

# **Technologies for E- Commerce**

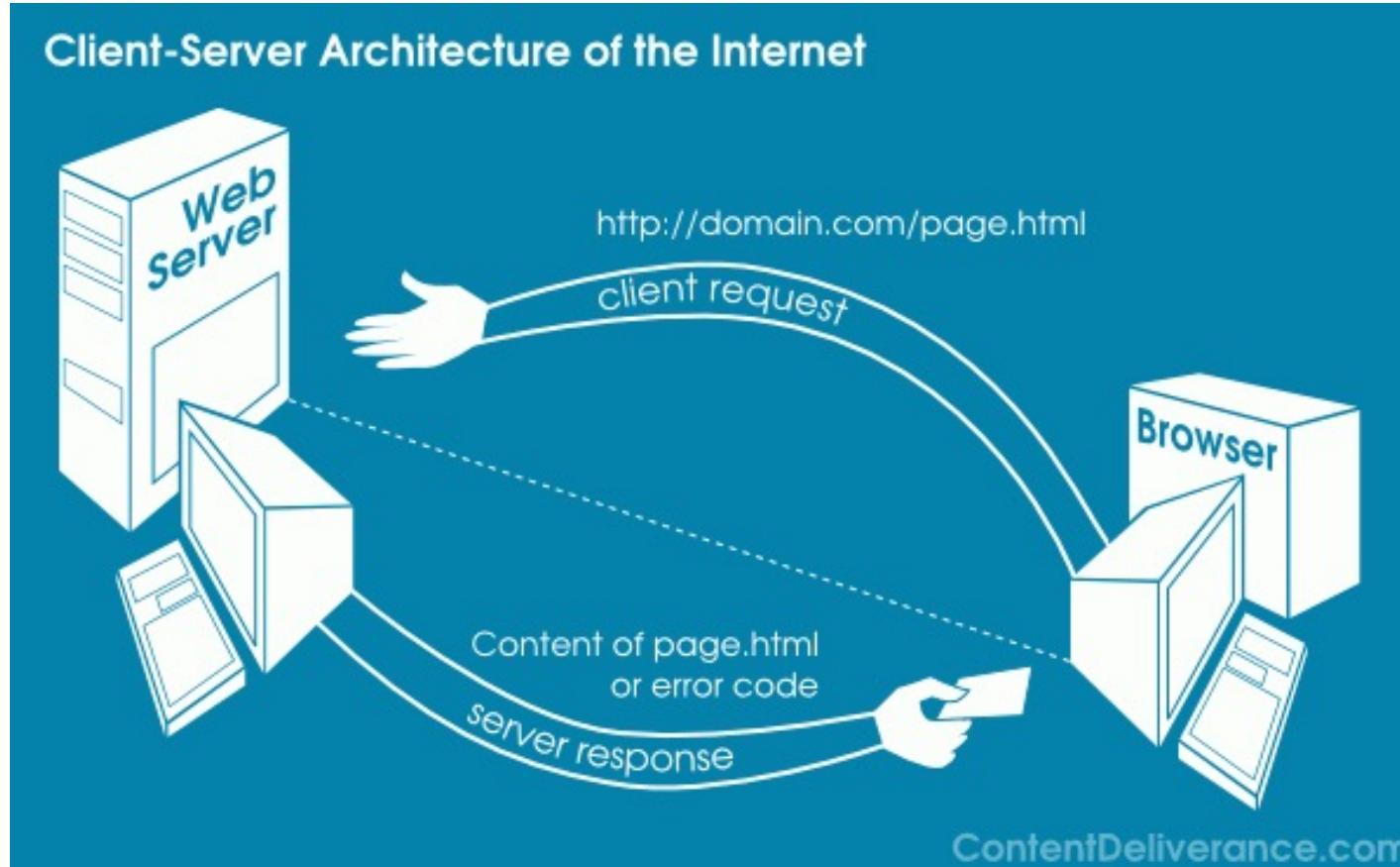
**CAN302**

**Department of Communications and Networking  
Xi'an Jiaotong-Liverpool University (XJTLU)**

# Week 4 – Server page and Database

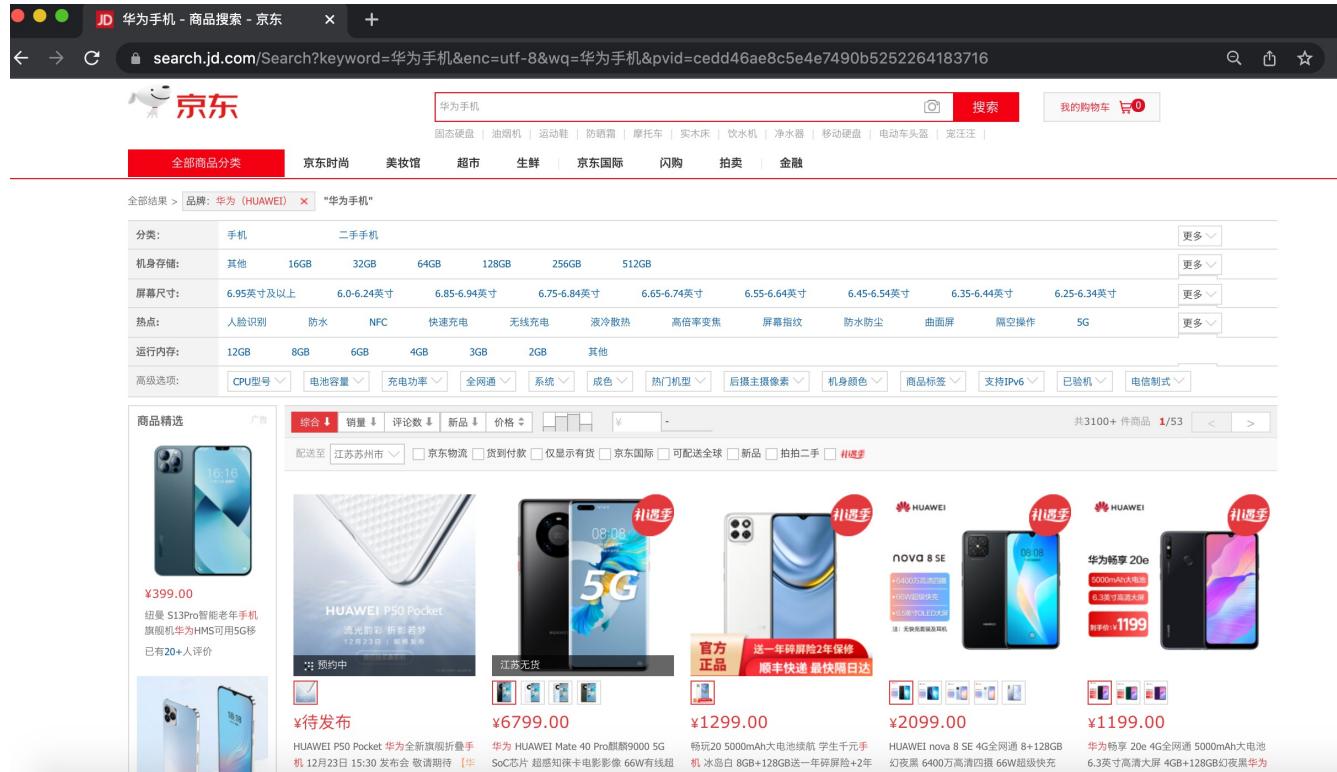


# Static Web



<http://courses.haigarmen.com/intd416/client-server-architecture/>

# Dynamic web: frame + data



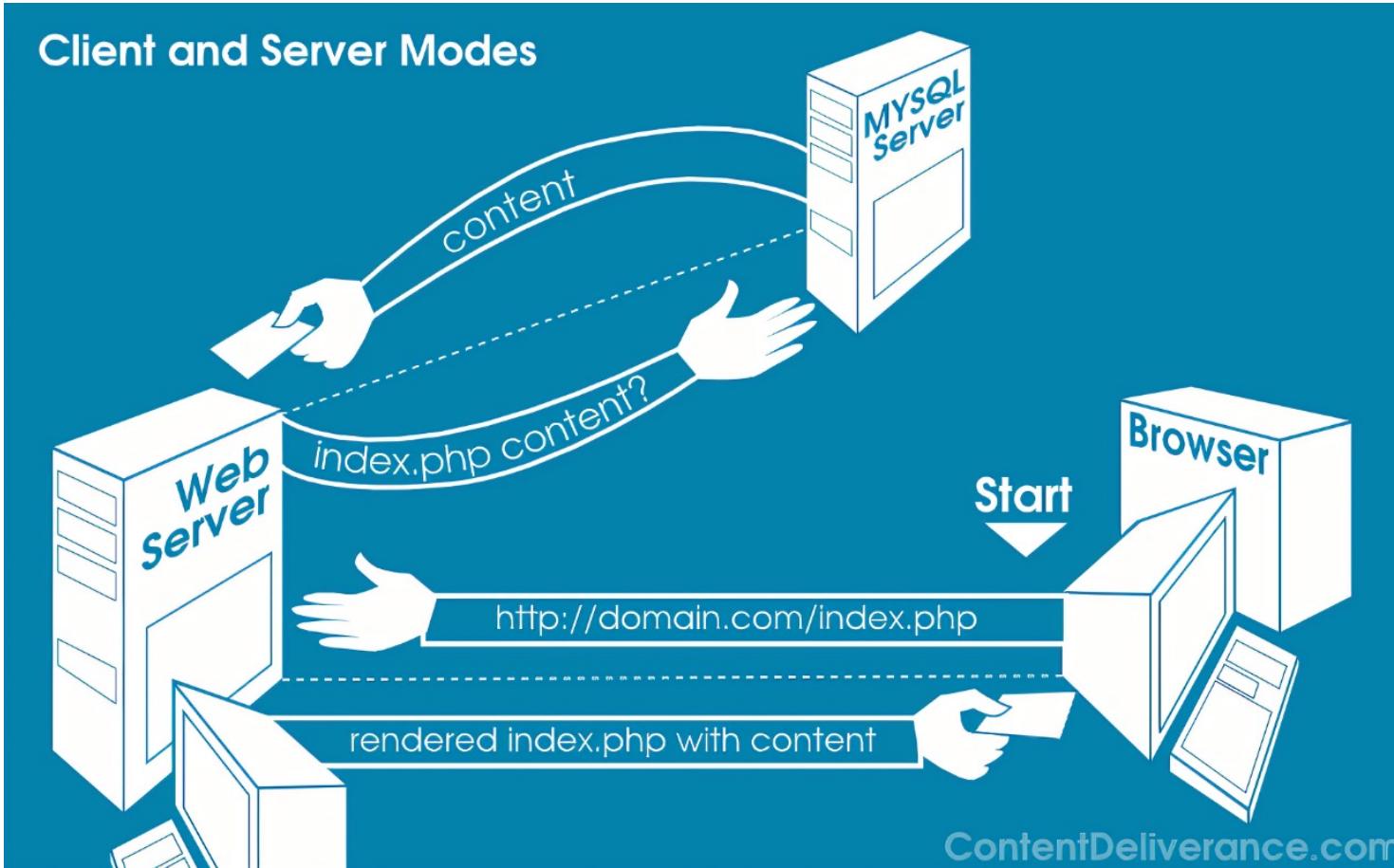
- Like a supermarket, we need to put products on the shelf in e-market.
- Manually update the products is not a possible way.
- It's time for dynamic web:  
**A dynamic website** is the webpage on the server side where different contents are shown when assessed at different timings.

