
WATERSPORTS

CS27020 Assignment by Petter Vang Brakalsvalet

NOVEMBER 7, 2018

Contents

Database example	2
Dependencies	2
Normalisation	2
UNF -> 1NF	3
1NF -> 2NF	3
2NF -> 3NF	3
ER diagram.....	3
PostgreSQL	4
Creating tables.....	4
Adding all tuples	4
Tuples for Tutors	4
Tuples for meetings.....	4
Tuples for equipment	5
Tuples for used equipment	5
Tuples for the main table Watersports	5
Queries	6
Assignment part 6.1	6
Assignment part 6.2	6
Assignment part 6.3	7
Evaluation.....	8
Planning.....	8
SQL.....	8

Database example

	Day	Activity	Location	Equipment	Purchase Date
Data Type	varchar()	varchar()	varchar()	varchar()	varchar()
Example Data	Monday	Intermediate SUP	Harbour	SUP1, SUP5, Lifejac	21/02/2016
Example Data	Wednesday	Surf for beginners	Borth beach	Surf1, Surf2, Surf3	01/10/2018

Next Inspection	TutorID	TutorName	PhoneNumber	Email
varchar()	varchar()	varchar()	varchar()	varchar()
21/11/2018	jef1	Jeff		jeff@ilovekayak.com
01/10/2019	jac1	Jackie	01970666543	

These are all the different table headers I'm using in my database. As you can see, I'm only using varchar in different lengths. The reason I used varchar is because I couldn't find a better way to do this. I wanted to use the data type data, but I could not get it to work in the time we had so I stuck with what I know how to use. As you can see in the table above, I had to use varchar for the PhoneNumber as well. This is because if I tried to use an integer, it would remove the 0 at the beginning of the number and that would make the phone number invalid.

Dependencies

Watersport is dependent on TutorID from table Tutor, and Day and Activity from table Meeting.

Used equipment table is dependent on Day and Activity from table Meeting, and Equipment from table Equipment.

Normalisation

UNF	1NF	2NF	3NF	Entity Name
✓Day	✓Day	✓TutorID	(FK)Day	Watersports
✓Activity	Activity	Tutor Name	(FK)Activity	
Location	Location	Tutor Number	(FK)TutorID	
Equipment	✓TutorID	Tutor Contact	✓TutorID	Tutors
Purchase Date	Tutor Name	✓Day	TutorName	
Next Inspection	Tutor Number	Activity	PhoneNumber	
TutorID	Tutor Contact	Location	Email	
Tutor Name	✓Equipment	✓Equipment	✓Day	Meetings
Tutor Number	Purchase Date	Purchase Date	✓Activity	
Tutor Contact	Next Inspection	Next Inspection	Location	
			✓Equipment	Equipment
			Purchase Date	
			Next Inspection	
			(FK)Day	Used equipment
			(FK)Activity	
			(FK)Equipment	

UNF -> 1NF

First, I separated all the data, so I would have no repeating data in my tuples so that every tuple has atomic data in it. I did that by making a new table for the repeating equipment and keeping the rest in one table for now.

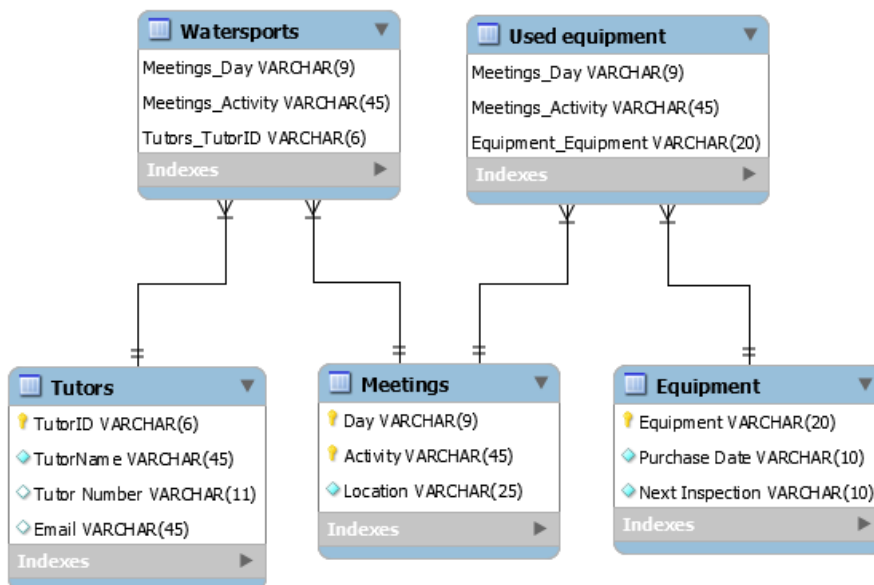
1NF -> 2NF

In the second normal form, I separated the table so that the attributes in the table are dependent on the domain but are also dependent on other tables.

