CSCU9B3

MySQL 3

Data Types

Data Types in MySQL

- Data types define the way data in a field can be manipulated
- For example, you can multiply two numbers but not two strings
- We have seen data types mentioned in **`CREATE TABLE`** statements
- Now we look at why it is useful to choose the correct type

  http://dev.mysql.com/doc/refman/5.7/en/data-types.html

Common Data Types

- Numeric
- Character String
- Date and Time

Consequences of Data Types

- Storage size
- Range of values specified
- Operations performed on the data

Numeric Types

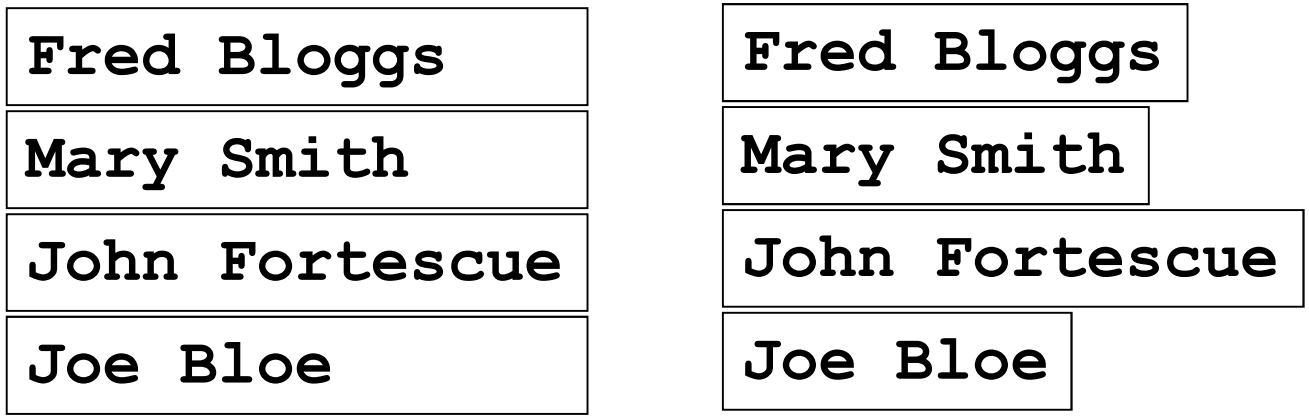
- Integer types of various sizes
  - (pick the right one to minimise storage space)
  - `Tinyint` = 1 byte (-128 to 127 or 0 to 256 UNSIGNED)
  - `Smallint` = 2 bytes
  - `Mediumint` = 3 bytes
  - `Int` = 4 bytes
  - `Bigint` = 8 bytes
  - Maximum number of digits can be specified: eg INT(size)
- Floating Point
  - `Float` = 4 bytes
  - `Double` = 8 bytes
- Fixed Point
  - `Numeric` or `Decimal(size, d)`
  - `d` = number of digits to right of decimal point
- `Bool` is implemented as `Tinyint`
  - Boolean: True = 1 and False = 0

String Types

- **`CHAR(length)`**
  - Stores strings with “length” characters – smaller strings are right padded with blanks
    - Up to a maximum of 255 characters
  - Blanks removed on retrieval
  - Indexed searching is faster (only if no fields are variable length).
- **`VARCHAR(length)`**
  - Stores variable size (up to “length”) strings with no padding ie only actual characters stored
    - Trailing blanks may or may not be removed (version dependent)
  - Stores prefix (1 or 2 bytes) that contains the length of the string
  - Uses less storage if strings really are variable length
  - If they are always the same length, actually uses more storage

String Types (2)

- **`CHAR(length)`** **`VARCHAR(length)`**



TEXT Types

- **`TEXT`** types hold variable length character strings
  - `TINYTEXT` (up to 255 characters)
  - `TEXT` (up to 65,535 characters)
  - `MEDIUMTEXT` (up to 16,777,215 characters)
  - `LONGTEXT` (up to 4,294,967,295 characters)

- **`TEXT`** has no trailing space removal on INSERT or SELECT (unlike CHAR or VARCHAR)
- **`TEXT`** has padding added in comparisons to fit the compared object (like CHAR and VARCHAR)

BLOB Types

- **`BLOB`** is a Binary Large OBject
- **`BLOB`** types hold variable length binary strings:
  - **`Tinyblob`**
  - **`Blob`**
  - **`Mediumblob`**
  - **`LongBlob`**
- Useful for encrypted or compressed data
- **`BLOB`** data is ordered by the binary values
- **`TEXT`** (and other character types) is ordered by the collation of the character set
  - _ci (case insensitive), _cs (case sensitive), or _bin (binary)
  - Case insensitive: A < b and a < B

Enumerated Types

- You can create enumerated types

```
CREATE TABLE sizes ( name ENUM('small', 'medium',
'large'))
```

- Enumerated types have an index that starts at 1
- Index zero is an error value (zero)
  - Entering a value that was not defined, e.g. ‘huge’
- Enumerated values are sorted by their index – which is the order in which they were defined so:
  - small < medium < large

Set Types

- The SET type can have zero or more values from a list of permitted values
- Each value should appear at most once

```
CREATE TABLE mytable (col SET('a', 'b', 'c', 'd'));
```

- Insert values like this

```
INSERT INTO mytable (col) VALUES ('a,d');
```

Set Types (2)

- Retrieval automatically removes duplicates:

```
CREATE TABLE myset (col SET('a', 'b', 'c',
'd'));
```

- If you insert the values 'a,d', 'd,a', 'a,d,a':

```
INSERT INTO myset (col) VALUES ('a,d'),
('d,a'), ('a,d,a');
```

- Then all these values appear as 'a,d' when retrieved:

```
SELECT col FROM myset;
```

```
+-----+
| col |
+-----+
| a,d |
| a,d |
| a,d |
+-----+
```

Date and Time Types

- **`DATETIME`** stores a date and time:
  YYYY-MM-DD HH:MM:SS
- **`DATE`** stores just a date: YYYY-MM-DD
- You can enter values in a reasonable number of formats:
  - yyyy-mm-dd as a string
  - yyyy/mm/dd as a string
  - yyyymmdd as a string or number
  - yymmdd as a string or number

- **`TIME`** as HH:MM:SS
- **`YEAR`** as YY or YYYY

Date and Time Functions

- There are a great many functions in MySQL for dealing with dates and times
- Part extraction:
  - `Hour(), Month()` etc.
- Current time:
  - `Now()`
- Counting
  - `Dayofmonth(), Dayofyear()`
- Adding
  - `Addtime()`
- You can also use standard comparisons:
  - `WHERE date1 > date2`
  - `WHERE date BETWEEN date1 AND date2`

Comparing Types

- MySQL is very tolerant of mixing of types
- Different types can be compared:
  - `SELECT * WHERE intfield = floatfield`
  - `SELECT * WHERE intfield = stringfield`
  - Are allowed
- You can use LIKE for numbers:
  - `SELECT * WHERE intfield LIKE 12%`
  - Finds all rows where intfield starts with 12

Data Type Integrity

- You can specify things about the qualities of data that is to be stored
- We have seen that **decimal** allows you to be precise about **decimal places**
- You can specify whether or not a field can contain **NULL** (ie whether it has to have a value)
- **Enumerating types** or using **sets** helps you control what values go into a field
- **Dates** are automatically checked for correct format
- **Variable length fields** have a maximum size that cannot be exceeded

End of Lecture

- The next SQL lecture will look at using PHP to connect to and query a database
- Before then there will be an introduction to the scripting language, PHP