**MCC Tutorial & Exercises**

*Important Note:* The purpose of this tutorial is to show you how to use SQL to satisfy different sorts of queries on relational databases. It is not to teach you all the different features and operators of the SQL language and it does not cover all the aspects of SQL.

The Database Schema

The following relations show data the Morpeth Cricket Club keeps of MCC club cricketers and their batting performances in matches against other clubs.

mccPlayer(registration\_number {pk}, name, age, p\_street, p\_town, p\_postcode)

mccMatches(Match\_date {pk}, opposing\_team, ground\_name)

mccGround(ground\_name {pk}, g\_street, g\_town, g\_postcode)

mccMatch\_performance(registration\_number {pk}, match\_date {pk}, batting\_score)

Primary keys are shown {pk}.

**Notes**. Dates are represented as numbers in the form yymmdd, e.g. the 26th May 2013 is 130526. Batting\_score, and age, are also numeric but all other attributes are character strings.

**One Table Queries**

List the names of all the players who live in the town 'Morpeth'.

The personal details of the players are all in the Player relation so we don't have to join different relations together. We don't want all the players, only those who live in Morpeth, so we have to work out a condition that will identify the players we do want. All conditions are expressed in terms of attributes and values and so the condition is p\_town = 'Morpeth'.

The SQL is:

SELECT name

FROM mccPlayer

WHERE p\_town = **'**Morpeth**'**;

It is not necessary to put the reserved words in capitals. I am doing so for clarity.

The FROM clause specifies the table(s) you are going to use and the WHERE clause specifies the conditions. The operation specified in the SELECT clause is always carried out last (but before ORDER BY clause, if any) so you don’t worry about losing columns.

Now you try some:

1. List the names of all grounds in London sorted in reverse-lexicographical order.
2. List all the details of the player whose name is 'Shane Watson'.
3. List the dates of any matches played against the opposing team 'Darlington'.

When specifying conditions you can use any of the comparators; =, >, <, =>, <=. You can also negate them, e.g. not =, not > etc. You can also use combine comparisons using logical ands ( AND ) and ors ( OR ).

For example:

* List the names of players who live in the towns 'Morpeth' or 'Alnwick' in lexicographical order.

SELECT name

FROM mccPlayer

WHERE p\_town = 'Morpeth' OR p\_town = 'Alnwick';

Note that Kumar comes after Michael. To make the proper order we need to add ORDER BY.

SELECT name

FROM mccPlayer

WHERE p\_town = 'Morpeth' OR p\_town = 'Alnwick'

ORDER BY name ASC;

As both conditions are applied to the same attribute you could specify the test values as a set, as in;

SELECT name

FROM mccPlayer

WHERE p\_town IN ('Morpeth', 'Alnwick');

* List the names of players who are more than 30 years old and live in 'Morpeth'.

SELECT name

FROM mccPlayer

WHERE p\_town = 'Morpeth' AND age > 30;

* List the names of players who are more than 30 years old and live in 'Morpeth' or 'Alnwick'.

SELECT name

FROM mccPlayer

WHERE (p\_town = 'Morpeth' OR p\_town = 'Alnwick') AND age > 30;

Note that I've used brackets to preserve the order of the conditions.

A few more for you to try;

1. List the registration numbers of any players who had a batting score of more than 50 on the 6th September 2013.
2. List the names of any grounds in Durham or Sunderland.
3. List the registration numbers of any players who had a batting score of 100 or more during the month of August in either 2011 or 2012.
4. List the names of opposing teams that MCC have played against at the ground called 'Collingwood Fields'.

A computer screen shot of a black screen

Description automatically generated