2NF -> 3NF

In the third normal form, I separated the tables so that none of the tables are dependent on any other tables. If one table needs some of the tuples from another table, I can take that tuple from the first table and add it to the second table as a foreign key; so that the tables aren't dependent on the other tables, but the attributes from the tables.

ER diagram



PostgreSQL

Creating tables

This is the part of the SQL code where I start to create all the different tables and the attributes the tables contain.

```
create table Tutors (TutorID varchar(6) not null,
    TutorName varchar(45) not null, phoneNumber varchar(11),
    Email varchar(45), primary key (TutorID)
);
create table Meetings(
    Day varchar(9) not null, Activity varchar(45) not null,
    Location varchar(25) not null, primary key (Day, Activity)
);
create table Equipment(
    Equipment varchar(20) not null, "purchase Date" varchar(10) not null,
    "Next inspection" varchar(10) not null, primary key (Equipment)
);
create table UsedEquipment(
    Day varchar(9) not null, Activity varchar(45) not null,
    Equipment varchar(20) not null, primary key (Day, Equipment),
    foreign key (Day, Activity) references Meetings,
    foreign key (Equipment) references Equipment
);
create table Watersports(
    Day varchar(9) not null, Activity varchar(45) not null,
    TutorID varchar(6) not null, primary key (TutorID, Day),
    foreign key (Day, Activity) references Meetings,
    foreign key (TutorID) references Tutors
);
```

Adding all tuples

This is where I'm adding all the tuples to the different tables.

Tuples for Tutors

```
insert into Tutors values ('daf1', 'Dafydd', '07845333444');
insert into Tutors values ('mar1', 'Mari', '01970666543');
insert into Tutors (TutorID, TutorName, Email) values ('jef1', 'Jeff',
'Jeff@ilovekayk.com');
insert into Tutors values ('ahm1', 'Ahmed', '07845333444');
insert into Tutors values ('jac1', 'Jackie', '01970666543');
```

Tuples for meetings

```
insert into Meetings values ('Monday', 'Beginners kayaking', 'Swimming pool');
insert into Meetings values ('Monday', 'Beginners SUP', 'Harbour');
insert into Meetings values ('Monday', 'Intermediate SUP', 'Harbour');
insert into Meetings values ('Tuesday', 'Open water swimming', 'South beach');
insert into Meetings values ('Tuesday', 'Intermediate kayaking', 'North beach');
insert into Meetings values ('Wednesday', 'Surf for beginners', 'Borth beach');
insert into Meetings values ('Thursday', 'Beginners kayaking', 'Swimming pool');
```

Tuples for equipment

```
insert into Equipment values ('Kayak1', '21/02/2016', '21/11/2018');
insert into Equipment values ('Kayak2', '21/02/2016', '21/11/2018');
insert into Equipment values ('Kayak3', '21/02/2016', '21/11/2018');
insert into Equipment values ('Kayak4', '21/02/2016', '21/11/2018');
insert into Equipment values ('SUP1', '01/10/2018', '30/10/2018');
insert into Equipment values ('SUP2', '01/10/2018', '30/10/2018');
insert into Equipment values ('SUP3', '01/10/2018', '30/10/2018');
insert into Equipment values ('SUP4', '01/10/2018', '30/10/2018');
insert into Equipment values ('SUP5', '01/10/2018', '30/10/2018');
insert into Equipment values ('Surf1', '30/03/2017', '30/12/2018');
insert into Equipment values ('Surf2', '30/03/2017', '30/12/2018');
insert into Equipment values ('Surf3', '30/03/2017', '30/12/2018');
insert into Equipment values ('Lifejacket1', '30/06/2018', '30/10/2018');
insert into Equipment values ('Lifejacket2', '30/06/2018', '30/10/2018');
insert into Equipment values ('Lifejacket3', '30/06/2018', '30/10/2018');
insert into Equipment values ('Lifejacket4', '30/06/2018', '30/10/2018');
insert into Equipment values ('Lifejacket5', '30/06/2018', '30/10/2018');
```

