LSINF2335 - Programming Paradigms

Project 1 report

Youri Mouton

Question 1

> How are your tables/rows stored in the Prolog knowledge base?

Columns are stored as two arity facts, the first argument being a unique identifier, the first column of the table. The second argument contains the value.

```
client_localite("F400","Bruxelles").
```

Tables are simply a fact containing a list of its columns.

```
client([ncli, nom, adresse, localite, cat, compte]).
```

The database stores the tables in a list as well.

```
table_index([client, commande, detail, produit]).
```

Question 2

> As a query language, Prolog is much more powerful than SQL. What is the "mismatch" between SQL and Prolog that forces us to write code to handle the differences?

- Lots of the useful mathematical operations in SQL have to be done manually in Prolog.
- SQL is designed to work with files while Prolog stores data in memory, so persistence has to be thought. `SWI` Prolog has utilities to deal with this.

Question 3

/!\ Please name the file `db.pl` otherwise `cleanup` will not work.

i. Load/reset the example data.

```
cleanup.
```

ii. Create a new table.

```
create_table(person,[name,gender,age]).
```

iii. Insert a row in a table.

```
insert(person,["Youri Mouton",m,23]).
```

iv. Select all rows in a table.

```
select(produit,*).
```

```
   |npro|libelle|prix|qstock|
1  |[CS262,Chev. Sapin 200*6*2,75,45]
2  |[CS264,Chev. Sapin 200*6*4,120,2690]
3  |[CS464,Chev. Sapin 400*6*4,220,450]
4  |[PA60,Pointe Acier 60 (10K),95,134]
5  |[PS222,PL. Sapin 200*20*2,185,1220]
6  |[PA45,POINTE ACIER 45 (20K),105,580]
7  |[PH222,PL. HETRE 200x20x2,185,1220]
true.
```

v. Select all rows that match a predicate.

```
select_where(client,*,localite,=,"Namur").
```

```
   |ncli|nom|adresse|localite|cat|compte|
1  |[B062,Goffin,72, rue de la Gare,Namur,B2,-3200]
2  |[C123,MERCIER,25, rue Lemaitre,Namur,C1,-2300]
3  |[L422,Franck,60, rue de Wépion,Namur,C1,0]
4  |[S127,Vanderka,3, avenue des Roses,Namur,C1,-4580]
true.
```

vi. Select some columns from rows that match a predicate.

```
select_where(produit,[npro,qstock],prix,=<,120).
```

```
   |npro|qstock|
1  |[CS262,45]
2  |[CS264,2690]
3  |[PA60,134]
4  |[PA45,580]
true.
```