**ASP, JSP, PHP:** generate the html **dynamically**.

<https://www.educba.com/dynamic-website/>

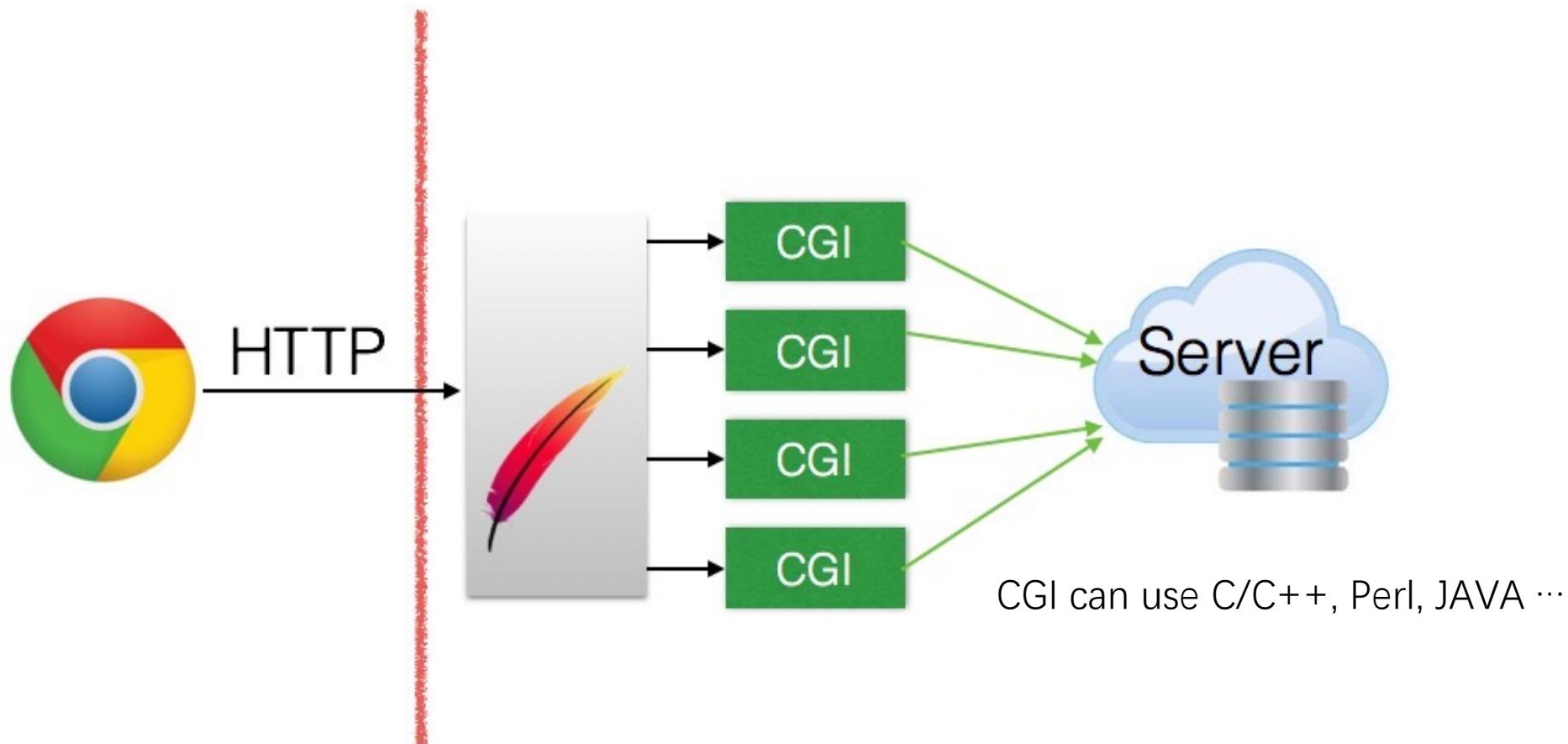
<https://blog.csdn.net/dhp1027/article/details/7725653>

# Web browser to DB



- Access DB through internet is very dangerous.
- web browser cannot fetch data from DB.
- We need web server interacts with DB.
- Static web server cannot make it.
- We need web server to be dynamic .

# CGI - Common Gateway Interface

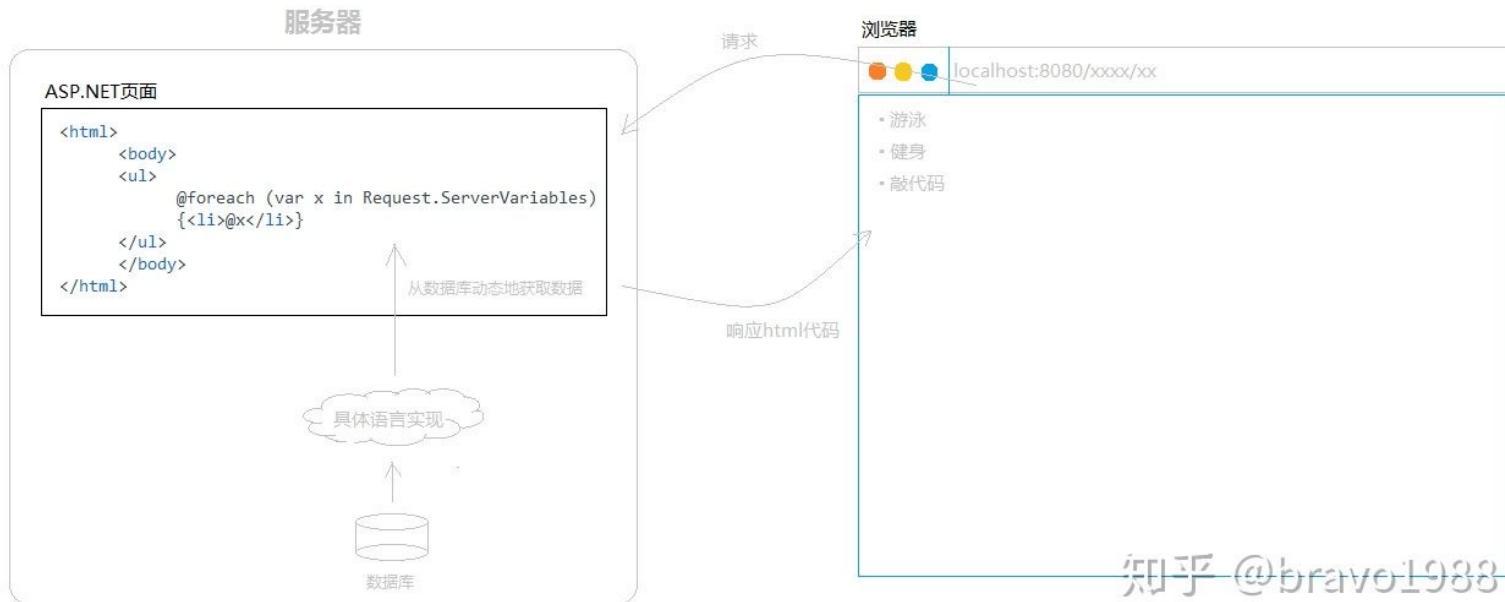


# Sever page



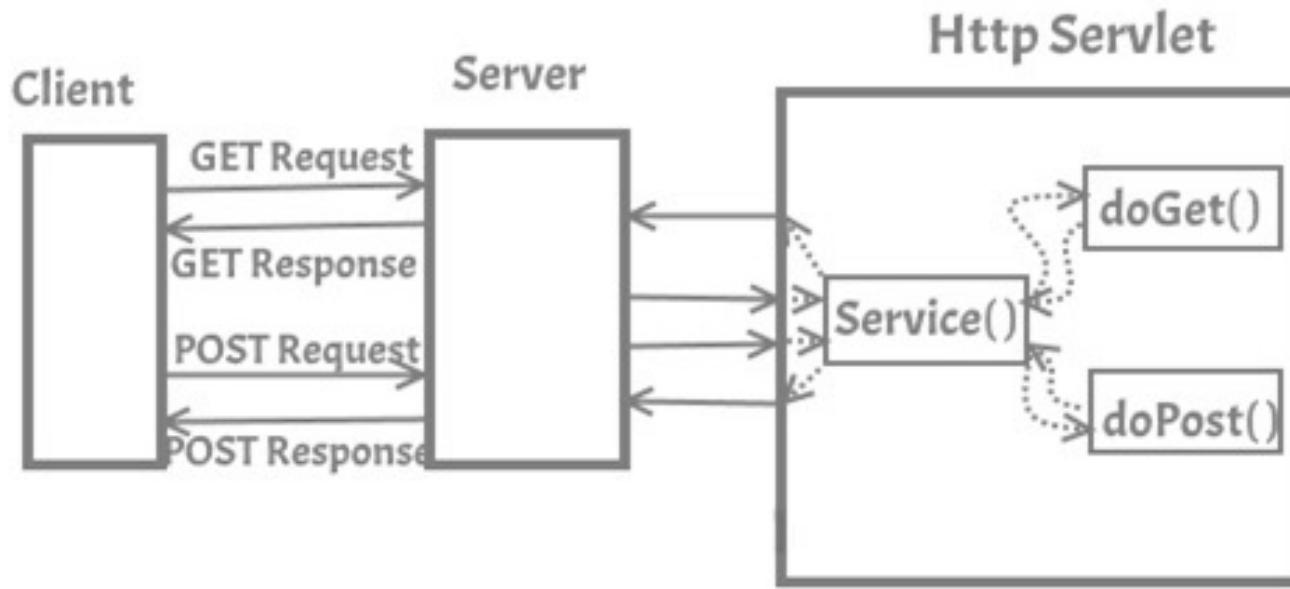
- ASP (Active Sever Page), 1996 by Microsoft.
  - Using VBScript/JavaScript to generate the dynamic html content
  - Calling the com components to extend its ability.
  - ASP.NET supports .NET languages, such as: C#, J#
- JSP (Java Server Page), 1998 by SUN.
  - It is JAVA but **NOT** JavaScript.
  - Html + JAVA code in one jsp file.
- PHP
  - 1.0 in 1994 by Rasmus Lerdorf, as Personal Home Page Tools, a CGI with C language.
  - 3.0 in 1998, full name changed to Hypertext Preprocessor, a new type of script language.

# Glance at ASP



- Embedded JS code in html file.
- The dynamic info (such as products, customers, orders...) can be stored in database.
- On request, the web server will run JS code (such as trace info from database) and generate the corresponding real-time html codes.
- Then, web server will merge the static codes and dynamic generated codes as a whole html and sent it back to the browser.

# JAVA Servlet



- If we treat CGI as a concept, Servlet can be regard as a Java type CGI.
- While it is very different from traditional CGI by C or Perl.
- To have the dynamic Html content, programmer need to write a lot of "System.out.println()".

<https://www.educba.com/what-is-servlet/>

<https://techdifferences.com/difference-between-cgi-and-servlet.html>

# JAVA Servlet

```
TestServlet.java x
1 package cn.itcast.ajax;
2
3 import java.io.IOException;
4
5 public class TestServlet extends HttpServlet {
6
7     /**
8      *
9      */
10    private static final long serialVersionUID = 1L;
11
12    /**
13     * The doGet method of the servlet. <br>
14     *
15     * This method is called when a form has its tag value method equals to get.
16     *
17     * @param request the request send by the client to the server
18     * @param response the response send by the server to the client
19     * @throws ServletException if an error occurred
20     * @throws IOException if an error occurred
21     */
22    public void doGet(HttpServletRequest request, HttpServletResponse response)
23        throws ServletException, IOException {
24
25        response.setContentType("text/html");
26        PrintWriter out = response.getWriter();
27        out.println("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">");
28        out.println("<HTML>");
29        out.println("  <HEAD><TITLE>A Servlet</TITLE></HEAD>");
30        out.println("  <BODY>");
31        out.print("    This is ");
32        out.print(this.getClass());
33        out.println(", using the GET method");
34        out.println("  </BODY>");
35        out.println("</HTML>");
36        out.flush();
37        out.close();
38    }
39}
40}
41}
42}
43}
44}
```

