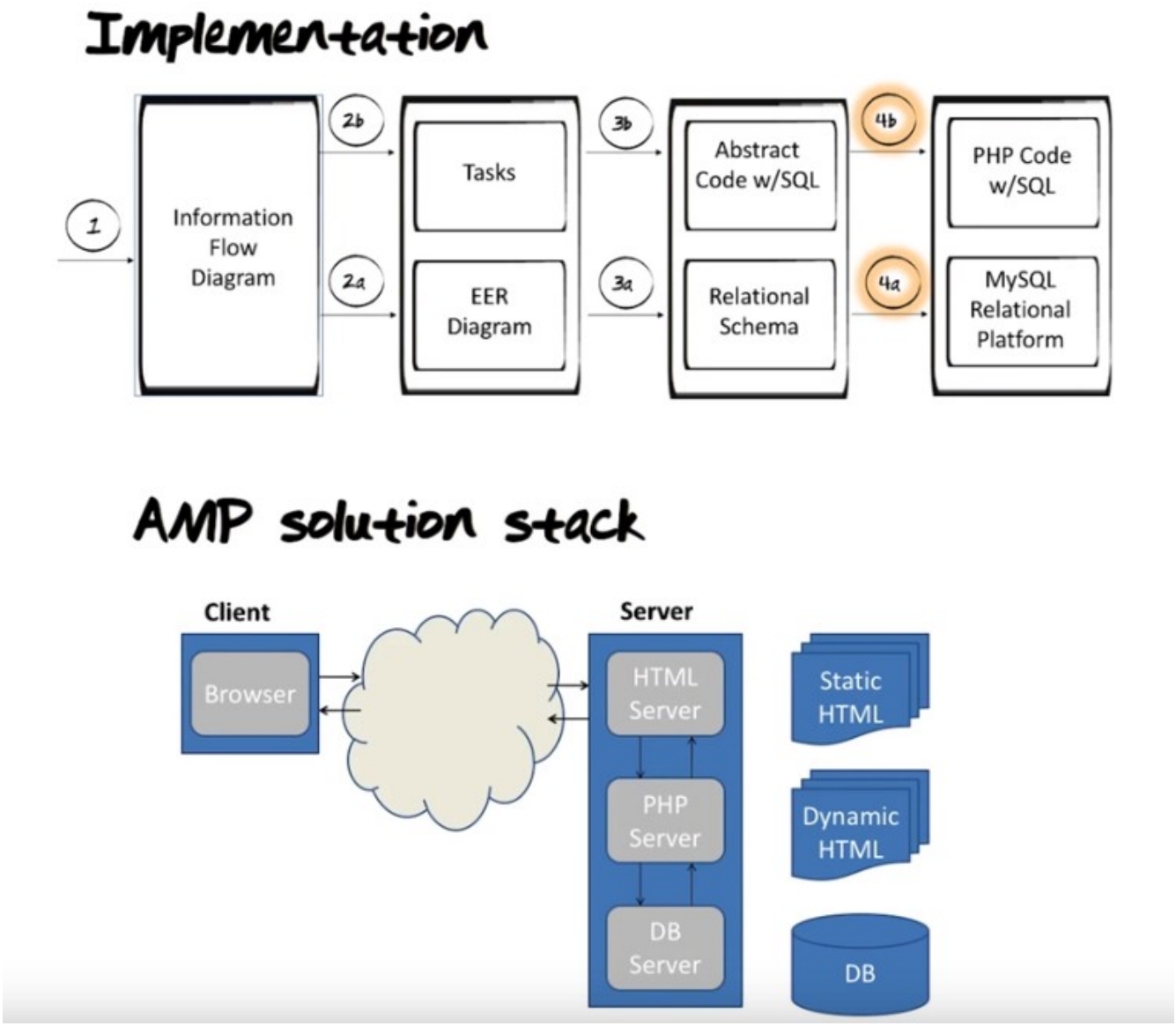


上圖中最後一列說得好.

From online:
WAMP is sometimes used as an abbreviated name for the software stack Windows, Apache, MySQL, PHP. It is derived from LAMP which stands for Linux, Apache, MySQL, and PHP. As the name implies, while LAMP is used on Linux servers, WAMP is used on Windows servers. Because WordPress isn't usually installed on Windows Servers, WAMP has become popular among developers as a method of installing WordPress on their personal computers.
The "A" in WAMP stands for Apache. Apache is server software that is used to serve webpages. Whenever someone types in your WordPress website's URL, Apache is the software that "serves" your WordPress site.
The "M" in WAMP stands for MySQL. MySQL is a database management system. It's job in the software stack is to store all of your website's content, user profiles, comments, etc.
The "P" in WAMP stands for PHP. PHP is the programming language that WordPress is written in. It is also the piece that holds the entire software stack together. It runs as a process in Apache and communicates with the MySQL database to dynamically build your webpages.

Implementation – Data first

- WAMP stack demos
- Install a WAMP/MAMP/LAMP stack
- All DBMSs come with bulk load utilities
- Load GTOnline database through phpMyAdmin using a script (downloadables)
- Play around with GTOnline database through phpMyAdmin

WAMP stack Demos

Install the WAMP stack

下載 LAMP: https://bitnami.com/stack/lamp
安裝時, 會出現 Create MySQL 'root' Account 提示, the root account is what's going to have all the access to the database.

