Cardiff's School of Computer Science & Informatics

CM1102 "Web Applications"

Databases and MySQL

Dr Natasha Edwards

(Special thanks to Dr Kirill Sidorov)

Static vs dynamic web pages

- So far, we have seen how to design *static* web pages using HTML/CSS.
  - The content of the page was fixed, predetermined.
  - Does not depend on the request of user, does not alter the state of the system, always the same.
  - So, mathematically, speaking, static web page is a pure function, which, moreover, is a constant.

Static vs dynamic web pages

- As you might have noticed, many web sites display content, which changes is response to the actions of the user(s), depends on the previous state of the system, as well as allow for side effects (change the state of the system).
  - Facebook, eBay, twitter, Amazon – examples are countless.
- Such web pages are classified as – *dynamic* (non-constant impure functions of user input and previous state).

Example: Facebook

- Suppose I login to Facebook.
- The page I now see (my profile) depends on my request (to display my profile) as well as on the previous state of the system (a record corresponding to my profile information in Facebook's database).
- Next, I upload some photos of me.
- The state of the system is now different (photos have been saved in a database). And the behaviour of the website is now different as well:
  - Some other user then logs in to Facebook, and requests to display my photos.
  - The page which this user now sees is a function of their request and of the state of the system.

Databases

- For the purposes of this module, we are interested in databases primarily as a way to store and manipulate the state of a dynamic web site in a principled and organised manner.
- Basics:
  - What is a database?
  - The relational model
  - SQL
- Databases will be examined in more detail in CM2102 "Database systems" module (run by Dr J. Shao)

Databases

Databases are facilities for storage and management of data in a principled manner:

- Avoid duplication
- Maintain consistency (avoid contradictions)
- Maintain integrity (data makes sense, conforms to user-specified constraints)
- Maintain security:
  - Access can be confined to authorised individuals.
  - System of privileges for operations on the data.
  - Protection against loss and corruption (back-up, roll-back)
- Support concurrency (simultaneous manipulation by multiple users)
- Maintain the independence of data from the mechanism of queries (abstraction)

DBs & Content Management Systems (CMS)

Databases may be used effectively to construct *content management systems* (CMS).

- The entire content of a website may be stored in a database.
- Site content is updated or changed **not by** altering the website's code (e.g. JavaScript, HTML, Python, PHP, etc.) - but by changing entries in a database.
- The database itself may be edited using a web-based interface.
- Examples: Wordpress, Joomla, various wikis...
- This introduces another layer of abstraction between the data and its presentation.

MySQL

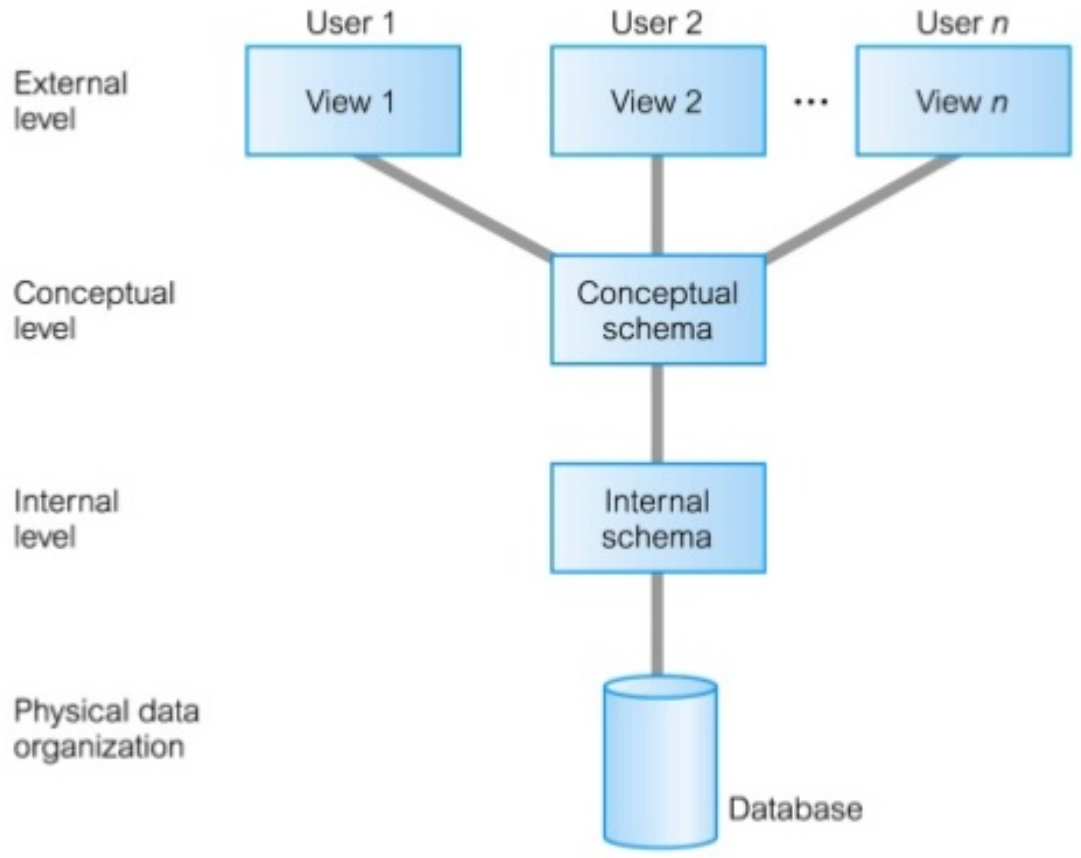
- MySQL is a *database management system* (DBMS) for relational databases, based on the Standard Query Language (SQL).
- MySQL is open source.
- Client-server architecture. Clients *connect* to the database server and submit queries.
- MySQL appears to very widely used: YouTube, Flickr, Facebook, twitter, google and many others use MySQL to manage content.
- *LAMP* is an acronym for "Linux, Apache, *MySQL*, Perl/*PHP*/*Python*".