知乎 @bravo1988

## 骨灰级程序员



<https://zhuanlan.zhihu.com/p/42343690>

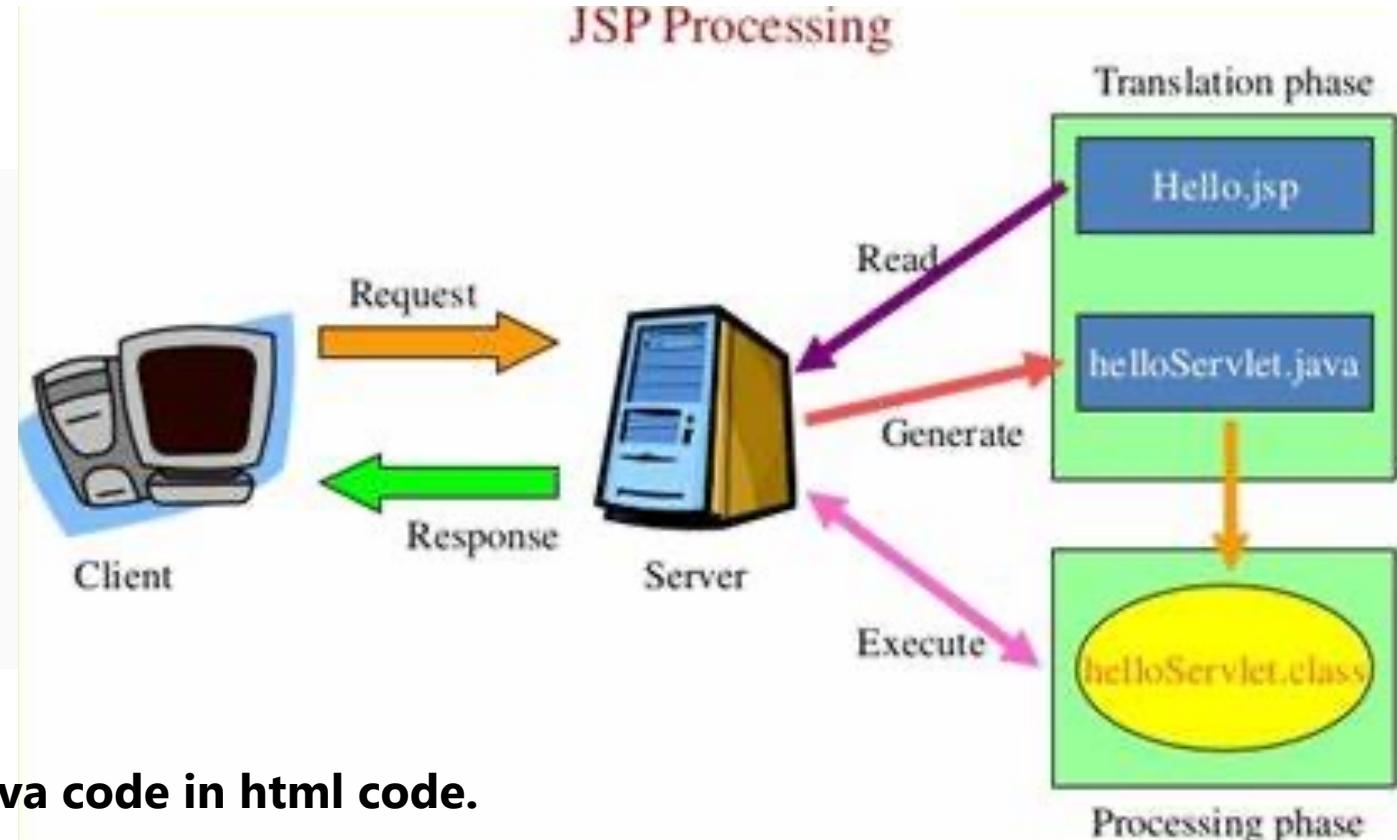
# Sample of Servlet



知乎 @bravo1988

# Jsp, more programmer friendly

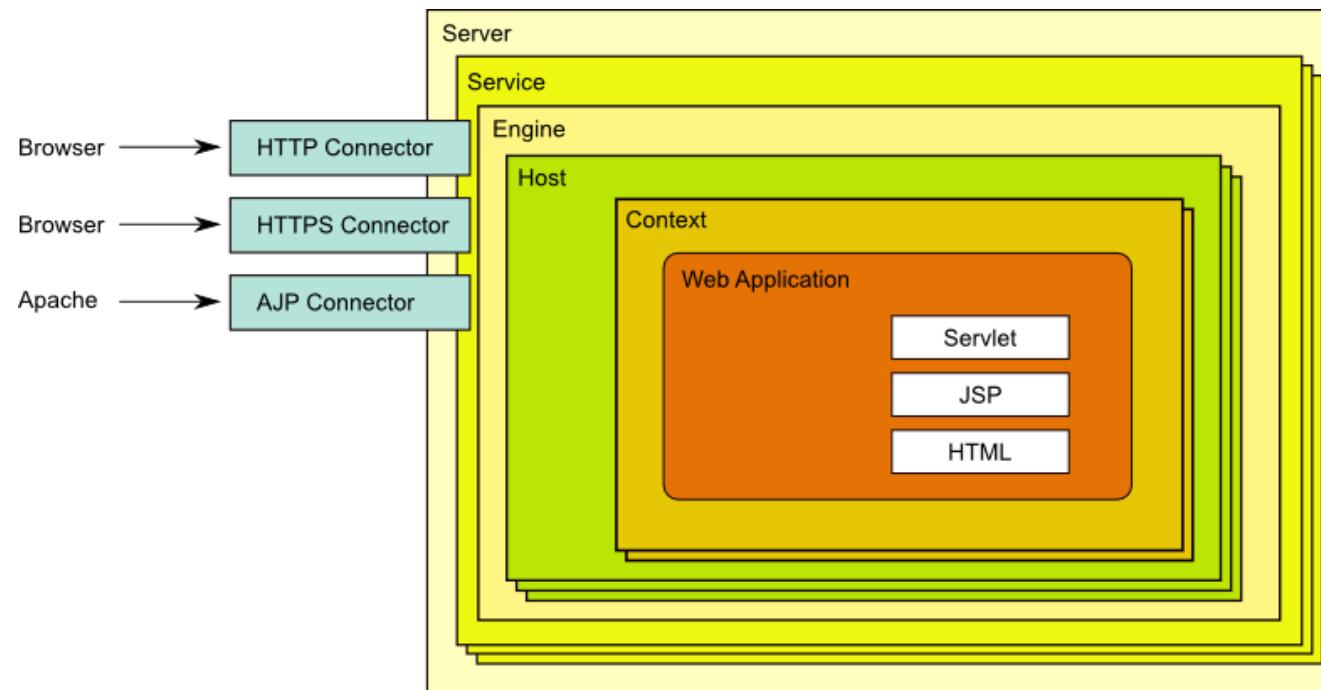
```
1 <html>
2   <head>
3     <title>第一个 JSP 程序</title>
4   </head>
5   <body>
6     <%          out.println("Hello World! ");
7       %>
8     </body>
9   </html>
```



- **Embed the java code in html code.**
- **Auto transfer it to Java code.**

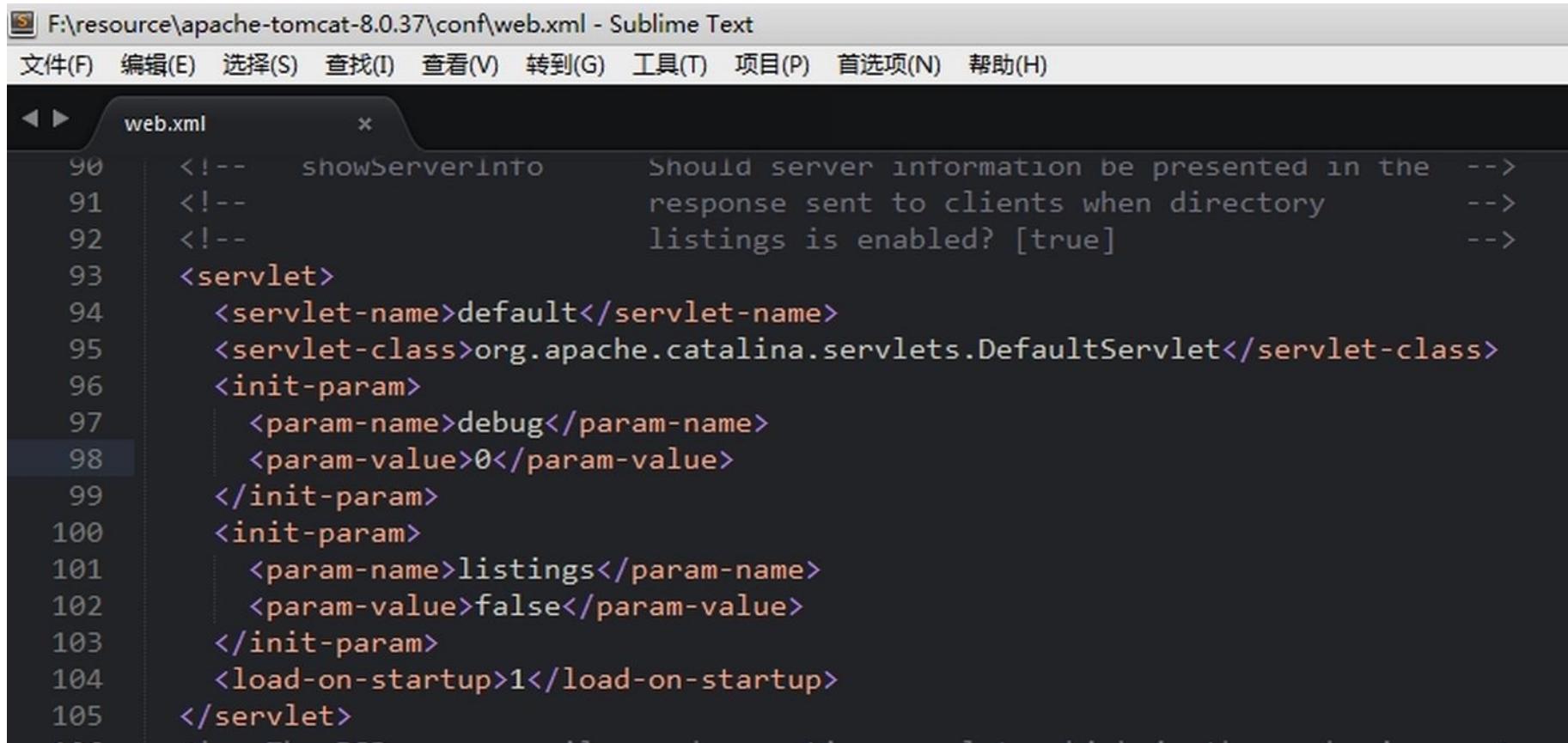
<https://blog.csdn.net/jie1521836/article/details/81323942>  
<https://zhuanlan.zhihu.com/p/42343690>

# Tomcat



- The Apache Tomcat **server** is Java-based, open source reference implementation (RI) web application and servlet container created to run servlets and Java Server Pages (JSP) based web applications.
- Again, Tomcat server is a **software**.
- It can work together with Apache server.

# Tomcat + Static web server



```
F:\resource\apache-tomcat-8.0.37\conf\web.xml - Sublime Text
文件(F) 编辑(E) 选择(S) 查找(I) 查看(V) 转到(G) 工具(T) 项目(P) 首选项(N) 帮助(H)

web.xml
90     <!-- showServerInfo      Should server information be presented in the -->
91     <!-- response sent to clients when directory -->
92     <!-- listings is enabled? [true] -->
93     <servlet>
94         <servlet-name>default</servlet-name>
95         <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
96         <init-param>
97             <param-name>debug</param-name>
98             <param-value>0</param-value>
99         </init-param>
100        <init-param>
101            <param-name>listings</param-name>
102            <param-value>false</param-value>
103        </init-param>
104        <load-on-startup>1</load-on-startup>
105    </servlet>
```

- Tomcat processes everything as JAVA.
- For static resources (such as pics, css, js…), they are unnecessary load for Tomcat.
- Uses static web server (Like Apache, nginx) + Tomcat combination can achieve better performance.

# PHP

```
<html>
<body>

<?php
for ($i=1; $i<=5; $i++)
{
    echo "The number is " . $i . "<br />";
}
?>

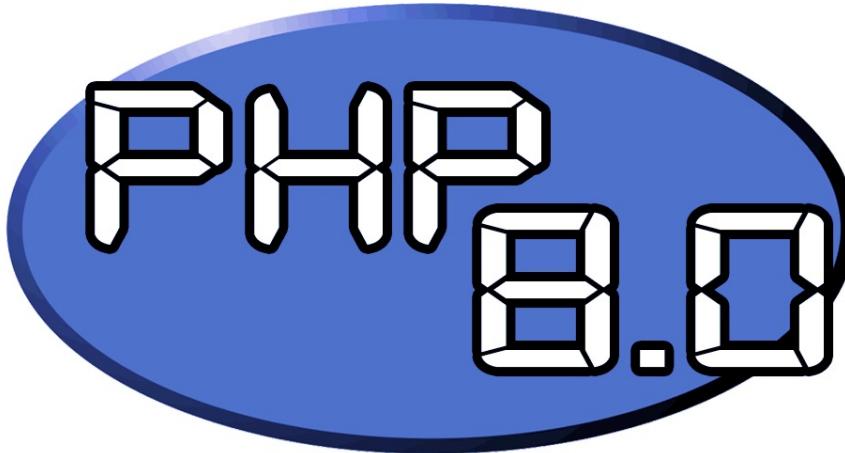
</body>
</html>
```

Output:

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

- PHP works in a similar way, embedded the scripts in html codes.
- PHP is optimized for Web development and can be used for general-purpose (However almost nobody use it beyond web).
- PHP is the widely-used, free, and still active.
- Apache have extension that support PHP.
- XAMPP is combination of all necessary software.

# Discussion



**Why PHP is still active but asp/jsp fade out?**

# Database basics



- Similar to **web server**, **database** is a software.
- A database is an organized collection of structured information, or data, typically stored electronically in a computer system.
- A database is usually controlled by a database management system (DBMS). Together, the data and the **DBMS**, along with the applications that are associated with them, are referred to as a database system.
- We often call a certain database which is actually the name of DBMS.

