

## Conceptual Model (EER Diagram) to Relational Schema and Some Basic SQL

### Goal:

The goal for these laboratory activities is for you to practice the technique of going from an (enhanced) entity-relationship (EER) diagram to a relational schema ready for implementation.

In the second part you will be practising your Basic SQL skills in the form of creating a Database, Inserting some data and running creating some queries.

### Task 1:

Take the 3 EER diagrams you created last week and model these to a Relational Schema. The 3 EER diagrams were:

1. A Taxi Company
2. The Tenning Club ("Kilburn Tennis")
3. Stockport Hats

Even if you do not think your EER diagram is complete I still suggest mapping it across as it is good practice to go through each step using your own design.

You are also welcome to use the Sample Model's (below) I have created from last weeks tasks.

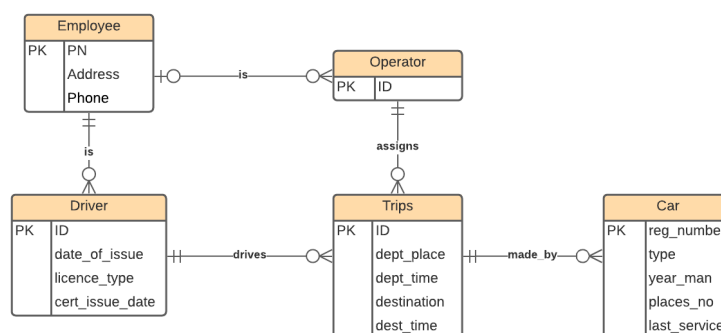


Figure 1: A Taxi Company

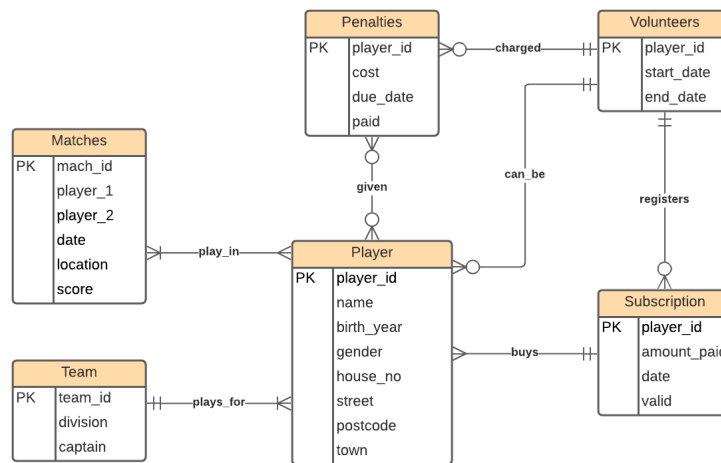


Figure 2: Kilburn Tennis Club

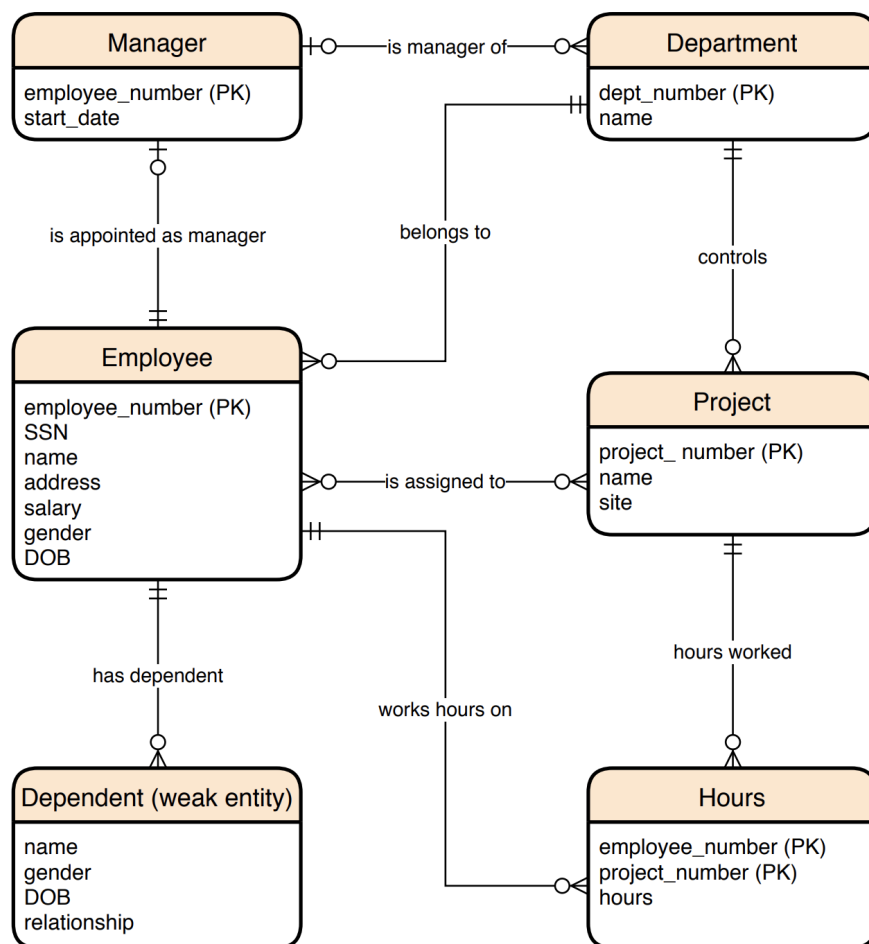


Figure 3: Stockport Hats

## Task 2:

### Setting up your Database:

1. Go to [web.cs.manchester.ac.uk](http://web.cs.manchester.ac.uk) and log in with your University username (i.e. a12345bc) and password.
2. You should see the same page as Figure 4. Only the lower box “MySQL Database Account” is relevant. Choose a password, but this should **NOT** be your university password. Choose something different and remember it: you will be needing it again. Click “Create database account”
3. This should then take you to a page like Figure 5. Again, only the “MySQL Database Account” is relevant. For \$database\_host and \$database\_name, your username will appear between the quotes.
4. Make a note of your credentials and then click on the like: [dbhost.cs.man.ac.uk/phpMyAdmin](http://dbhost.cs.man.ac.uk/phpMyAdmin) and log in with your university username and the new password you have just created.
5. Here you can use the SQL Tab to paste in SQL Queries for creating and manipulating your databases.

The screenshot shows a web interface with two main sections. The top section, titled 'Web Publishing', contains instructions about publishing a website from a Git project. It has input fields for 'Git project' (containing a URL) and 'Subdirectory' (containing 'project\_dir'), and a 'Publish project' button. The bottom section, titled 'MySQL Database account', contains instructions about creating a database account. It has input fields for 'Database password' and 'Confirm password', and a 'Create database account' button. Both sections have a small information icon (i) in the top right corner. At the bottom of the page, there is a link to 'open a ticket'.

**Web Publishing**

You do not currently have a published website. Please enter the URL of a git project to publish as your website in the box below. Note that the project you select *must be public*: you can not publish private projects.