MySQL

- *phpMyAdmin* provides convenient way to interact with and manage a MySQL database.
  - Provides an administrative interface to MySQL.
  - Itself is written in PHP.
- You have access through: `http://www.cs.cf.ac.uk/phpMyAdmin/`
- Short guide on how to use it: `https://docs.cs.cf.ac.uk/notes/administering-mysql-with-phpmyadmin/`
- Before using phpMyAdmin make sure you have MySQL username, database name and password. You should have had an email with these details. **(Note: these credentials are different from your Uni network username and password.)** Contact our network support if you have problems.
- You can also use a 'stand-alone' software package, e.g. MySQL Workbench (see: `https://docs.cs.cf.ac.uk/notes/accessing-mysql-from-windows/`)

Database abstraction

ANSI/SPARC three levels of abstraction:



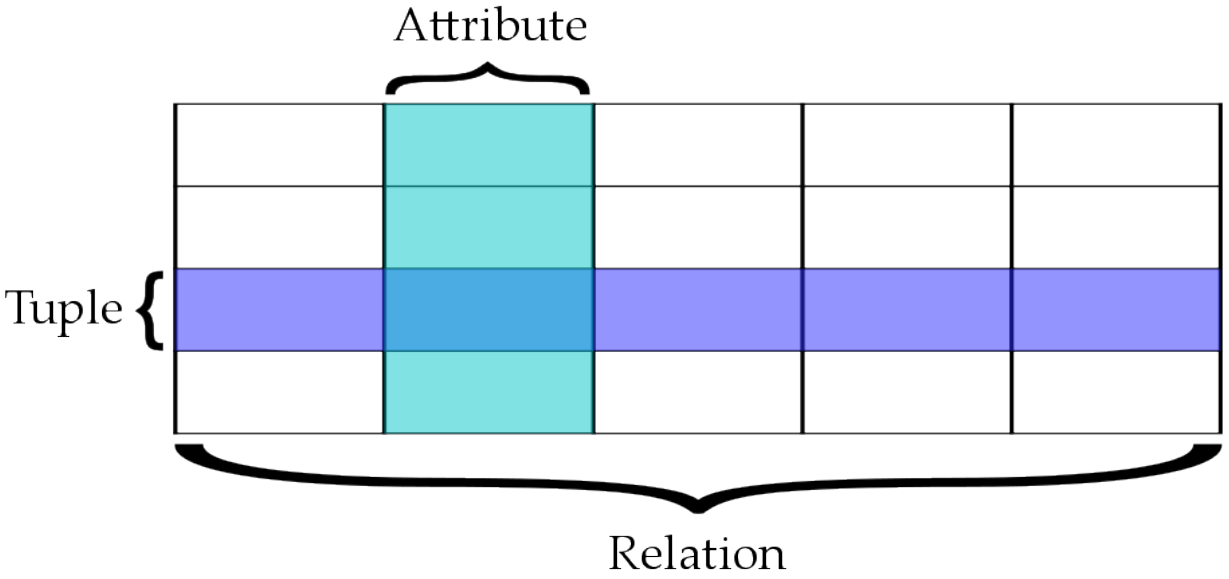
Database abstraction

ANSI/SPARC three levels of abstraction (schemas):

- ***External level*** (or view level): End-user view of the data. The same data can be regarded from several user/application-dependent perspectives according to its different uses.
- ***Conceptual level***: The information components. What data entities are stored, what their attributes, and the relations between data.
- ***Internal level***: Data types, query methods, record structures, indexing, encryption, compression...
    - ***Physical level***: How the data is stored physically, e.g. media.

