Databases - Practical exercises

Monday - before break

a) Data modelling #1

b) Using phpMyAdmin to begin a database

Aim: create a data model of a proposed small scale database with a handful of entities using the ER diagramming convention; and begin your database

Consult the tutor’s Cobbett data model for diagramming conventions.

If using your own data, start a diagram from scratch

If using tutor’s data on Cobbett’s Rides, replicate the supplied diagram (just the ‘extract’ version for the time being) in your drawing tool.

Consult the text at <http://www.gutenberg.org/files/34238/34238-h/34238-h.htm> for more data.

Beginning to model the agile way:

* first cut entities: things in the world
* first cut attributes: their attributes in the world
* with list of attributes in hand, consider each - you only want to keep those which are dependent (ie. whose value is dictated by) the primary key - remove the others into a new entity
* attributes which have more than one value for given entity (repeating group) also get moved out into new entity

Begin your database, guided by your model

* if using the tutor’s Cobbett model, create two tables
* data model (ER form) - focus on a fragment: ‘ride’ and ‘stop’ and their rel’ship
* first cut: make ‘ride’ a table and ‘stop’ a table
* import data from supplied spreadsheet (or type it in yourself if that takes your fancy – from text at link above – some people find that typing is an exercise which helps them think)

Cobbett\_DHcopy\_base\_STOP.csv, Cobbett\_DHcopy\_base\_RIDE.csv

* query the db to find the Godalming stop
* add another field to ‘ride’ for practice

[ALTER](http://localhost/phpMyAdmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/alter-table.html" \t "mysql_doc) [TABLE](http://localhost/phpMyAdmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/alter-table.html" \t "mysql_doc) `ride` ADD `chapter\_name` [TEXT](http://localhost/phpMyAdmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/string-types.html" \t "mysql_doc) [NOT](http://localhost/phpMyAdmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html" \l "operator_not" \t "mysql_doc) NULL AFTER `ride\_name`;

* if using your own model, choose the two entites which seem most important or central to the domain and create two tables

Monday – after break

SQL practice - repeat in raw SQL what you did in phpMyAdmin (PMA)

Preliminary – switch to command line mysql - see separate instruction sheet

Then, check you have access to your database:

mysql>show databases;

You should see more than the ‘information\_schema’ database (which is always there and belongs to mySQL itself).

The tutor’s database is called ‘cobbett’. If using this, type:

mysql>use cobbett;

Remind yourself of what tables it contains:

mysql>show tables;

Choose one if its tables and remind yourself of what fields are defined in it:

mysql>select \* from tablename;

\* means ‘all the fields’

select means ‘show the data’

Some things to do:

ALTER TABLE `ride` ADD `gobbet\_of\_Cobbett` TEXT NULL AFTER `ride\_name`;

And, as did not add a primary key when first defined table:

ALTER TABLE `stop` ADD INDEX(`stop\_placename`);

--> BUT actually, ‘stop\_placename’ is not unique within whole (projected) database - perhaps not even within one ride --> what to do now?

(Hint: add a field to ‘stop’)

After all this, we are now in a position to get back to our query – finding whatever the database holds for the Godalming stop. Use the SQL select statement.

Tuesday - before break

SQL practice continued – querying across tables

Querying across tables - select with a join:

but first try:

select \* from ‘stop’, ‘ride’;

BUT, this appears to give a quite useful result only because of the special initial data we have - there is only one record in the Ride table

Tuesday - after break

SQL practice continued – wholesale changes

Some statements to practice (simplified syntax):

create table tablename;

drop table tablename; --> deletes all the data and the table definition

truncate table tablename; --> deletes all the data but table still defined in database

drop database databasename; --> deletes whole database with all its data

Both these affect your database structure.

**Warmup:**

drop table --> deletes the table definition, and therefore **all its data** --> in the real world --> be careful! Here in class - go for it!

If you really want to save the data you have already entered in the table, first copy the table:

create table ride\_copy select \* from ride;

Double check that it was indeed created:

show tables;

And double check that the clone of your table really has all the data:

select \* from ride\_copy;

And double check that the clone really has the right table definition:

show full fields from ride\_copy;

**Now play with your database: break, remake, enter data**