The publish mechanism will generate a `config.inc.php` file in the root of your web directory containing your database credentials if you have any. Please see the documentation for more information.

**Git project**

**Subdirectory**

**Publish project**

**MySQL Database account**

You have not yet created a MySQL database account. In order to create your account, please enter a password to use in the form below. **Do not** use your University central account password, or a password you use for any important systems as your database password. Your database credentials will be written to a `config.inc.php` file added to the root of your web directory.

In order to reduce the likelihood of various common problems with database passwords, the system will only allow you to specify alphanumeric (`a-z, A-Z, 0-9`) characters and `_, -`, and `+` in your database password. Passwords should be 8 characters or more in length.

**Database password**

**Confirm password**

**Create database account**

If you encounter any problems using this system or your website or database, please [open a ticket](#).

Figure 4: MySQL Database Account - Log In

### Web Publishing

You do not currently have a published website. Please enter the URL of a git project to publish as your website in the box below. Note that the project you select *must be public*: you can not publish private projects.

The publish mechanism will generate a `config.inc.php` file in the root of your web directory containing your database credentials if you have any. Please see the documentation for more information.

**Git project**

**Subdirectory**

[Publish project](#)

### MySQL Database account

Your database account and personal database connection information is shown below. Note that, with the exception of the host name, the information is case sensitive:

```
<?php
$database_host = "dbhost.cs.man.ac.uk";
$database_user = " ";
$database_pass = "<the password you set>";
$database_name = " ";
?>
```

This information has been written to `config.inc.php` files in each of your web directories, if you have published any. You can manage the data in your database through the web using the phpMyAdmin system here:

<https://dbhost.cs.man.ac.uk/phpMyAdmin/>

[Change database password](#) [Delete database account](#)

If you encounter any problems using this system or your website or database, please [open a ticket](#).

Figure 5: MySQL Database Account Credentials

### Task 3:

#### Creating and Querying a Database:

We will now look at creating a Mario Kart Database for use in this task:

1. Create your database based on the database schema below where the Orange cells give the attribute data type and additional constraints are provided in the green cells. You will need to create the tables first and then insert the relevant data.

