

Think
ThinkinLAMP.com

PHPCON
JUL11-12
SHANGHAI CHINA

2015 PHPCON SHANGHAI



MySQL 5.7下的PHP开发

@姜承尧

公众账号: InsideMySQL

www.innomysql.net

关于我——Inside君

• 工作经历

- | | | |
|----------------|------|-----------|
| • 2011 —— 至今 | 网易 | 数据库技术负责人 |
| • 2006 —— 2011 | 久游网 | 数据库工程部负责人 |
| • 2004 —— 2006 | 自主创业 | 技术负责人 |

• MySQL版本经历

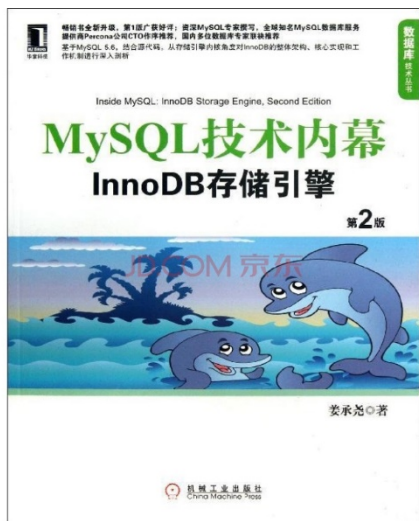
- 3.23 ~ 5.7.7
- 2003 ~ 2015（至今）
- 2013 Oracle MySQL ACE

• 我的MySQL分支版本

- InnoSQL
- 开源国产MySQL版本
- 欢迎加入InnoSQL社区
- www.innosql.net

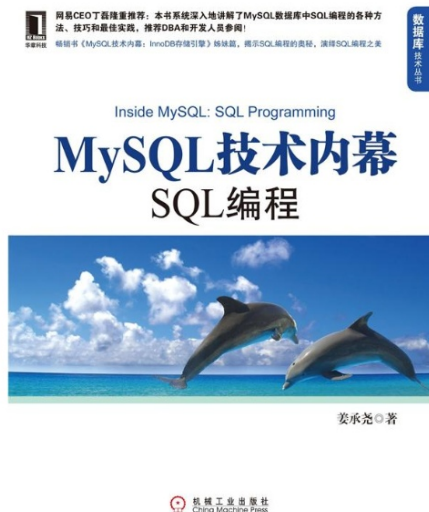


关于我——MySQL书籍



讲述如何用好InnoDB
适合人群：

- DBA
- 应用开发人员
- 应用架构师



讲述SQL高效开发
适合人群：

- DBA
- 应用开发人员



内核运行机制
适合人群：

- 资深DBA
- 内核开发人员

关于我——联系方式



@姜承尧



82946772



jiangchengyao
@gmail.com



InsideMySQL



waitoo.com/insidemysql

开发现状



非结构化
数据存储

全文索引
开发

地理位置
服务开发

开发现状——非结构化数据存储

- 结构化存储 => 关系型存储
 - 数据库

userid	cellphone	email	level	group
1024	13988888888	32@1024.com	1	junior
4096	18611111111	64@4096.com	10	senior
.....				

优点:

- 传统数据模型
- 易于用户理解

缺点:

- 数据模型僵化
- 模型不宜扩展
 - ALTER TABLE的代价

开发现状——非结构化数据存储

- 非结构化数据存储

- JSON
- BSON(Binary JSON)

优点:

➤ schema-free

缺点:

➤ schema-free

```
{
  _id: <ObjectId>,
  username: "123xyz",
  contact: {
    phone: "123-456-7890",
    email: "xyz@example.com"
  },
  access: {
    level: 5,
    group: "dev"
  }
}
```