# Database basics

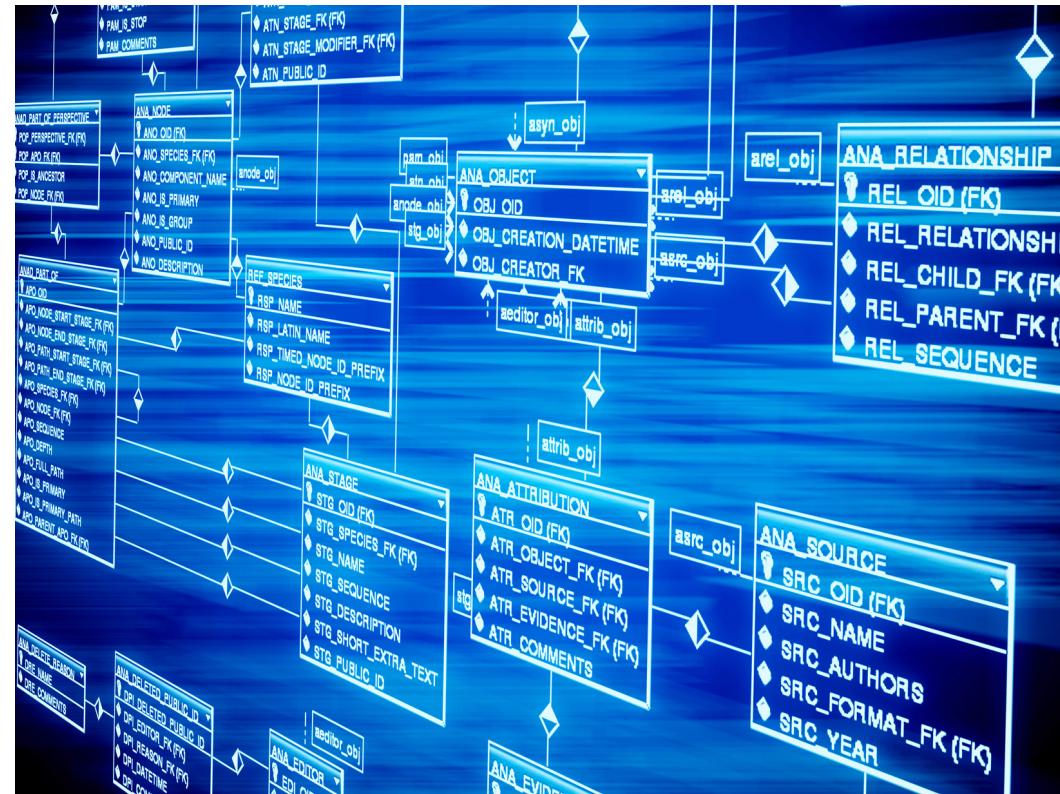
The screenshot shows a database interface with three tabs: Statement 1, Statement 2, and Database Explorer 3. Statement 1 contains the following SQL code:

```
1SELECT reported.city,
2     reported.count_accidents::float /
3     (reported.count_accidents
4      + not_reported.count_accidents)::float AS ratio
5FROM
6  (SELECT city, COUNT(*) AS count_accidents
7   FROM accidents
8   WHERE reported = 'true'
9   GROUP BY city) as reported
10JOIN
11  (SELECT city, COUNT(*) AS count_accidents
12   FROM accidents
13   WHERE reported = 'false'
14   GROUP BY city) as not_reported
15ON reported.city = not_reported.city
16ORDER BY ratio DESC
17LIMIT 3;
```

The Result 1 tab displays the query results in a table:

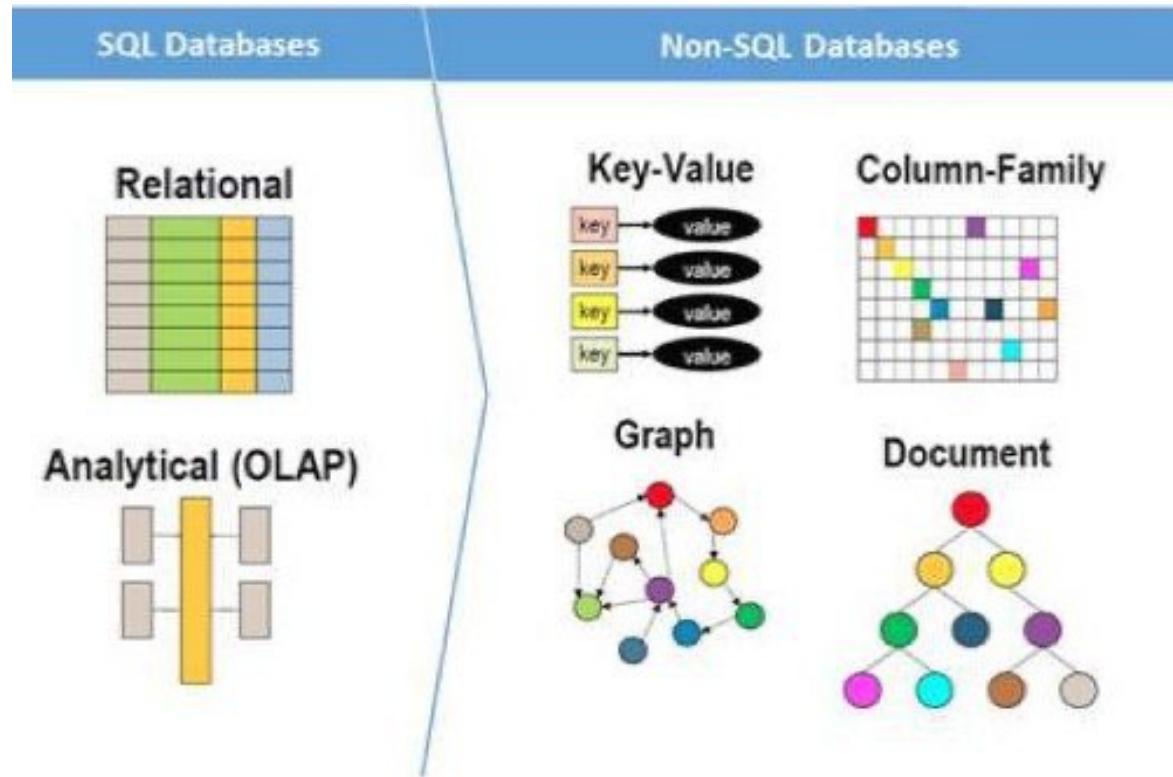
city	ratio
ROSTOCK	0.67
MAGDEBURG	0.67
BREMEN	0.55

At the bottom, there are performance metrics: L:16 C:20 | 0.21s | Timeout: 0 | Max. Rows: 0 | 1-3/3.



- SQL is a programming language used by nearly all [relational databases](#) to query, manipulate, and define data, and to provide access control.

# Database basics

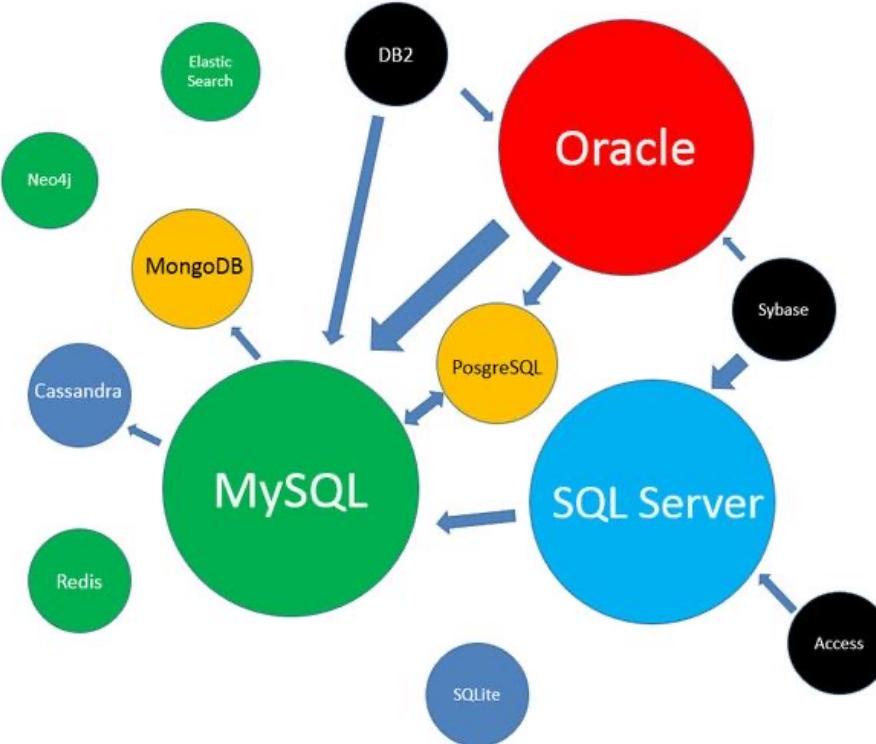


NoSQL



- Be careful, NoSQL means **non** but NOT **no**

# Database basics



OceanBase  
A Powerful Database System

TDSQL  
腾讯分布式数据库

HUAWEI | GaussDB  
Better DB Better Life

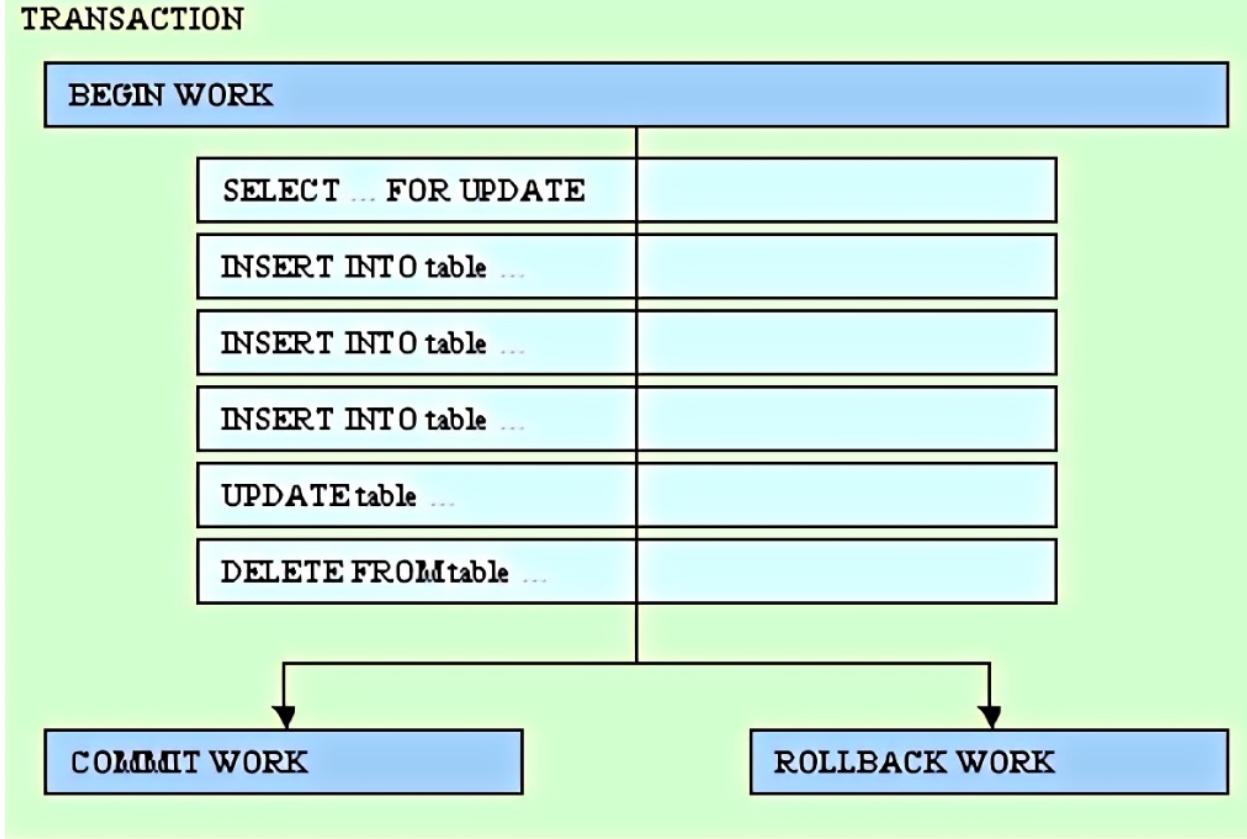
POLARDB

CynosDB



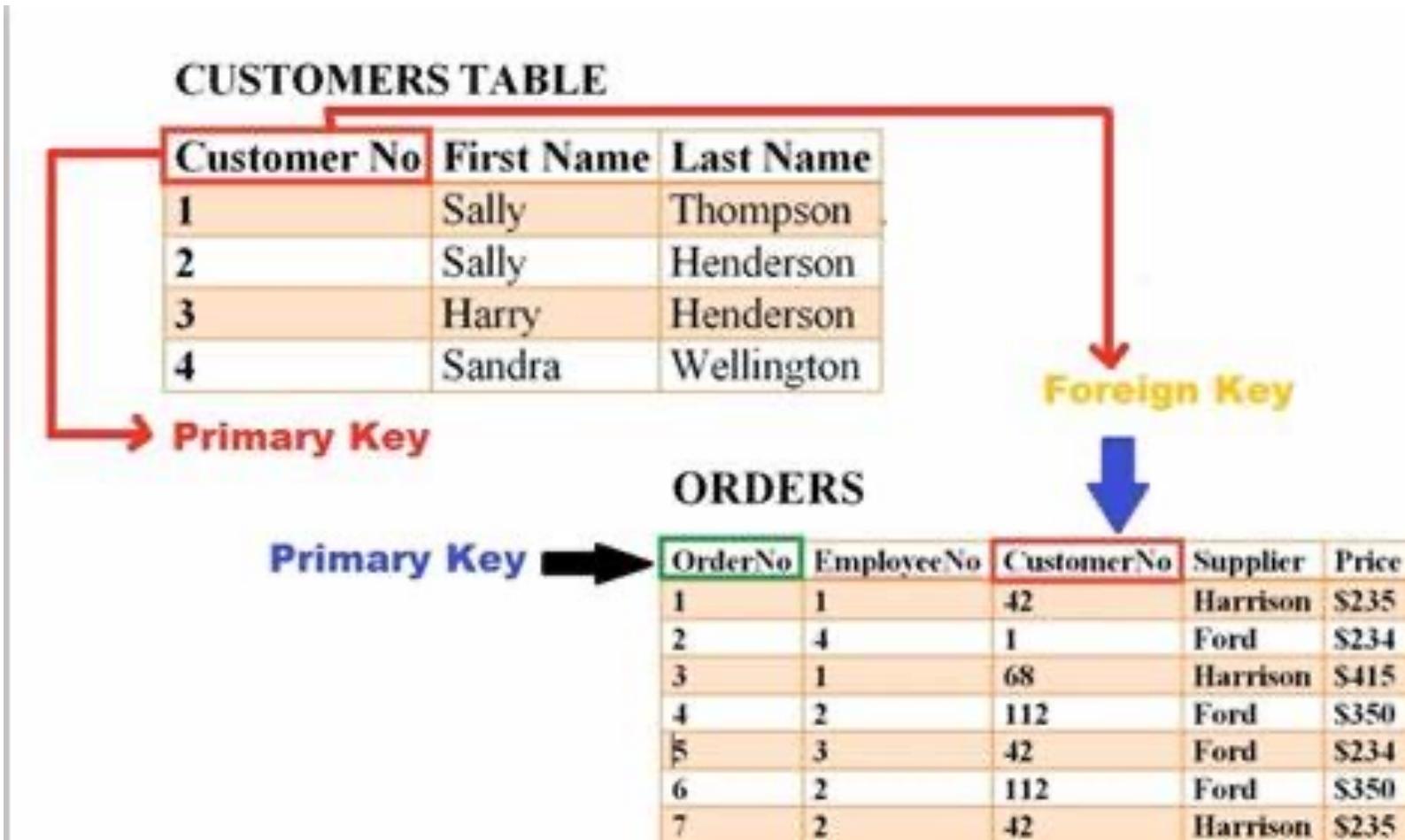
- Main DBMS on the market.