RDBMS & Relational model

- Relational database is a collection of relations.
- A *relation* is a table of values in rows and columns
  - The *rows*, also called *tuples*, define real world objects (entities) or relationships between real world objects.
  - The *columns* are attributes of the represented objects.



Relationships: types

- One-to-one: Each car has one vehicle registration number. Each registration number corresponds to one car.
- One-to-many: Each employee (`staffNo`) works in one branch. A branch may have more than one employee.
- Many-to-many: Each actor may have starred in several movies. Each movie can have several actors.
  - Which actors played together in the same movie?

Relationships: use

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

We can determine:

- In which branch a particular employee works.
- Which employees work in a particular branch.
- Which branches have employees with a salary $>$ 20,000.
- Which branches do not have a manager.
- etc.

SQL: data types

Example types (there are several others):

- `INT` — integer.
- `FLOAT` — floating point number.
- `VARCHAR(n)` — string of max length `n`.
- `TEXT` — larger pieces of text.
- `DATE` — date in format YYYY-MM-DD.

Can follow `INT` with `AUTO_INCREMENT` in which case the value is set automatically and incremented whenever a new value is encountered (set the attribute to `NULL` when entering data with `INSERT`).

For more info about data types in SQL/MySQL:
`http://dev.mysql.com/doc/refman/5.0/en/data-types.html`

SQL: case sensitivity

- MySQL commands are case-insensitive, but traditionally written in UPPER case:
  - e.g. `CREATE TABLE`, `SELECT`, `INSERT`, etc.
- MySQL 'identifiers' (names of databases, tables, index, columns, stored procedures, etc.) **can be** case sensitive, depending on your MySQL and/or OS configuration
  - e.g. see documentation for identifiers' case sensitivity here: `https://dev.mysql.com/doc/refman/5.7/en/identifier-case-sensitivity.html`

SQL: creating tables

The command `CREATE TABLE` creates a table in a database.
Example:

```
CREATE TABLE Branch (branchNo VARCHAR(10) NOT NULL,
  street VARCHAR(20) NOT NULL,
  city   VARCHAR(20) NOT NULL);
```

- These fields declared as strings using the `VARCHAR` type.
- `VARCHAR(10)` — maximum of 10 characters.
- `NOT NULL` — must have value.

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

SQL: adding rows

To add a row (record) to a table, use the `INSERT` command. Example:

```
INSERT INTO Branch (branchNo, street, city)
  VALUES ("B008", "45 Low St", "Upborough");
```

If all fields of a row are being inserted then we don't need to specify the list of fields, provided they are listed in correct order.

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London
B008	45 Low St	Upborough

SQL: changing values

To change a value of a field use the `UPDATE` command together with a condition (`WHERE`...). Example:

```
UPDATE Branch SET city = "Downborough"
  WHERE Branch.branchNo ="B008";
```

Various operators can be used in the `WHERE` clause:

- Logical operators: `AND`, `OR`, `NOT`
- Equivalence: `==`, `!=`
- Comparision: `>`, `<`, etc.

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London
B008	45 Low St	Downborough

SQL: deleting rows

The DELETE command deletes entire rows. We can use conditions on which rows to delete. Example:

```
DELETE FROM Branch
  WHERE branchNo = "B008";
```

Various operators can be used in the WHERE clause:

- All rows that satisfy the condition are deleted (only one in this example).
- Here the row to delete is the one where the branchNo is B008.

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London
~~B008~~	~~45 Low St~~	~~Downborough~~

SQL: retrieving data

The SELECT command provides many ways to retrieve data. For example, find records from a table where an attribute has a particular value:

```
SELECT branchNo, street, city FROM Branch
  WHERE city="Bristol";
```

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London
B008	45 Low St	Upborough

Result:

branchNo	street	city
B004	32 Manse Rd	Bristol

SQL: retrieving data

The `WHERE` clause of the `SELECT` command can take other conditions. For example, let us find staff with salary more than 20,000. Tip: to retrieve all fields use `*` for the list of fields.

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

```
SELECT * FROM Staff WHERE salary > 20000;
```

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG14	David	Ford	Supervisor	25000	B003
SG5	Susan	Brand	Manager	32000	B003

Retrieving data: linking tables

- To retrieve associated data from multiple tables they need to have at least one attribute (*key*) in common.

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

- Which is a common attribute?