Embedded sub-document

Embedded sub-document

开发现状——非结构化数据存储

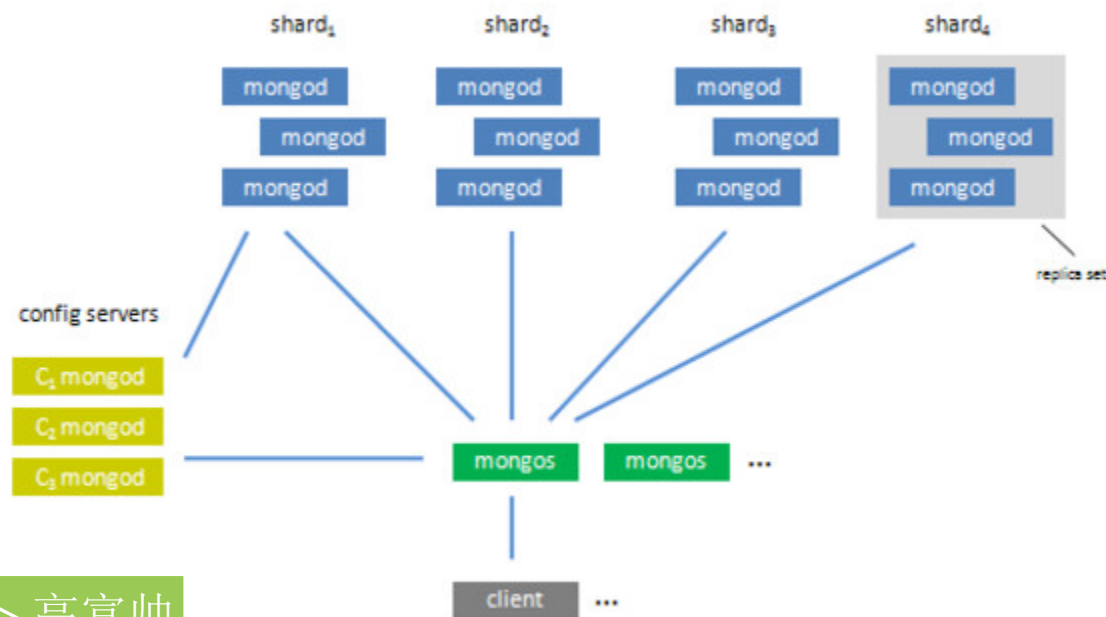
- 用户的选择



原生支持用户心心念念的sharding功能

单机 => 分布式

屌丝 => 高富帅



开发现状——全文索引开发

- 全文索引
 - 解决搜索like ‘%ABC%’问题
 - 搜索引擎核心技术之一



苍井空 番号 封面



百度一下

网页 新闻 贴吧 知道 音乐 图片 视频 地图 文库 更多»

百度为您找到相关结果约16,500,000个

搜索工具

[苍井空\(苍井そら\)作品番号-苍井空\(苍井そら\)种子ed2k封面内涵图片...](#)



2015年1月20日 - 首准速献[苍井空\(苍井そら\)](#)作品番号准确资料作品动态,昼夜持续关注更新[苍井空\(苍井そら\)](#)封面电影列表等相关信息供你观览,君之所好[苍井空\(苍井そら\)](#)内涵...

www.9bb.cc/jpavny/cang... - 百度快照 - 评价

[苍井空76部全集](#) [苍井空番号封面](#) [苍井空作品编号](#) [苍井空无...](#) [都市圈](#)

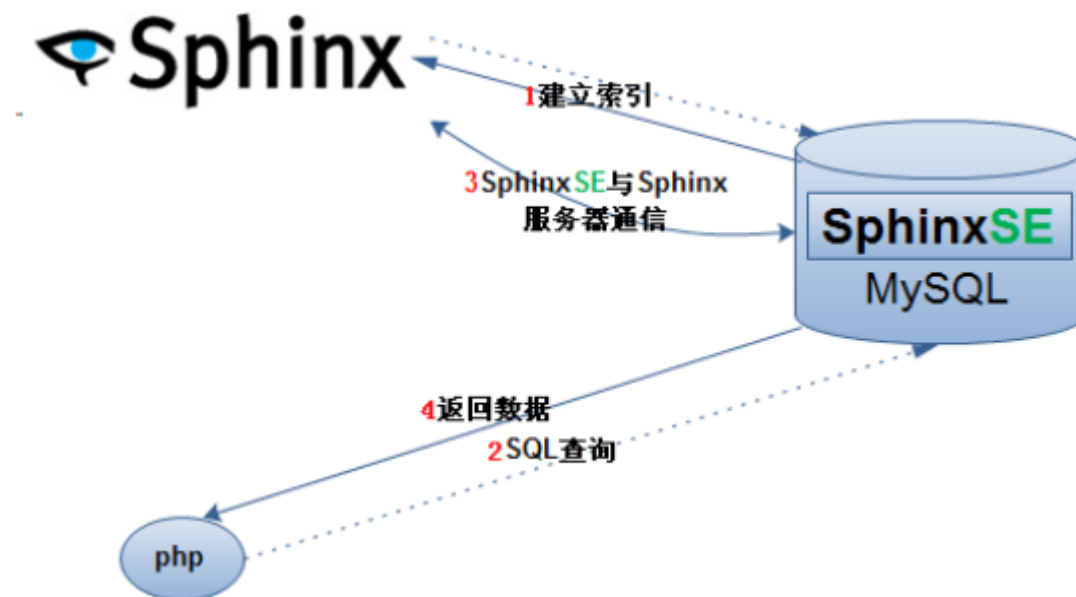


2015年3月6日 - 导语:[苍井空76部全集](#),[苍井空番号封面](#),[苍井空作品编号](#),[苍井空无乱码番号作品](#),下面小编为你揭秘。[苍井空76部全集](#) [苍井空番号封面](#) [苍井空作品编号](#) [苍井空无乱码...](#)