# Transaction: ACID



- **Atomicity** – ensures that all operations within the work unit are completed successfully. Otherwise, the transaction is aborted at the point of failure and all the previous operations are rolled back to their former state.
- **Consistency** – ensures that the database properly changes states upon a successfully committed transaction.
- **Isolation** – enables transactions to operate independently of and transparent to each other.
- **Durability** – ensures that the result or effect of a committed transaction persists in case of a system failure.

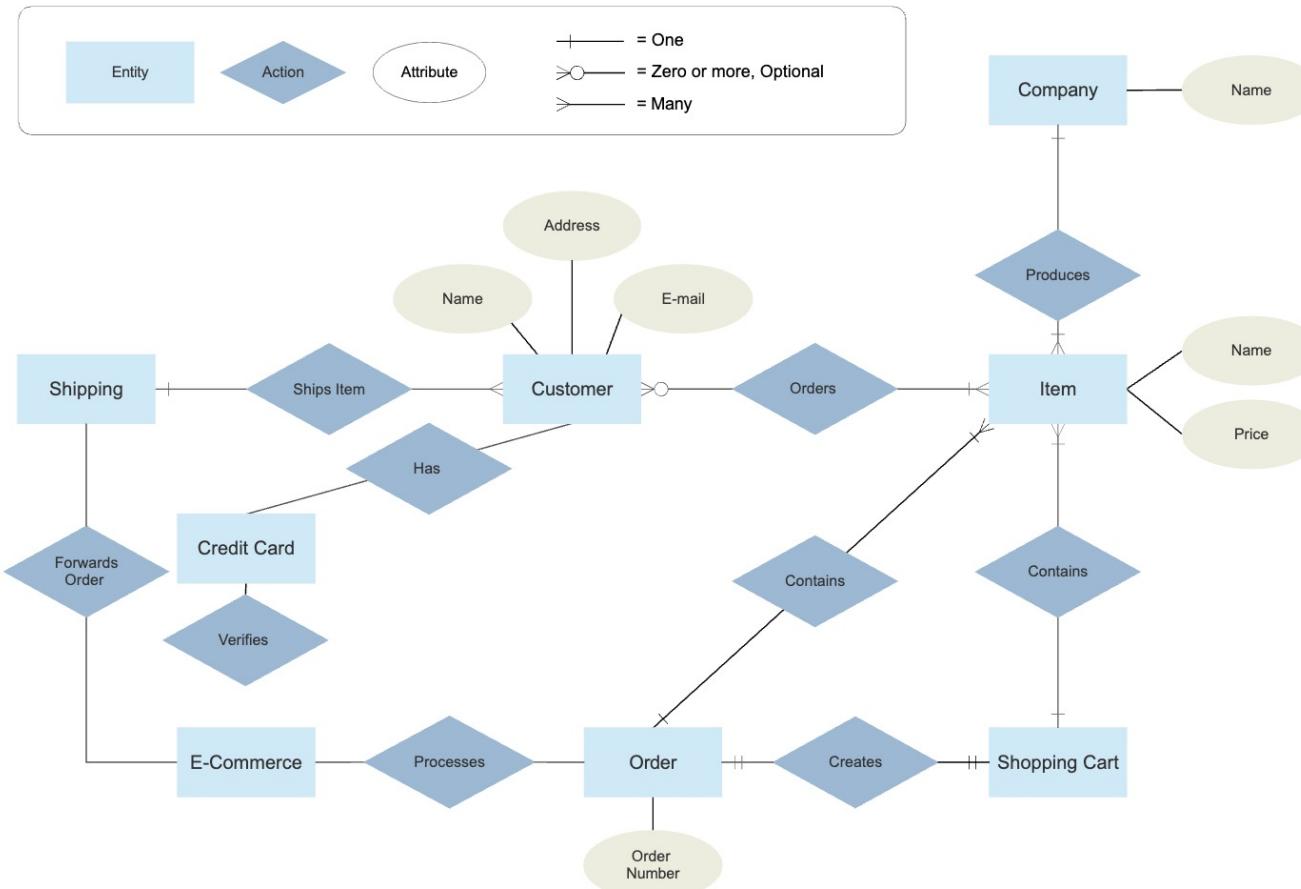
# Primary and foreign keys



- **Primary key** – ensures data is unique.
- **Foreign key** – ensures the reference relationship

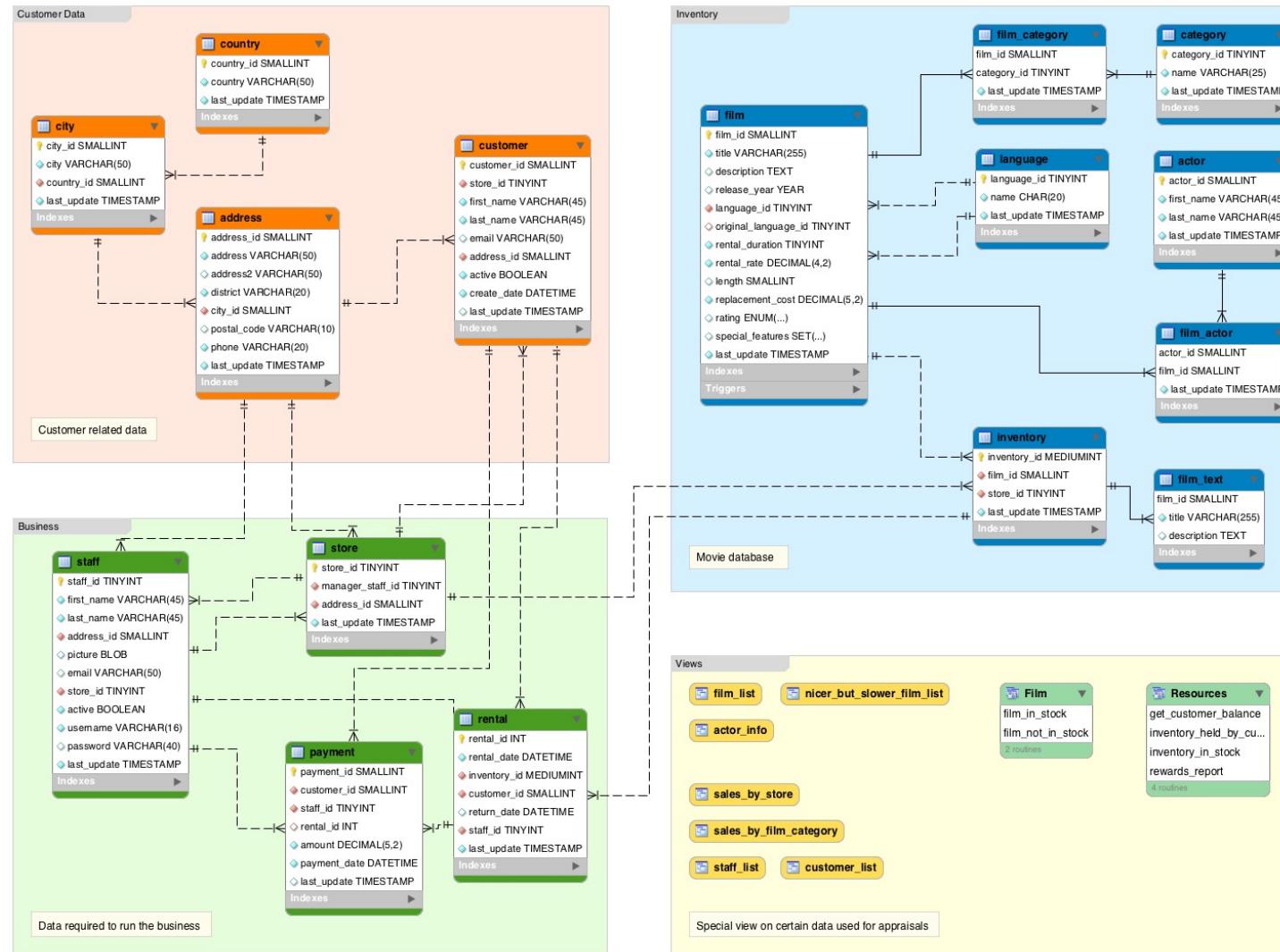
# Entity Relationship Diagram

Entity Relationship Diagram - Internet Sales Model



- An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database.