Retrieving data: a join between two tables

To retrieve related data that are in two tables use a common attribute (e.g. `branchNo`) to match records. This is called a *join*. Example:

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

```
SELECT S.firstName, S.lastName
  FROM Branch B, Staff S
  WHERE S.branchNo = B.branchNo
  AND B.city = "London";
```

Result:

firstName	lastName
John	White
Julie	Lee

SQL: a join between two tables

```
SELECT * FROM Branch B, Staff S
  WHERE S.branchNo = B.branchNo;
```

Is equivalent to:

```
SELECT * FROM Staff INNER JOIN Branch
  ON Staff.branchNo = Branch.branchNo;
```

Result:

branchNo	city	street	firstName	lastName	position	salary
B005	London	22 Deer Rd	John	White	Manager	40000
B005	London	22 Deer Rd	Julie	Lee	Assistant	14000
B007	Aberdeen	17 Argyll St	Mary	Howe	Assistant	15000
B003	Glasgow	163 Main St	Ann	Beech	Assistant	18000
B003	Glasgow	163 Main St	David	Ford	Supervisor	25000
B003	Glasgow	163 Main St	Susan	Brand	Manager	32000

SQL: example queries

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

1. Which persons work in London?
2. Are there any managers working in 163 Main St?
3. Whose salary is greater than average?
4. Employees in which city have the highest salary?
5. What is the lowest salary in Glasgow?

SQL: example queries (1)

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

Which persons work in London?

```
SELECT S.firstName, S.lastName FROM Branch B, Staff S
  WHERE S.branchNo = B.branchNo
  AND B.city = "London";
```

```
+-----------+----------+
| firstName | lastName |
+-----------+----------+
| John      | White    |
| Julie     | Lee      |
+-----------+----------+
```

SQL: example queries (2)

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

Are there any managers working in 163 Main St?

```
SELECT S.firstName, S.lastName FROM
  Branch B, Staff S WHERE S.branchNo = B.branchNo
  AND B.street="163 Main St" AND S.position="Manager";
```

```
+-----------+----------+
| firstName | lastName |
+-----------+----------+
| Susan     | Brand    |
+-----------+----------+
```

SQL: example queries (3)

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

Whose salary is greater than average?

```
SELECT firstName, lastName, salary FROM Staff
  WHERE salary > (SELECT AVG(SALARY) FROM Staff);
```

```
+-----------+----------+--------+
| firstName | lastName | salary |
+-----------+----------+--------+
| John      | White    |  40000 |
| David     | Ford     |  25000 |
| Susan     | Brand    |  32000 |
+-----------+----------+--------+
```

SQL: example queries (4)

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

Employees in which city have the highest salary?

```
SELECT city, salary FROM Staff INNER JOIN Branch
  ON Staff.branchNo=Branch.branchNo
  ORDER BY salary DESC LIMIT 1;
```

```
+--------+-------------+
| city   | MAX(salary) |
+--------+-------------+
| London |       40000 |
+--------+-------------+
```

SQL: example queries (5)

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

branchNo	street	city
B005	22 Deer Rd	London
B007	16 Argyll St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

What is the lowest salary in Glasgow?

```
SELECT firstName, lastName, salary FROM
  Staff INNER JOIN Branch ON Staff.branchNo=Branch.branchNo AND
  Branch.city="Glasgow" ORDER BY salary ASC LIMIT 1;
```

```
+-----------+----------+--------+
| firstName | lastName | salary |
+-----------+----------+--------+
| Ann       | Beech    |  18000 |
+-----------+----------+--------+
```

SQL: String matching

staffNo	firstName	lastName	position	salary	branchNo
SL21	John	White	Manager	40000	B005
SG37	Ann	Beech	Assistant	18000	B003
SG14	David	Ford	Supervisor	25000	B003
SA9	Mary	Howe	Assistant	15000	B007
SG5	Susan	Brand	Manager	32000	B003
SL41	Julie	Lee	Assistant	14000	B005

The *LIKE* operator performs string matching with wildcards: % matches any number of characters, _ matches exactly one character.

Whose last name begins with a "B" *or* ends with an "d"?

```
SELECT firstName, lastName FROM Staff
  WHERE ((lastName LIKE "b%") OR (firstName LIKE "%d"));
```

```
+-----------+----------+
| firstName | lastName |
+-----------+----------+
| Ann       | Beech    |
| David     | Ford     |
| Susan     | Brand    |
+-----------+----------+
```

Way Forward

Resources and further reading:

- MySQL documentation for download:
  -> `http://dev.mysql.com/doc/`
- MySQL tutorials on the Web, e.g.
  -> `https://dev.mysql.com/doc/refman/5.7/en/tutorial.html`
  -> `https://www.tutorialspoint.com/mysql/`
- SQL Quick Reference From W3Schools:
  -> `https://www.w3schools.com/sql/sql_quickref.asp`
- Lot of database books in our library