gz.o.cn/movie/news/149... - 百度快照 - 评价

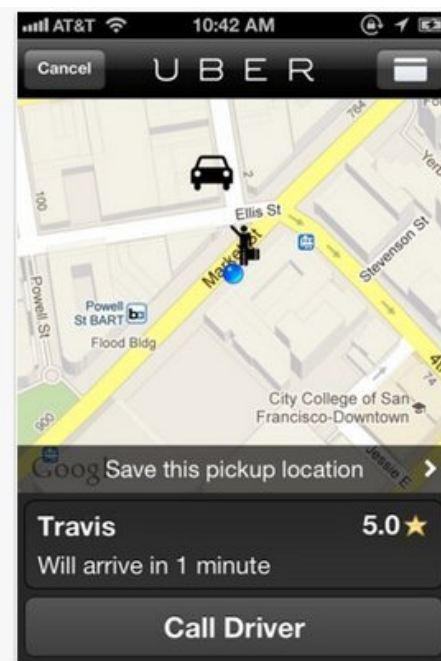
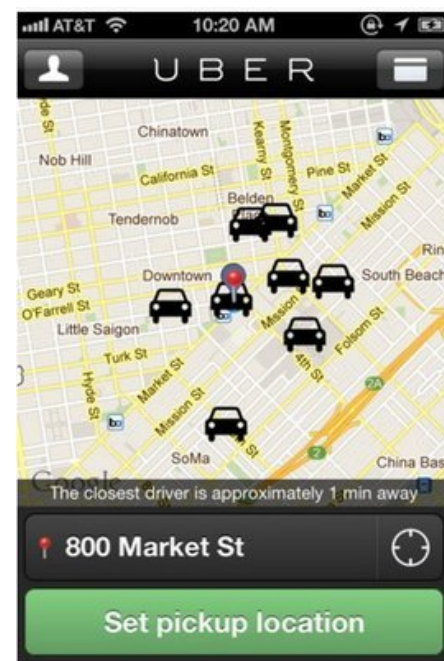
开发现状——全文索引开发