# Entity Relationship Diagram

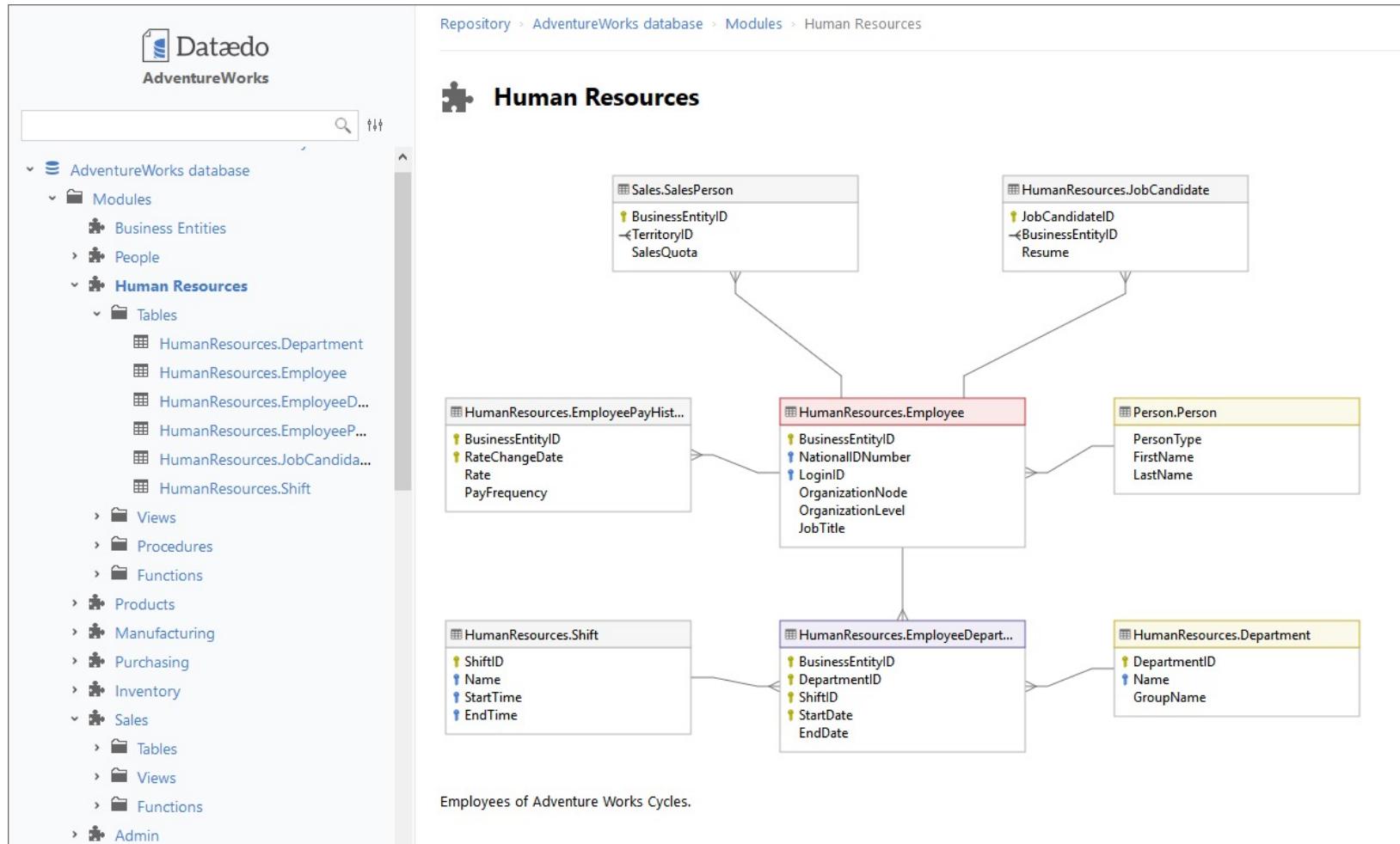


- It will help to see the data structure of whole design

# Entity Relationship Diagram

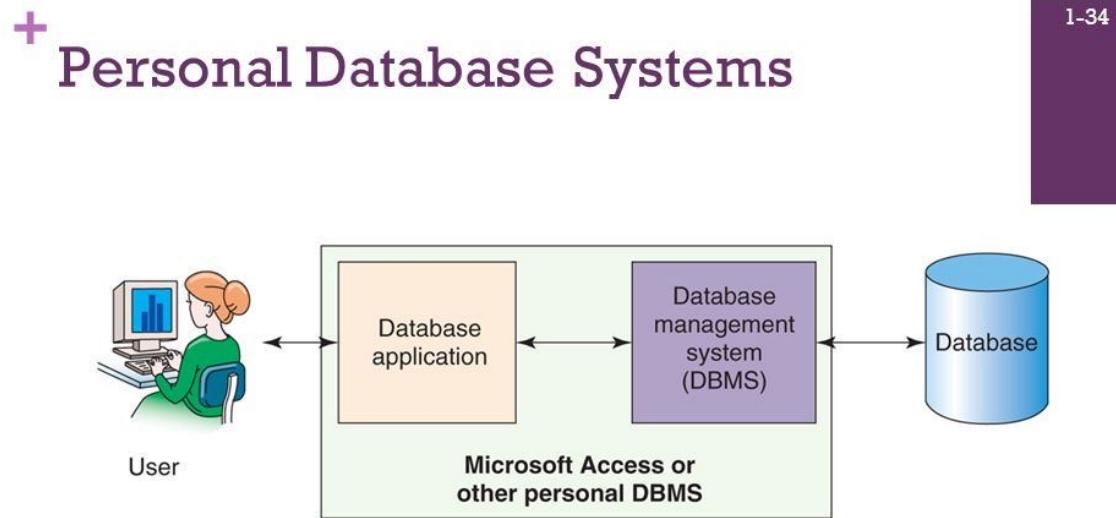


# Entity Relationship Diagram

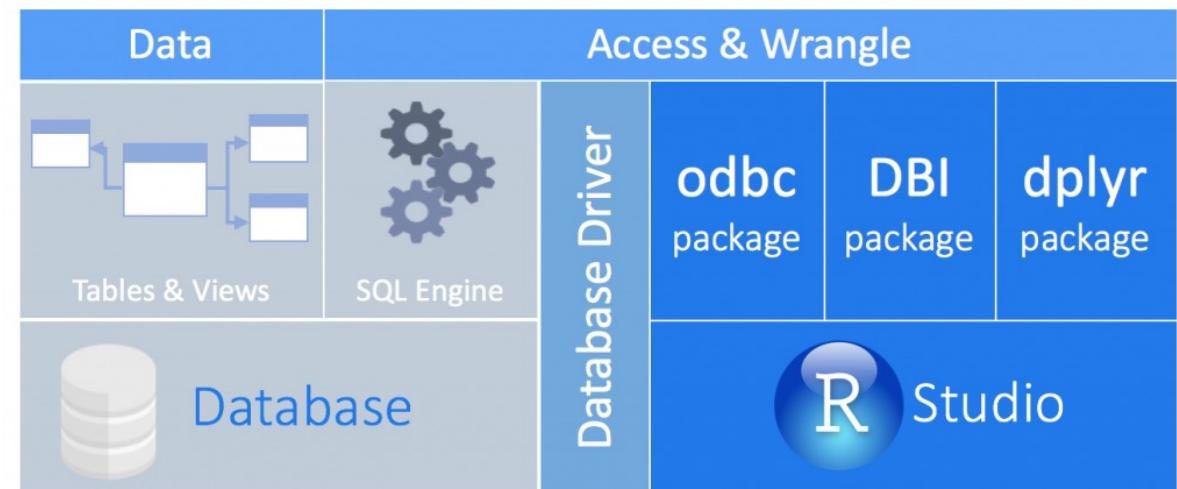


<https://dataedo.com/blog/you-dont-need-an-er-diagram-to-understand-your-database>

# Connect DB



## Commercial Databases



<https://dev.mysql.com/doc/refman/5.6/en/transport-protocols.html>

# Connect MySQL DB

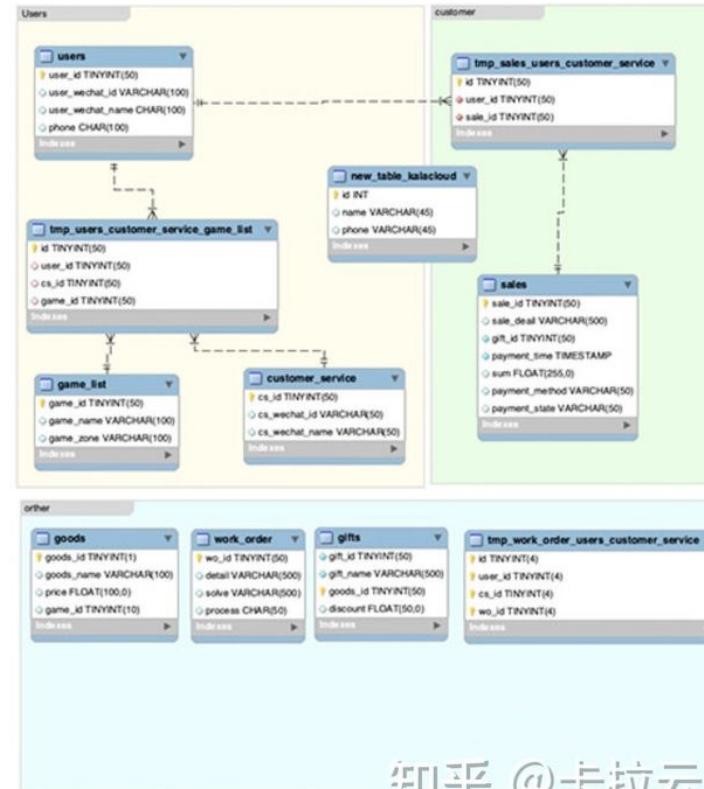
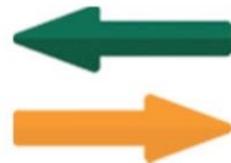
--protocol Value	Transport Protocol Used	Applicable Platforms
TCP	TCP/IP	All
SOCKET	Unix socket file	Unix and Unix-like systems
PIPE	Named pipe	Windows
MEMORY	Shared memory	Windows

- **DB is a client/server style software.**
- **DBMS works as the server.**
- **Different DB supports different way to connect, the above shows transport protocols supported by MySQL.**

<https://dev.mysql.com/doc/refman/5.6/en/transport-protocols.html>

# Tools for DB – MySQL Workbench

```
25
26 DROP TABLE IF EXISTS `customer_service`;
27
28 CREATE TABLE `customer_service` (
29   `cs_id` tinyint(50) NOT NULL,
30   `cs_wechat_id` varchar(50) DEFAULT NULL,
31   `cs_wechat_name` varchar(50) DEFAULT NULL,
32   PRIMARY KEY (`cs_id`)
33 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
34
35 LOCK TABLES `customer_service` WRITE;
36 /*!40000 ALTER TABLE `customer_service` DISABLE KEYS */;
37
38 INSERT INTO `customer_service` (`cs_id`, `cs_wechat_id`,
39 VALUES
40   (1, 'VIP01', '开心客服'),
41   (2, 'VIP02', '客服总监'),
42   (3, 'VIP_game_super', '霸道总裁');
43
44 /*!40000 ALTER TABLE `customer_service` ENABLE KEYS */;
45 UNLOCK TABLES;
46
47 kalacloud.com
48
49 # Dump of table game_list
50 #
51
52 DROP TABLE IF EXISTS `game_list`;
53
54 CREATE TABLE `game_list` (
55   `game_id` tinyint(50) NOT NULL AUTO_INCREMENT,
56   `game_name` varchar(100) DEFAULT NULL,
57   `game_zone` varchar(100) DEFAULT NULL,
58   PRIMARY KEY (`game_id`)
59 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
60
61 LOCK TABLES `game_list` WRITE;
62 /*!40000 ALTER TABLE `game_list` DISABLE KEYS */;
63
64 INSERT INTO `game_list` (`game_id`, `game_name`, `game_20
65 VALUES
66   (1, '传奇世界', '美服'),
67   (2, '梦幻西游', '战服'),
68   (3, '征途', '全面战争');
69
70 /*!40000 ALTER TABLE `game_list` ENABLE KEYS */;
71 UNLOCK TABLES;
72
73
```

