Labs start tomorrow, there are two time slots. Check online for which one you're in.

Data Manipulation (cont.)

Deleting Rows

Example:

```
DELETE
FROM students
WHERE id_numbers = '987654321';
```

There is a problem with the following example:

```
DELETE
FROM students
WHERE first_name = 'Gráinne' AND last_name = 'Gogerty';
```

The problem is that there could potentially be many Gráinne Gogertys, rather than just the one. Using the key above ensures that only one student will be removed.

Can also delete multiple rows:

```
DELETE
FROM students
WHERE hometown = 'Tralee';
```

Updating Values Within a Table

Example:

```
UPDATE students
SET points = 500
WHERE id_number = '112356489';
```

Use a WHERE condition to specify the target.

Can update multiple values at once (multiple rows and multiple columns):

```
UPDATE students
SET points = 1.2*points, first_name = 'Loretta'
WHERE hometown = 'Tralee';
```

Data Definition Language

So far we've looked at SQL's *data manipulation language* (DML). Now we're going to look at *data definition language* (DDL) which allows us to set up a table. DDL allows us to:

- specify the structure of the DBs table(s)
- create a table with this structure

A Table Definition for the Running Example

```
CREATE TABLE students
(
id_number CHAR(9),
first_name VARCHAR(20),
last_name VARCHAR(30),
date_of_birth DATE,
hometown VARCHAR(30),
course CHAR(5),
points INTEGER,
…
)
```

This creates a table called students with the specified structure. We enter attribute names (e.g. points) first, followed by *types* (e.g. INTEGER), with lengths in brackets (e.g. (30)).

You can go below the length limit for CHAR, it'll usually pad them with blanks to fit the limit. You *cannot* go above.

The difference between CHAR and VARCHAR:

- CHAR has fixed length, and will pad with blanks to reach the limit
- VARCHAR has an upper limit, and won't pad anything.

You shouldn't rely on CHAR padding things with blanks, because it depends on the implementation, and also, padded blanks can cause problems later anyway.

Generally you use VARCHAR unless you know the exact length.

Lengths are important for managing space. The implementation may try to save space by allocating less memory when VARCHARs aren't filled, for example, if it's worth doing.

Overview of Data Types in SQL

Broken into 4 broad categories:

- Textual: strings of characters
- Numerical: numbers (integers and more)
- Temporal: dates, times, date-time timestamps etc.
- Others: BLOB (binary large object), more

Integer Types

The INTEGER type is limited to the range
-2,147,483,468…2,147,483,467 in MySQL.

Most systems offer variants (SMALLINT, BIGINT) with different ranges;
require different amounts of space; need to choose carefully.

Other Numerical Types

DECIMAL(n, d):

- any n-digit number with d digits after the decimal point (d-digit mantissa)

FLOAT

- scientific notation (e.g. 1.34E+12 = 1.34 x 10^{+12})
- useful for scientific data
- system-dependent limits on number size