Tuples for used equipment

```
insert into UsedEquipment values ('Monday', 'Beginners kayaking', 'Kayak1');
insert into UsedEquipment values ('Monday', 'Beginners kayaking', 'Kayak2');
insert into UsedEquipment values ('Monday', 'Beginners kayaking', 'Kayak3');
insert into UsedEquipment values ('Monday', 'Beginners SUP', 'SUP2');
insert into UsedEquipment values ('Monday', 'Beginners SUP', 'SUP3');
insert into UsedEquipment values ('Monday', 'Beginners SUP', 'SUP4');
insert into UsedEquipment values ('Monday', 'Beginners SUP', 'Lifejacket2');
insert into UsedEquipment values ('Monday', 'Beginners SUP', 'Lifejacket3');
insert into UsedEquipment values ('Monday', 'Beginners SUP', 'Lifejacket5');
insert into UsedEquipment values ('Monday', 'Intermediate SUP', 'SUP1');
insert into UsedEquipment values ('Monday', 'Intermediate SUP', 'SUP5');
insert into UsedEquipment values ('Monday', 'Intermediate SUP', 'Lifejacket1');
insert into UsedEquipment values ('Monday', 'Intermediate SUP', 'Lifejacket4');
insert into UsedEquipment values ('Tuesday', 'Intermediate kayaking', 'Kayak2');
insert into UsedEquipment values ('Tuesday', 'Intermediate kayaking', 'Kayak3');
insert into UsedEquipment values ('Tuesday', 'Intermediate kayaking', 'Kayak4');
insert into UsedEquipment values ('Tuesday', 'Intermediate kayaking', 'Lifejacket2');
insert into UsedEquipment values ('Tuesday', 'Intermediate kayaking', 'Lifejacket3');
insert into UsedEquipment values ('Tuesday', 'Intermediate kayaking', 'Lifejacket5');
insert into UsedEquipment values ('Wednesday', 'Surf for beginners', 'Surf1');
insert into UsedEquipment values ('Wednesday', 'Surf for beginners', 'Surf2');
insert into UsedEquipment values ('Wednesday', 'Surf for beginners', 'Surf3');
insert into UsedEquipment values ('Thursday', 'Beginners kayaking', 'Kayak1');
insert into UsedEquipment values ('Thursday', 'Beginners kayaking', 'Kayak2');
insert into UsedEquipment values ('Thursday', 'Beginners kayaking', 'Kayak4');
```

Tuples for the main table Watersports

```
insert into Watersports values ('Monday', 'Beginners kayaking', 'daf1');
insert into Watersports values ('Monday', 'Beginners SUP', 'mar1');
insert into Watersports values ('Monday', 'Intermediate SUP', 'jef1');
insert into Watersports values ('Tuesday', 'Open water swimming', 'ahm1');
insert into Watersports values ('Tuesday', 'Intermediate kayaking', 'daf1');
insert into Watersports values ('Wednesday', 'Surf for beginners', 'jac1');
insert into Watersports values ('Thursday', 'Beginners kayaking', 'ahm1');
```

Queries

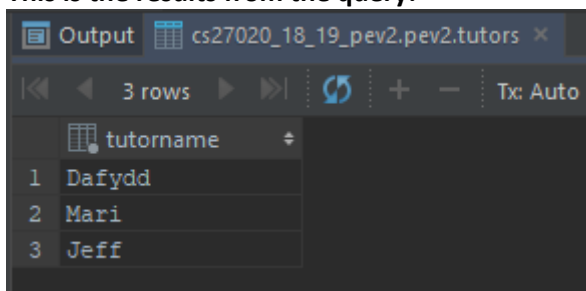
Assignment part 6.1

Question: Find all tutors who teach on a given day (say 'Monday');

To do this I'm saying that I want the TutorName from the table Tutor. The way we are obtaining this is by joining Watersports and the Tutor tables together and saying that the TutorID in both of these tables is the same. However, I only want the name of the tutors who are tutoring on a Monday.

```
select TutorName from Tutors join Watersports
  ON Tutors.TutorID = Watersports.TutorID WHERE Watersports.Day = 'Monday';
```

This is the results from the query:



	tutorname
1	Dafydd
2	Mari
3	Jeff

As you can see Dafydd, Mari and Jeff is tutoring an activity on Monday.