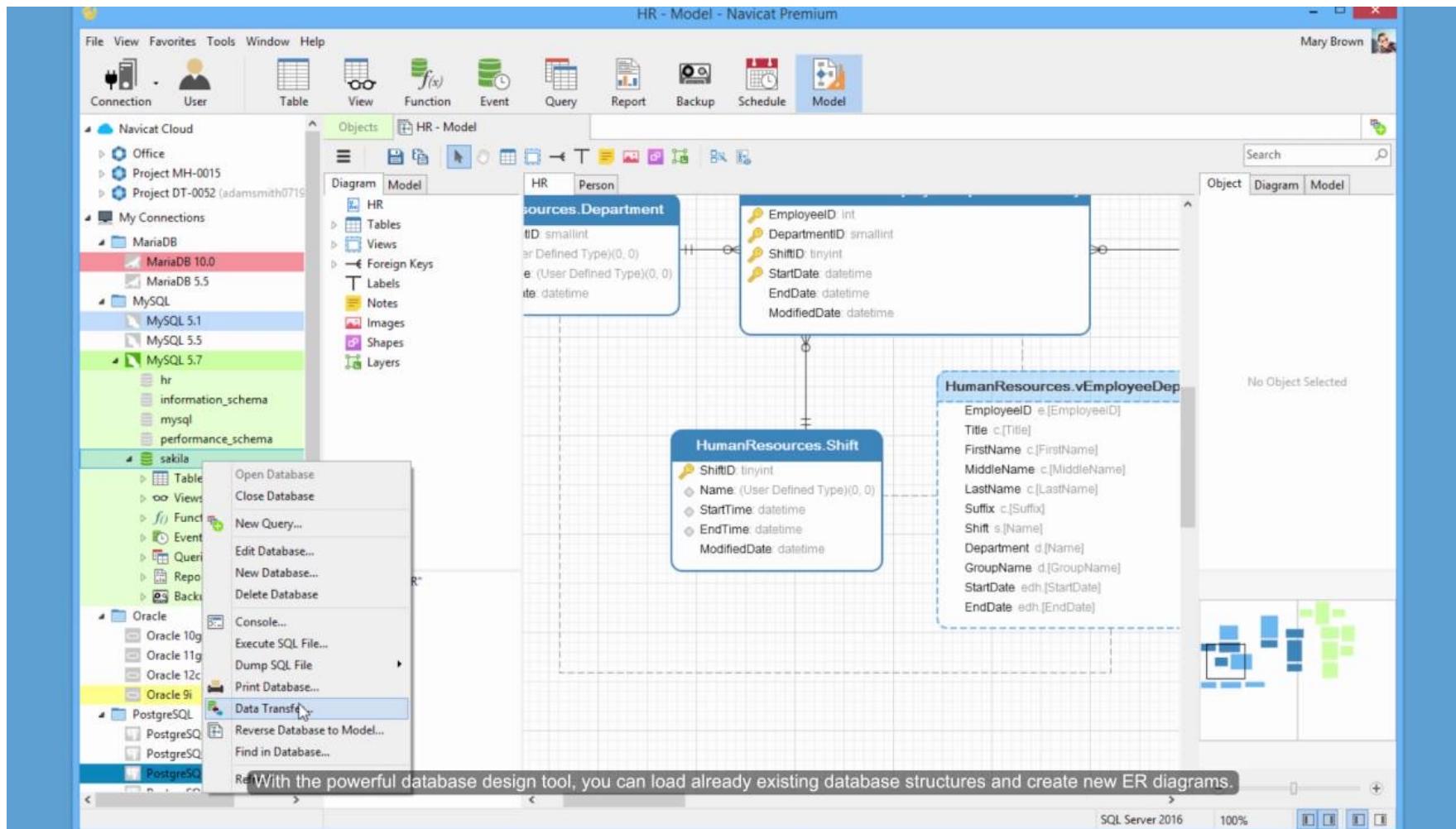


知乎 @卡拉云

<https://www.zhihu.com/question/44003854>

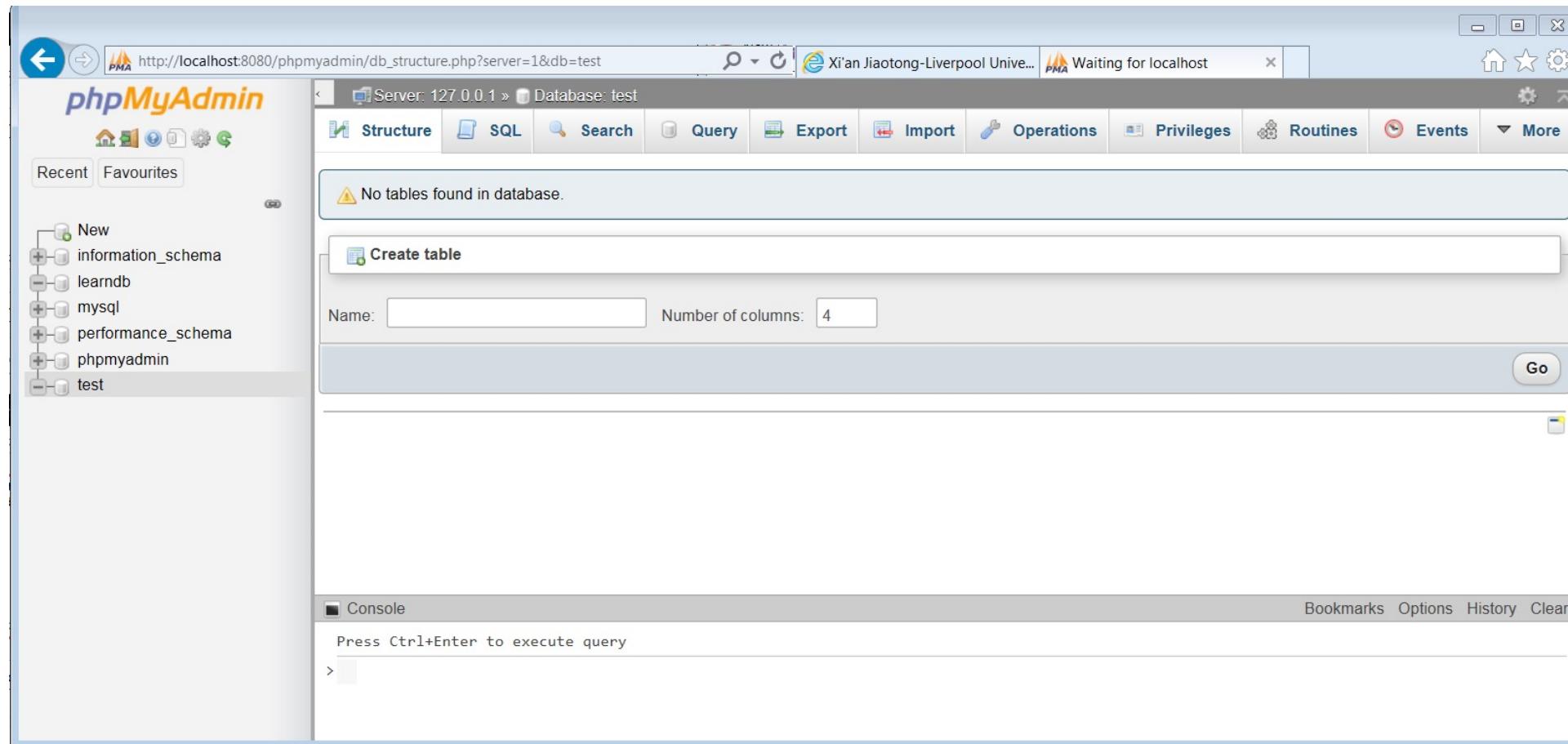
<https://www.mysql.com/products/workbench/>

# Tools for DB – Navicat



<https://navicat.com/en>

# Tools for DB – PHPMyAdmin



<http://localhost/phpmyadmin/>

# Create a table by PHPMyAdmin

The screenshot shows the PHPMyAdmin interface for creating a table named 'user'. The table has four columns: 'id', 'first', 'last', and 'email'. The 'id' column is defined as an int(11) type with AUTO\_INCREMENT, set as primary and auto-increment. The other three columns are varchar(20) and varchar(50) respectively, both using utf8mb4\_general\_ci collation.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)	utf8mb4_general_ci		No	None	AUTO_INCREMENT		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	first	varchar(20)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	last	varchar(20)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
4	email	varchar(50)	utf8mb4_general_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Below the table structure, there are buttons for 'Check all', 'With selected:', and various column operations like 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'. There are also buttons for 'Add to central columns' and 'Remove from central columns'.

At the bottom, there are links for 'Print', 'Propose table structure', 'Track table', 'Move columns', and 'Normalize'. A search bar allows adding 1 or more columns after 'email'.

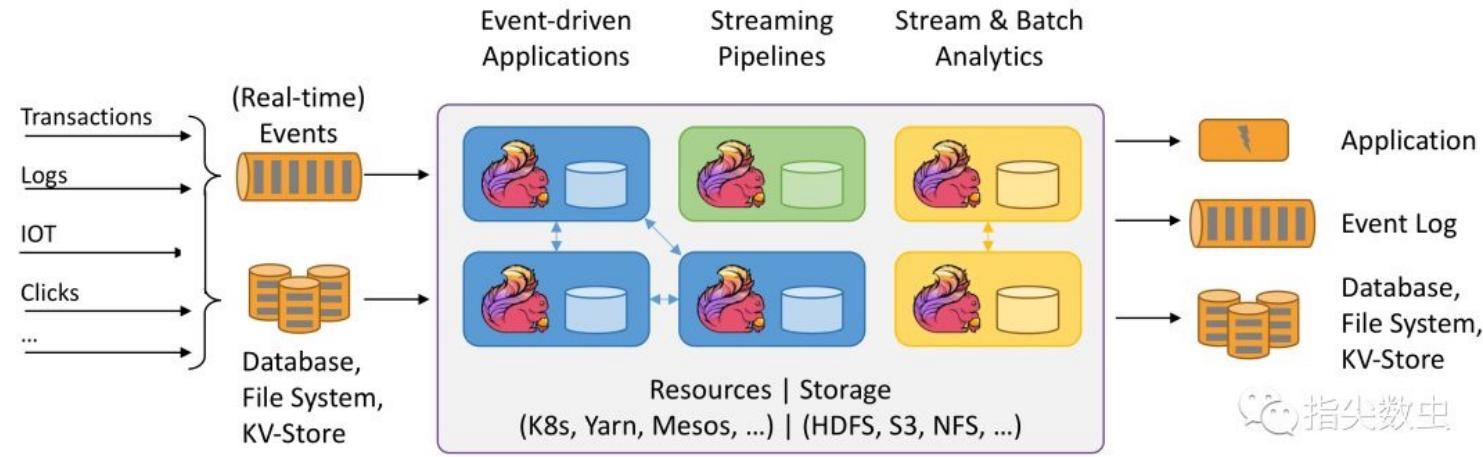
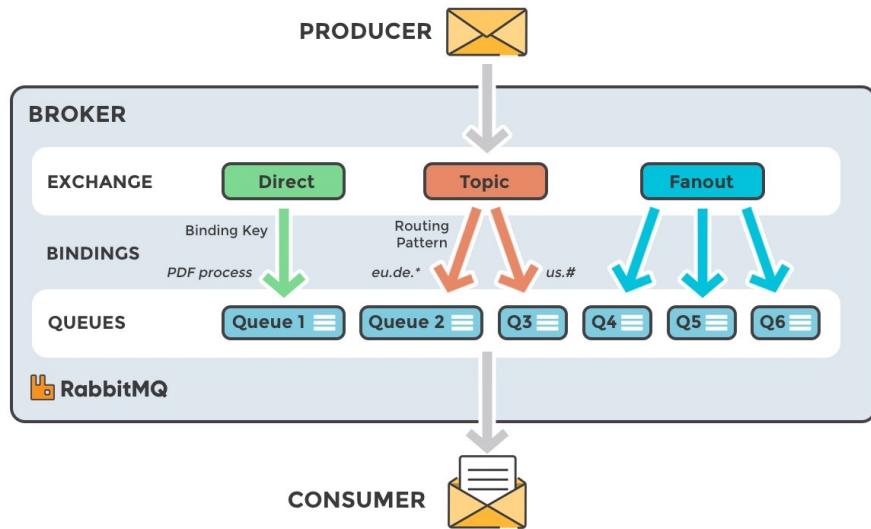
The 'Indexes' section shows a single index named 'PRIMARY' as BTREE, with 'id' as the column. It includes options for 'Edit', 'Rename', and 'Drop'.

The 'Console' section at the bottom contains a query history and a text input field for executing SQL queries. The current query shown is:

```
>CREATE TABLE `can302`.`user` ( `id` INT NOT NULL AUTO_INCREMENT , `first` VARCHAR(20) NOT NULL , `last` VARCHAR(20) NOT NUL...>
```

- Create a database can302
- Create a table user with 4 columns
- Set id as primary and auto-increasement

# More than database



- Redis cache
- MQ (message queue), somehow like an email system.
- Apache Flink, real-time to handle huge amount data flow.
- Search engine

<https://www.zhihu.com/question/54152397>

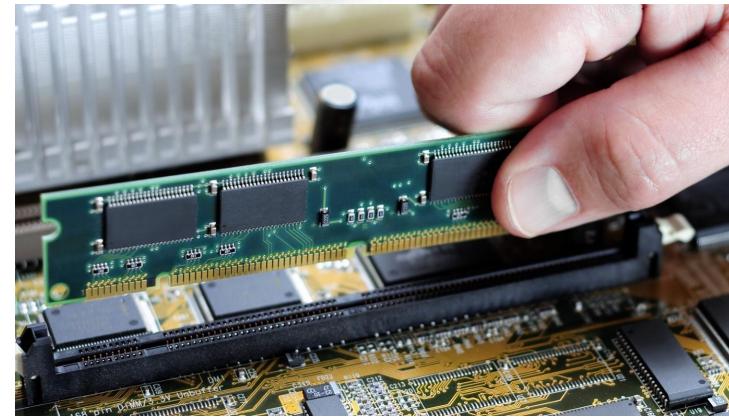
<https://zhuanlan.zhihu.com/p/428870096>

<https://www.cnblogs.com/lovezbs/p/14252254.html>

[https://blog.csdn.net/tiantang\\_1986/article/details/81204597](https://blog.csdn.net/tiantang_1986/article/details/81204597)

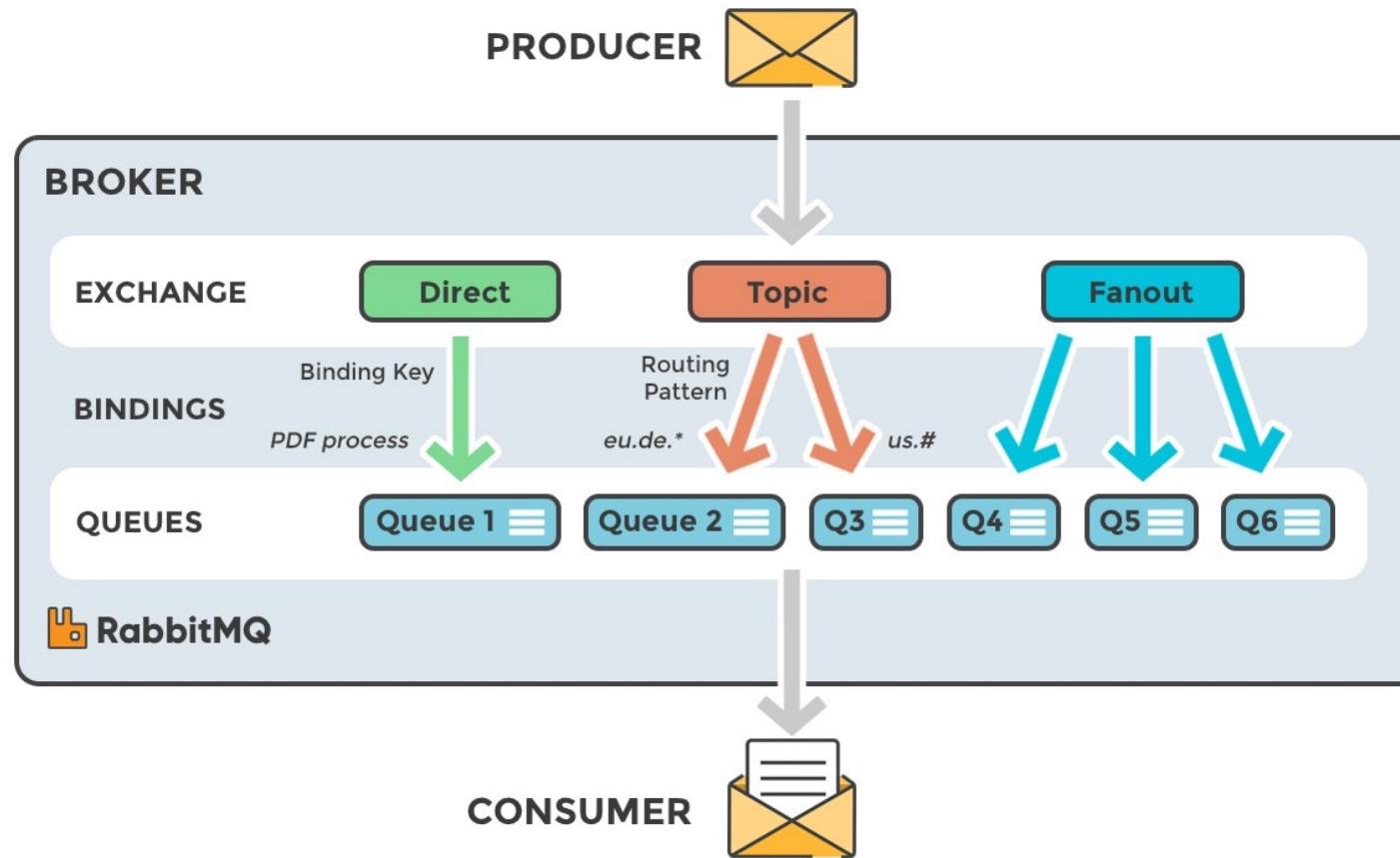


# Redis



- Redis is an open source (BSD licensed), **in-memory data structure store**, used as a database, cache, and message broker.