##### Player

<u>ID</u>	Name	Age	ChosenChar	Vehicle	Track
1	Alice	21	Mario	MiniBeast	MushroomCup
2	Bob	15	Luigi	JetSetter	MushroomCup
3	Carol	61	KingKoopas	Sprinter	FlowerCup
4	Daniel	83	Peach	MiniBeast	FlowerCup

INT	VARCHAR(30)	INT	VARCHAR(30)	VARCHAR(30)	VARCHAR(30)
PK	not NULL		FK to GameCharacter(Name)	FK to Vehicle(Name)	FK to Track(Name)

##### GameCharacters

<u>Name</u>	Gender	IsHuman
Mario	M	TRUE
Luigi	M	TRUE
Peach	F	TRUE
KingKoopas	M	FALSE

VARCHAR(30)	CHAR	BOOLEAN
PK	not NULL	Default: TRUE

##### Vehicle

<u>Name</u>	Speed	Weight	Acceleration
MiniBeast	55	32	29
JetSetter	69	56	21
Sprinter	64	48	27

VARCHAR(30)	INT	INT	INT
PK	not NULL	not NULL	not NULL

### Track

<u>Name</u>	Difficulty	IsFinalStage
MushroomCup	5	FALSE
FlowerCup	7	FALSE
SpecialCup	10	TRUE

VARCHAR(30)	INT	BOOLEAN
PK	not NULL	not NULL, Default: TRUE

### Result

<u>Player1</u>	<u>Player2</u>	Player1Wins
1	2	TRUE
3	4	FALSE

INT	INT	BOOLEAN
PK, FK to Player(ID)	PK, FK to Player(ID)	

2. You now need to write a query to produce specific outcomes:

- Find the name(s) and Track(s) of the player(s) who choose MiniBeast as their vehicles and are younger than 60 years old.
- Find the ID, Name, Chosen Character and if the track is a final stage, of those player(s) who are older than 20 and run on a track having a difficulty less than 8.
- Find the name(s) and difficulty level(s) of the track(s) that are used by player(s) who won the game and who chose Male characters, and show the names of the male characters.

There is usually more than 1 way to structure your queries so I suggest you experiment with creating different MySQL queries based on the Mario Database to see how you can achieve the same results but with different queries.

## Task 4:

### Using a Larger Database:

Northwind is a sample database distributed with MS Access. It comprises 150K of data. So not particularly large, but not tiny either.

1. Download the following two files to your laptop:

- *northwind-schema.sql*
- *northwind-data.sql*

In phpMyAdmin, select your username in the lefthand menubar (immediately under “information\_schema”). Then go to the Import tab, and under “File to import”, select “Browse your computer” and choose your downloaded northwind-schema.sql. Then click “Go”. After a short while a lot of MySQL confirmation messages come up, which you can ignore.

Now, select your username in the lefthand menubar again, and you should see all the Northwind tables have been created according to the schema in the file you submitted.

Open *northwind-schema.sql* in a text editor on your laptop and browse its contents.

Now, in phpMyAdmin, click on the names of some of the tables to see their structure, and see how that structure relates to the SQL commands that create it.

The tables are empty of data so let's put some in.

Making sure your username in the lefthand menubar is selected, again go to the Import tab and this time select and upload northwind-data.sql. And “Go”.

Repeat the above to examine the data in the tables, and again look at the SQL code in the data file.

Now use the Designer tool to try and make some sense of the structure. Don't worry if you can't. But this should show you the difficulty of visualising non-trivial databases.

2. You now need to write a query to produce specific outcomes, the expected results can be seen below in Figure 6:
  - (a) Write a query on the “customers” table to retrieve the names of all customers who are company owners in Seattle.
  - (b) Write a query on the “purchase\_orders” and “purchase\_order\_status” tables to retrieve the ids of all purchase orders which have not been approved.

- (c) Write a query on the “purchase\_orders”, “purchase\_order\_details” and “purchase\_order\_status” tables to retrieve the ids of all purchase orders which have been approved and which have a quantity of between 80 and 120, and order the result ascending by quantity.

last_name	first_name	job_title	city
Bedecs	Anna	Owner	Seattle
Bagel	Jean Philippe	Owner	Seattle

id	status
146	Submitted
147	Submitted
148	Submitted

id	status	quantity
91	Approved	80.0000
93	Approved	80.0000
105	Approved	100.0000
93	Approved	100.0000
100	Approved	100.0000
92	Approved	100.0000
92	Approved	100.0000
96	Approved	100.0000
90	Approved	100.0000
91	Approved	100.0000
92	Approved	120.0000
93	Approved	120.0000

Figure 6: Expected Outputs, a on the left, b in the middle, c on the right

## End of Laboratory Exercises