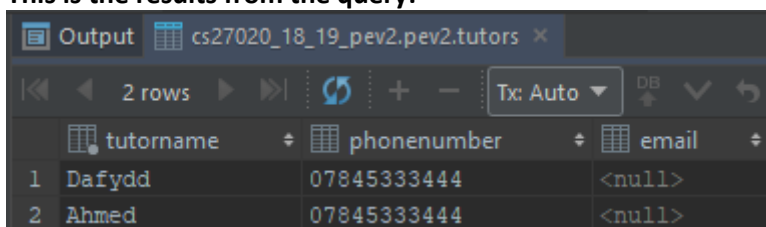
Assignment part 6.2

Question: Kayak4 is missing – find the contact details of all tutors who have led classes involving this boat.

To do this I'm saying that I want the TutorName, phoneNumber and Email from the Tutor table by joining it to the Watersport table and telling it that TutorID is the same in both tables. However, I also need to join UsedEquipment to Watersport and clarify that Activity is the same in both tables. Then, I'm saying that I need to select the people that have used the Equipment 'Kayak4'. However, I'm also saying that the day in UsedEquipment and the day in Watersports have to be the same.

```
select TutorName, phoneNumber, Email from Tutors join Watersports
  on Tutors.TutorID = Watersports.TutorID join UsedEquipment
  on Watersports.Activity = UsedEquipment.Activity where
  Equipment = 'Kayak4' and UsedEquipment.Day = Watersports.Day;
```

This is the results from the query:



	tutorname	phonenummer	email
1	Dafydd	07845333444	<null>
2	Ahmed	07845333444	<null>

As you can see Dafydd and Ahmed has used Kayak4 during this week.

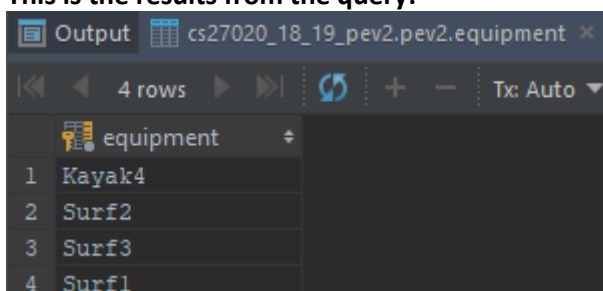
Assignment part 6.3

Question: Find all equipment which is not booked on a particular day, again, say 'Monday' (and can therefore be hired).

To do this, I'm saying that I want equipment from the Equipment table. I'm doing that by joining it to the UsedEquipment table and saying that the Equipment in both of them is the same; also, the second select statement must be 'notted' in order to return all the equipment that ISN'T used on a Monday.

```
select distinct Equipment.Equipment from Equipment join UsedEquipment
on Equipment.Equipment = UsedEquipment.Equipment where UsedEquipment.Equipment NOT IN
(select UsedEquipment.Equipment from UsedEquipment where Day = 'Monday');
```

This is the results from the query:



The screenshot shows a database query output window titled 'Output' with a file path 'cs27020_18_19_pev2.pev2.equipment'. It displays 4 rows of results. The first column is labeled 'equipment' and contains the following values: Kayak4, Surf2, Surf3, and Surf1. The second column is empty.

	equipment	
1	Kayak4	
2	Surf2	
3	Surf3	
4	Surf1	

As you see Kayak4, Surf1, Surf2, Surf3 is the unused equipment on Monday.

Evaluation

I believe that I deserve 70-80% on this assignment. I feel this because I think my solution is a good and efficient solution but there is probably a better way to do it, and I know that my write-up isn't the best and most detailed, but I think it explains what I did.

Planning

I found the first parts of the assignment quite easy once I understood what the question was. However, I had many problems with the normalisation of the tables, because I feel like the test we have to do for the normalisations are vague, and it's very much up to interpretation by the user.

SQL

I found that the SQL bit of this was quite simple, but it was a lot of repetitive commands that I had to use to add all the tuples for the tables. I also wrote the SQL in JetBrains IntelliJ because I like the IntelliJ layout and the ease of use. IntelliJ also keeps all the commands I have used for each table, and I can then easily see what I have done if I have a problem.