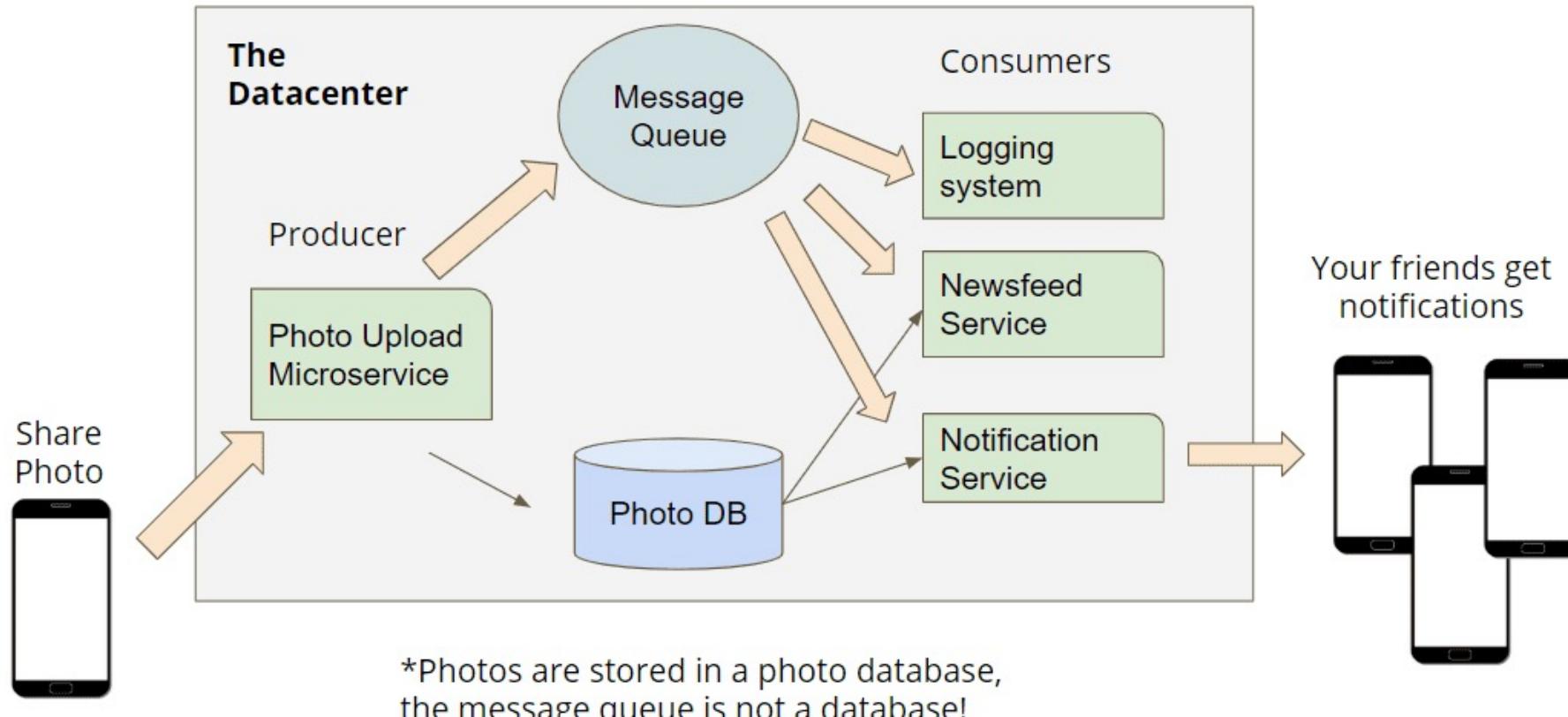
# MQ



- MQ like a mail system, decoupling the producer and consumer

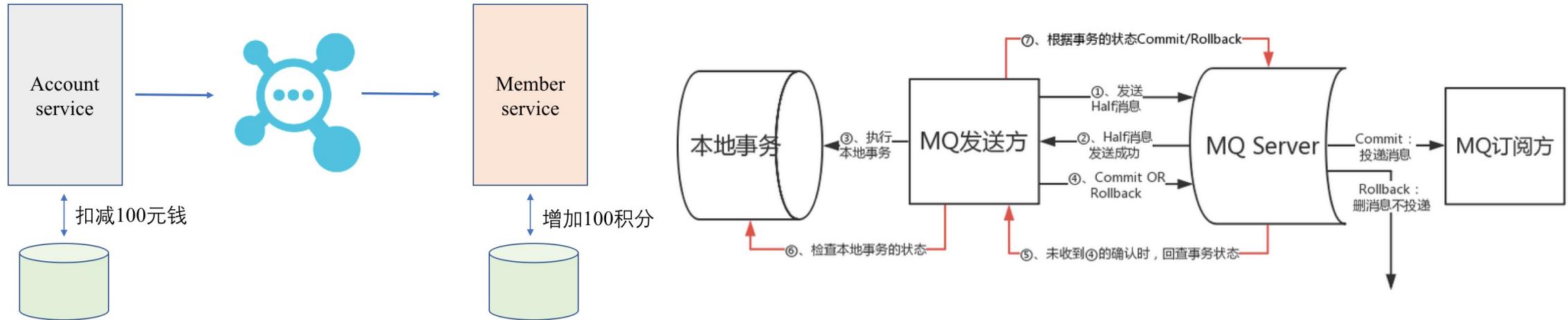
# MQ

## A less simple architectural diagram - A guess at how a large social media site handles photo uploads



- An application for notice

# Atomicity between systems



- Keep the account service status in MQ
- Update the member service from MQ

# Apache Flink



<https://www.jianshu.com/p/2a0188bcdab3>

# Index

## Index

friends_name_asc	
name	id
Andrew	3
Blake	5
Dave	2
Evan	6
Matt	1
Nick	7
Todd	4
Zack	8

points to

## Table

Friends		
id	name	city
1	Matt	San Francisco
2	Dave	Oakland
3	Andrew	Blacksburg
4	Todd	Chicago
5	Blake	Atlanta
6	Evan	Detroit
7	Nick	New York City
8	Zack	Seattle

- Improve the search efficiency

# How to index this one?

	<code>id</code>	<code>title</code>	<code>sell_point</code>
□	909245	酷派 8076D 咖啡棕 移动3G手机 双卡双待	4.0英寸屏幕，5
□	912107	创维(Skyworth)LED 42E5DH-R 42英寸 海量影视资源智能网络平板液晶电视二代	二代新品！！智
□	915676	联想 P780 极速版 深邃黑 联通3G手机 双卡双待	待机王，5吋HD
□	917460	华为 P6 (P6-C00) 黑 电信3G手机 双卡双待双通	经典旗舰！雅然
□	917461	华为 P6 (P6-C00) 白 电信3G手机 双卡双待双通	经典旗舰，万人
□	917770	华为 P6-C00 电信3G手机(粉色) CDMA2000/GSM 双模双待双通	情人节神器！粉

- MySQL cannot work on that

# Search engine

Key (term)	Value	Key (term)	Value
床	前	床	前
《静夜思》	床前明月光...	《春晓》	春眠不觉晓...
《水调歌头》	明月几时有？把酒问青天...	有	明月几时有...
		有	《水调歌头》

# Search engine

```
# ik 分词器，粗粒度分词
GET _analyze
{
  "analyzer": "ik_max_word",
  "text": "我爱北京天安门"
}
```

```
tokens : [
  {
    "token" : "我",
    "start_offset" : 0,
    "end_offset" : 1,
    "type" : "CN_CHAR",
    "position" : 0
  },
  {
    "token" : "爱",
    "start_offset" : 1,
    "end_offset" : 2,
    "type" : "CN_CHAR",
    "position" : 1
  },
  {
    "token" : "北京",
    "start_offset" : 2,
    "end_offset" : 4,
    "type" : "CN_WORD",
    "position" : 2
  },
  {
    "token" : "天安门",
    "start_offset" : 4,
    "end_offset" : 7,
    "type" : "CN_WORD",
    "position" : 3
  },
  {
    "token" : "天安",
    "start_offset" : 4,
    "end_offset" : 6,
    "type" : "CN_WORD",
    "position" : 4
  }
]
```



Elastic Search Analyzer :

- Standard Analyzer
- Simple Analyzer
- Stop Analyzer
- Whitespace Analyzer
- Keyword Analyzer
- Patter Analyzer
- Language : support 30+ languages

# A fragile demo

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>Test mysql</title>
7      <link rel="stylesheet" href="/m-dev-store/styles/bootstrap-337.min.css">
8      <link rel="stylesheet" href="/m-dev-store/font-awsome/css/font-awesome.min.css">
9      <link rel="stylesheet" href="/m-dev-store/styles/style.css">
10     <script src="/m-dev-store/js/jquery-331.min.js"></script>
11     <script src="/m-dev-store/js/bootstrap-337.min.js"></script>
12 </head>
13 <body>
14     <div class="container">
15         <h2> Database demo @ CAN302 </h2>
16         <p> A table to show info in table user</p>
17         <table class="table">
18             <thead>
19                 <tr>
20                     <th>id</th>
21                     <th>Firstname</th>
22                     <th>Lastname</th>
23                     <th>email</th>
24                 </tr>
25             </thead>
26             <tbody>
27                 <?php
28 /*connect to database */
29                 $con = mysqli_connect("localhost", "root", "", "can302");
30                 if (mysqli_connect_errno($con)) {
31                     die("连接 MySQL 失败: " . mysqli_connect_error());
32                 }
33                 mysqli_query($con,"set names utf8"); //数据库编码格式
34 //function to get post data safely.
35                 function mypost($str) {
36                     $val = !empty($_POST[$str]) ? $_POST[$str] : '';
37                     return $val;
38                 }
39 //get post data
40                 $last = mypost('last');
41                 $first = mypost('first');
42                 $email = mypost('email');
43                 $add = mypost('add');
44                 $search = mypost('search');
```

```
44                     $search = mypost('search');
45                     if ($add =='add') {
46                         $sql = "INSERT INTO `user` (`id`, `first`, `last`, `email`) VALUES (NULL, '".$first."', '".$last."', '".$email."')";
47                         echo $sql;
48                         $query = mysqli_query($con,$sql);
49                     }
50                     if ($search =='search'){
51                         $sql = "select * from user where first LIKE '%".$first."%' and last LIKE '%".$last."%' and email LIKE '%".$email."%'";
52                     } else {
53                         $sql = "select * from user";
54                     }
55                     echo $sql;
56                     $query = mysqli_query($con,$sql);
57                     while($row = mysqli_fetch_array($query)){
58                         echo "<tr>";
59                         echo "<td>".$row['id']."</td>";
60                         echo "<td>".$row['last']."'</td>";
61                         echo "<td>".$row['first']."'</td>";
62                         echo "<td>".$row['email']."'</td>";
63                         echo "</tr>";
64                     }
65                     mysqli_close($con);
66                 ?>
67             </tbody>
68         </table>
69     </div>
70     <br>
71     <div class="container">
72         <form class="form-inline" role="form" action="db-test.php" contentType:"application/x-www-form-urlencoded;charset=UTF-8"
73             method="post" >
74             <label class="form-control" for="first"> Firstname </label>
75             <input type="text" class="form-control" id="first" placeholder="Input first name" name="first">
76             <label class="form-control" for="last"> Lastname </label>
77             <input type="text" class="form-control" id="last" placeholder="Input last name" name="last">
78             <label class="form-control" for="email"> Email </label>
79             <input type="text" class="form-control" id="email" placeholder="Input email address" name="email">
80             <button type="submit" class="btn btn-primary" id="add" name="add" value="add"> Add </button>
81             <button type="submit" class="btn btn-primary" id="search" name="search" value="search"> Search </button>
82         </form>
83     </div>
84 </body>
85 </html>
```

# A fragile demo

The screenshot shows a web browser window titled "Test mysql". The address bar displays "localhost/m-dev-store/dbtest/db-test.php". The main content area is titled "Database demo @ CAN302" and contains the following text:  
A table to show info in table user  
select \* from user

<b>id</b>	<b>Firstname</b>	<b>Lastname</b>	<b>email</b>
1	302	can	can@sat.xjtlu.cn
2	1	1	1
3			
4	222	11	test@test.cn

Below the table are four input fields: "Firstname" (placeholder "Input first name"), "Lastname" (placeholder "Input last name"), "Email" (placeholder "Input email address"), and a search button. There are also "Add" and "Search" buttons.

# Discussion II



**How to make the codes robust?**