vii. Select all rows from a cross join between two tables.

```
cross_join(produit,commande).
```

```
   |npro|libelle|prix|qstock|ncom_co|ncli_co|date|
1  |[CS262,Chev. Sapin 200*6*2,75,45,30178,K111,2008-12-22]
2  |[CS262,Chev. Sapin 200*6*2,75,45,30179,C400,2008-12-22]
3  |[CS262,Chev. Sapin 200*6*2,75,45,30182,S127,2008-12-23]
4  |[CS262,Chev. Sapin 200*6*2,75,45,30184,C400,2008-12-23]
5  |[CS262,Chev. Sapin 200*6*2,75,45,30185,F011,2009-01-02]
6  |[CS262,Chev. Sapin 200*6*2,75,45,30186,C400,2009-01-02]
7  |[CS262,Chev. Sapin 200*6*2,75,45,30188,B512,2009-01-02]
8  |[CS264,Chev. Sapin 200*6*4,120,2690,30178,K111,2008-12-22]
9  |[CS264,Chev. Sapin 200*6*4,120,2690,30179,C400,2008-12-22]
10 |[CS264,Chev. Sapin 200*6*4,120,2690,30182,S127,2008-12-23]
11 |[CS264,Chev. Sapin 200*6*4,120,2690,30184,C400,2008-12-23]
12 |[CS264,Chev. Sapin 200*6*4,120,2690,30185,F011,2009-01-02]
13 |[CS264,Chev. Sapin 200*6*4,120,2690,30186,C400,2009-01-02]
14 |[CS264,Chev. Sapin 200*6*4,120,2690,30188,B512,2009-01-02]
15 |[CS464,Chev. Sapin 400*6*4,220,450,30178,K111,2008-12-22]
16 |[CS464,Chev. Sapin 400*6*4,220,450,30179,C400,2008-12-22]
17 |[CS464,Chev. Sapin 400*6*4,220,450,30182,S127,2008-12-23]
18 |[CS464,Chev. Sapin 400*6*4,220,450,30184,C400,2008-12-23]
19 |[CS464,Chev. Sapin 400*6*4,220,450,30185,F011,2009-01-02]
20 |[CS464,Chev. Sapin 400*6*4,220,450,30186,C400,2009-01-02]
21 |[CS464,Chev. Sapin 400*6*4,220,450,30188,B512,2009-01-02]
22 |[PA60,Pointe Acier 60 (10K),95,134,30178,K111,2008-12-22]
23 |[PA60,Pointe Acier 60 (10K),95,134,30179,C400,2008-12-22]
24 |[PA60,Pointe Acier 60 (10K),95,134,30182,S127,2008-12-23]
25 |[PA60,Pointe Acier 60 (10K),95,134,30184,C400,2008-12-23]
26 |[PA60,Pointe Acier 60 (10K),95,134,30185,F011,2009-01-02]
27 |[PA60,Pointe Acier 60 (10K),95,134,30186,C400,2009-01-02]
28 |[PA60,Pointe Acier 60 (10K),95,134,30188,B512,2009-01-02]
29 |[PS222,PL. Sapin 200*20*2,185,1220,30178,K111,2008-12-22]
30 |[PS222,PL. Sapin 200*20*2,185,1220,30179,C400,2008-12-22]
31 |[PS222,PL. Sapin 200*20*2,185,1220,30182,S127,2008-12-23]
```

```
32 |[PS222,PL. Sapin 200*20*2,185,1220,30184,C400,2008-12-23]
33 |[PS222,PL. Sapin 200*20*2,185,1220,30185,F011,2009-01-02]
34 |[PS222,PL. Sapin 200*20*2,185,1220,30186,C400,2009-01-02]
35 |[PS222,PL. Sapin 200*20*2,185,1220,30188,B512,2009-01-02]
36 |[PA45,POINTE ACIER 45 (20K),105,580,30178,K111,2008-12-22]
37 |[PA45,POINTE ACIER 45 (20K),105,580,30179,C400,2008-12-22]
38 |[PA45,POINTE ACIER 45 (20K),105,580,30182,S127,2008-12-23]
39 |[PA45,POINTE ACIER 45 (20K),105,580,30184,C400,2008-12-23]
40 |[PA45,POINTE ACIER 45 (20K),105,580,30185,F011,2009-01-02]
41 |[PA45,POINTE ACIER 45 (20K),105,580,30186,C400,2009-01-02]
42 |[PA45,POINTE ACIER 45 (20K),105,580,30188,B512,2009-01-02]
43 |[PH222,PL. HETRE 200x20x2,185,1220,30178,K111,2008-12-22]
44 |[PH222,PL. HETRE 200x20x2,185,1220,30179,C400,2008-12-22]
45 |[PH222,PL. HETRE 200x20x2,185,1220,30182,S127,2008-12-23]
46 |[PH222,PL. HETRE 200x20x2,185,1220,30184,C400,2008-12-23]
47 |[PH222,PL. HETRE 200x20x2,185,1220,30185,F011,2009-01-02]
48 |[PH222,PL. HETRE 200x20x2,185,1220,30186,C400,2009-01-02]
49 |[PH222,PL. HETRE 200x20x2,185,1220,30188,B512,2009-01-02]
50 true
```

