide the original table (**without applying the changes you suggested in Question 4a**) into two tables to bring it into second normal form (2NF) **but not 3NF**. List the attributes for each table and mark the primary key you suggest.

You can use Powerpoint to draw the table or simply write the tables and their attributes as lists in the provided space.

Your answer:

Tutor_id [PK]	Tutor_name
388777	Peter Nguyen
364533	Kumar Singh

Tute_id [PK]	Subj_id	time	student_id	Tutor_id [FK]
TU-01	IT3005	Wed 11am	113588	388777
TU-01	IT3005	Wed 11am	101573	388777
TU-02	CS1009	Thu 2PM	106843	364533
TU-02	CS1009	Thu 2PM	121153	364533

Question 9

Create another table to bring the result into 3NF. List all three tables.You can use Powerpoint to draw the table or simply write the tables and their attributes as lists in the provided space.

Your answer:

Tute_id [FK]	student_id
TU-01	113588
TU-01	101573
TU-02	106843
TU-02	121153

Tutor_id [PK]	Tutor_name
388777	Peter nguyen
364533	Kumar Singh

Tute_id [PK]	Subj_id [PK]	time	Tutor_id [FK]
TU-01	IT3005	Wed 11am	388777
TU-02	CS1009	Thu 2PM	364533

Question 10

Consider the following tables with data in them. A fruit and vegetable wholesale business sells produce to retail customers, that's why each sale item represents the sale of one single item of produce. The prices are given per kilogram. Sales quantities are kilograms.

Write an SQL statement to fetch the cultivar and price of the apples and pears.

Your answer:

```
Select cultivar, Name, Price
From FruitVeg
Where Name= 'Apple' or 'Pear';
```

Question 11

Consider the following tables with data in them. A fruit and vegetable wholesale business sells produce to retail customers, that's why each sale item represents the sale of one single item of produce. The prices are given per kilogram. Sales quantities are kilograms.

Write a query that shows the average price per fruit or vegetable. You can use the AVG() function. The output (result set) would look like:

Your answer:

```
Select Name, AVG(Price) as average_price
From FruitVeg
Group by Name;
```

Question 12

Consider the following tables with data in them. A fruit and vegetable wholesale business sells produce to retail customers, that's why each sale item represents the sale of one single item of produce. The prices are given per kilogram. Sales quantities are kilograms.

Write a query that shows how much money each customer has spent on each cultivar. (We have attributes of the same name in different tables, so you'll have to work with aliases.) You can use a view if you think it's easier. The output (result set) should look like:

Your answer:

```
Select first || ' ' || last as customer_name, cultivar, SUM(f.price*s.qty)as total_amount
From FruitVeg f Join Sales s Join Customer c
On f.id = s.fruit
On c.id = s.cus_id
Group by cultivar, customer_name;
```

Question 13

Consider the following tables with data in them. A fruit and vegetable wholesale business sells produce to retail customers, that's why each sale item represents the sale of one single item of produce. The prices are given per kilogram. Sales quantities are kilograms.

The output (result set) of question 5c should look like:

Thomas Brown is new and hasn't bought anything yet. How do you change the query at 5c to ensure he is included in the result?

Write the modified query as your answer.

Your answer:

```
Select first || ' ' || last as customer_name, cultivar, SUM(f.price*s.qty)as total_amount
From FruitVeg f Join Sales s Right Outer Customer c
On f.id = s.fruit
On c.id = s.cus_id
Group by cultivar,customer_name;
```

Question 14

Consider the following tables with data in them. A fruit and vegetable wholesale business sells produce to retail customers, that's why each sale item represents the sale of one single item of produce. The prices are given per kilogram. Sales quantities are kilograms.

The tables have been created sloppily with no foreign key constraint in place. Write the code to add the appropriate foreign key constraints. Assume the tables already exist in the database.

Your answer:

```
Alter table FruitVeg
ADD Constraint id Primary Key(id);

Alter table Sale
ADD Constraint id Primary Key(id),
ADD FOREIGN KEY (fruit) REFERENCES FruitVeg(id),
ADD FOREIGN KEY (cus_id) REFERENCES Customer(id);

Alter table Customer
ADD Constraint id Primary Key(id);
```

Question 15

Consider the following tables with data in them. A fruit and vegetable wholesale business sells produce to retail customers, that's why each sale item represents the sale of one single item of produce. The prices are given per kilogram. Sales quantities are kilograms.

We are assuming that the **foreign key constraints have now been added**, and that the data shown in the tables is **the only data**. Thomas Brown has passed his first order. Write an Insert statement that adds a new entry to the sales table. He has bought 500 Tomatoes 'Cherry Black' which have an id of 3666. The sale happened today. The sales id can be an autonumber.

Your answer:

Insert into Sale(id, cus_id,fruit, qty, date)
Values(1238, 365, 3666, 500, 2020-10-12);

Insert into FruitVeg(id, cultivar, name, price)
Values(3666, 'Cherry Black', Tomato, 5.80);

(gave a price similar to that of other fruits as price can't be null and price hasn't been stated in the question)

Question 16

Consider the following tables with data in them. A fruit and vegetable wholesale business sells produce to retail customers, that's why each sale item represents the sale of one single item of produce. The prices are given per kilogram. Sales quantities are kilograms.

Do you think the insertion in Question 5f will succeed? Why/why not?

Your answer:

It will succeed with the modification i had made with price and the parent table. Otherwise there would have been a foreign key error as we were trying to input data into child whereas the data wasn't present in the parent table in case of the new fruit tomato of cherry black