Find the sql port:
mysql -uroot -p
mysql> SHOW GLOBAL VARIABLES LIKE 'PORT';

Run the WAMP stack

videos 8-13:

To index, or not? – many things to consider

- Size of table
- A single table can be in the access paths of multiple queries/updates
- Frequencies of reads and/or writes
- Already indexed? Some DBMSs, like MySQL, don't give you a choice
- Some advice

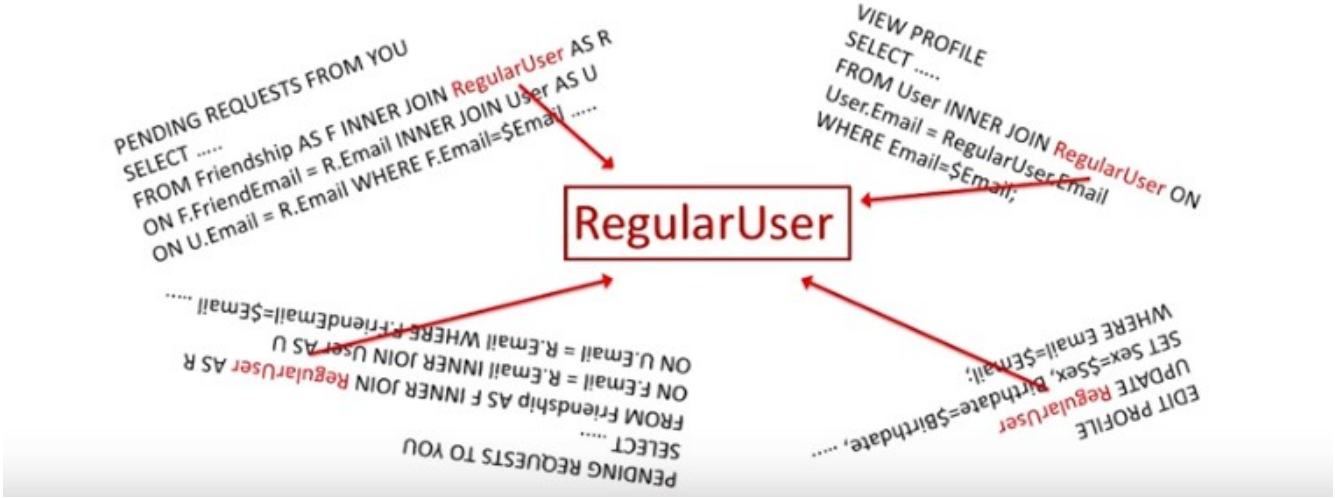
To index, or not? – size of table

- record size: 159 bytes
- block size: 4K
- filled: ~80%
- records/block: ~20
- #records: 4 million
- #blocks: ~200,000
- table size: ~800MB

```
RegularUser(
    Email varchar(50),
    Sex char(1),
    Birthdate datetime,
    CurrentCity varchar(50),
    Hometown varchar(50),
    ..... );
```

Could cost 200,000 x 0.01sec = 33 min to scan if not clustered and/or indexed

To index, or not? - multiple access paths add up



To index, or not? – reads and/or writes?

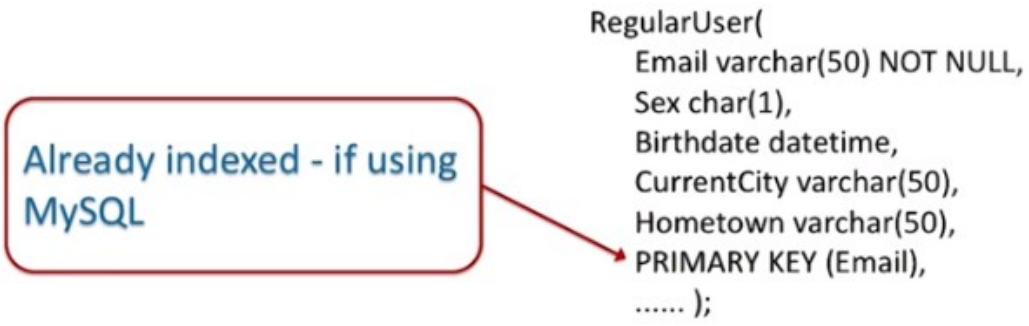
Reads are faster if table is indexed.
Writes are slower if table is indexed, BUT writes are preceded by reads!

VIEW PROFILE
SELECT .....
FROM User INNER JOIN RegularUser ON
User.Email = RegularUser.Email
WHERE Email=$Email;

EDIT PROFILE
UPDATE RegularUser
SET Sex=$Sex, Birthdate=$Birthdate, ...
WHERE Email=$Email;

NEW USER
INSERT into RegularUser
(Email, Sex, Birthdate, ...)
VALUES (.....)

To index, or not? – already indexed?



To index, or not? – some advice

- Pay attention to size
- Multiple frequencies add up
- Writes suffer when indices must be maintained
- Some DBMS come with tools to measure performance
- Some DBMS show you the query graphs
- Think globally, act locally
- Compare before and after performance
- Learn about efficiency, physical organization, query optimization