viii. Select all rows from an inner join between two tables.

```
inner_join(produit,detail,npro,npro_de).
```

```
   |npro|libelle|prix|qstock|ncom_de|npro_de|qcom|
1  |[CS262,Chev. Sapin 200*6*2,75,45,30179,CS262,60]
2  |[CS464,Chev. Sapin 400*6*4,220,450,30178,CS464,25]
3  |[CS464,Chev. Sapin 400*6*4,220,450,30184,CS464,120]
4  |[CS464,Chev. Sapin 400*6*4,220,450,30185,CS464,260]
5  |[CS464,Chev. Sapin 400*6*4,220,450,30188,CS464,180]
6  |[PA60,Pointe Acier 60 (10K),95,134,30179,PA60,20]
7  |[PA60,Pointe Acier 60 (10K),95,134,30182,PA60,30]
8  |[PA60,Pointe Acier 60 (10K),95,134,30185,PA60,15]
9  |[PA60,Pointe Acier 60 (10K),95,134,30188,PA60,70]
10 |[PS222,PL. Sapin 200*20*2,185,1220,30185,PS222,600]
11 |[PA45,POINTE ACIER 45 (20K),105,580,30186,PA45,3]
12 |[PA45,POINTE ACIER 45 (20K),105,580,30184,PA45,20]
13 |[PA45,POINTE ACIER 45 (20K),105,580,30188,PA45,22]
14 |[PH222,PL. HETRE 200x20x2,185,1220,30188,PH222,92]
true.
```

ix. Delete rows that matches a predicate.

```
delete_where(produit,qstock,>,1000).
```

x. Update rows that match a predicate.

```
update_where(produit,[prix,libelle],[300,"sapin très cher"],prix,>=,100).
```

xi. Drop a table.

```
drop_table(person).
```

Additional features

Namespaces

Different tables can have the same column names, consider the following code:

```
?- select(produit,[npro]).
   |npro|
1  |[CS262]
    .
    .
    .

?- create_table(npro_t,[npro]).
true.

?- insert(npro_t,[cS262]).
true.

?- select(npro_t,*).
   |npro|
1  |[cS262]
true.

?- select(produit,[npro]).
   |npro|
1  |[CS262]
    .
    .
    .
```

Select where not

`select_where_not` simply returns the inverse of the select_where command.

```
?- select_where_not(produit,*,qstock,<,1000).
   |npro|libelle|prix|qstock|
1  |[CS264,Chev. Sapin 200*6*4,120,2690]
2  |[PS222,PL. Sapin 200*20*2,185,1220]
3  |[PH222,PL. HETRE 200x20x2,185,1220]
true.
```

Order by

`order_by_desc` will show table ordered by the column in argument in descending order.

```
?- order_by_desc(produit,qstock).
   |npro|libelle|prix|qstock|
1  |[CS264,Chev. Sapin 200*6*4,120,2690]
2  |[PS222,PL. Sapin 200*20*2,185,1220]
3  |[PH222,PL. HETRE 200x20x2,185,1220]
4  |[PA45,POINTE ACIER 45 (20K),105,580]
5  |[CS464,Chev. Sapin 400*6*4,220,450]
6  |[PA60,Pointe Acier 60 (10K),95,134]
7  |[CS262,Chev. Sapin 200*6*2,75,45]
true.
```

`order_by_asc` will do the same but in ascending order.

```
?- order_by_asc(produit,qstock).
   |npro|libelle|prix|qstock|
1  |[CS262,Chev. Sapin 200*6*2,75,45]
2  |[PA60,Pointe Acier 60 (10K),95,134]
3  |[CS464,Chev. Sapin 400*6*4,220,450]
4  |[PA45,POINTE ACIER 45 (20K),105,580]
5  |[PS222,PL. Sapin 200*20*2,185,1220]
6  |[PH222,PL. HETRE 200x20x2,185,1220]
7  |[CS264,Chev. Sapin 200*6*4,120,2690]
true.
```

Aggregation functions

`max`, `min`, `sum` and `avg` are implemented.

```
?- select_max(produit,prix).
max(prix) = 220
true.

?- select_min(produit,prix).
min(prix) = 75
true.

?- select_sum(produit,prix).
sum(prix) = 985
true.

?- select_avg(produit,prix).
sum(prix) = 140.71428571428572
true.
```