- 用户的选择
 - Lucence
 - Sphinx



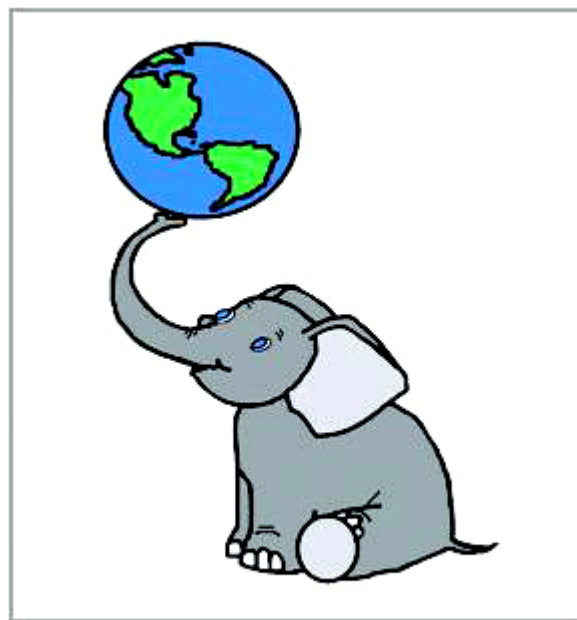
开发现状——基于地理位置服务开发

- LSB (Location-Based Service)
 - 陌陌——社交
 - 大众点评——O2O
 - Uber



开发现状——基于地理位置服务开发

- 用户的选择
 - PostGIS
 - MongoDB



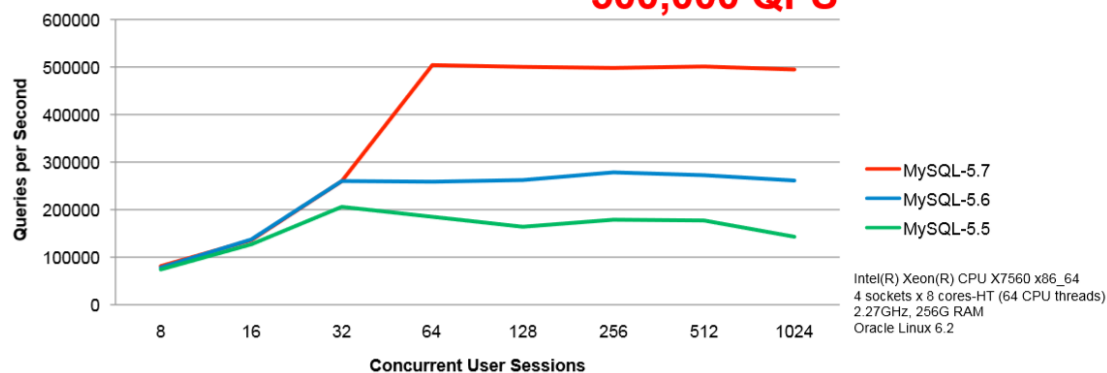
PostGIS/PostgreSQL
Spatial Database

MySQL 5.7——我想和这个世界谈谈

- MySQL 5.7.7 RC版本
 - Release Candidate

Sysbench Point Select

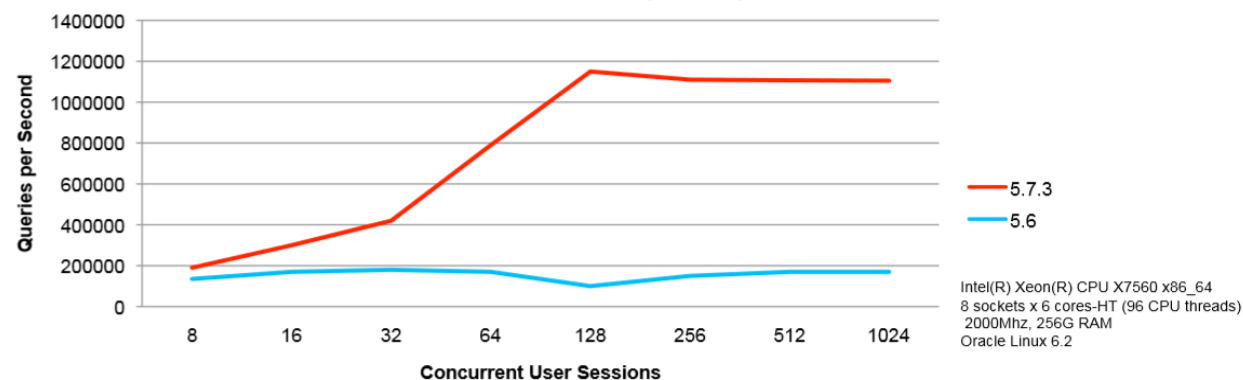
500,000 QPS



95% Faster than MySQL 5.6
172% Faster than MySQL 5.5

InnoDB Memcached

1,150,000 QPS



550% Faster than MySQL 5.6

MySQL 5.7——我想和这个世界谈谈

本次分享关注

```
graph TD; A[本次分享关注] --- B[原生JSON格式支持]; A --- C[虚拟列功能]; A --- D[更好GIS支持]; A --- E[全文索引中文支持];
```

原生JSON
格式支持

虚拟列功
能

更好GIS支
持

全文索引
中文支持

MySQL 5.7——原生JSON格式支持

JSON vs BLOB

- 数据有效性检查
- 查询性能提升
- 支持部分属性索引

```
mysql> create table user (  
-> uid int auto_increment,  
-> data json,  
-> primary key(uid))engine=innodb;
```

```
mysql> insert into user values (NULL,  
-> '{"name":"David","mail":"jiangchengyao@gmail.com","address":"Shangahai"}');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into user values (NULL,'{"name":"Amy","mail":"amy@gmail.com"}');
```

```
mysql> insert into user values (NULL,"test");  
ERROR 3130 (22032): Invalid JSON text: "Invalid value" at position 2 in value (or column) 'test'.
```


MySQL 5.7——原生JSON格式支持

```
mysql> select  
->  jsn_extract(data, '$.name'),  
->  jsn_extract(data, '$.address') from user;
```

jsn_extract(data, '\$.name')		jsn_extract(data, '\$.address')
"David"	"Shangahai"	
"Amy"	NULL	

JSON相关函数:

- ❑ json_append
- ❑ json_contains_key
- ❑ json_extrace
- ❑ json_merge
- ❑ json_remove
- ❑ json_replace
- ❑ json_search
- ❑ json_set
- ❑ json_test_parser
- ❑ json_valid

MySQL 5.7——虚拟列

- 虚拟列（Virtual Column）
 - 不占用任何存储空间
 - 可在虚拟列上创建索引

```
mysql> explain select * from user
-> where user_name="Amy"\G
***** 1. row *****
id: 1
select_type: SIMPLE
table: user
partitions: NULL
type: ref
possible_keys: idx_username
key: idx_username
key_len: 131
ref: const
rows: 1
filtered: 100.00
Extra: NULL
1 row in set, 1 warning (0.00 sec)
```

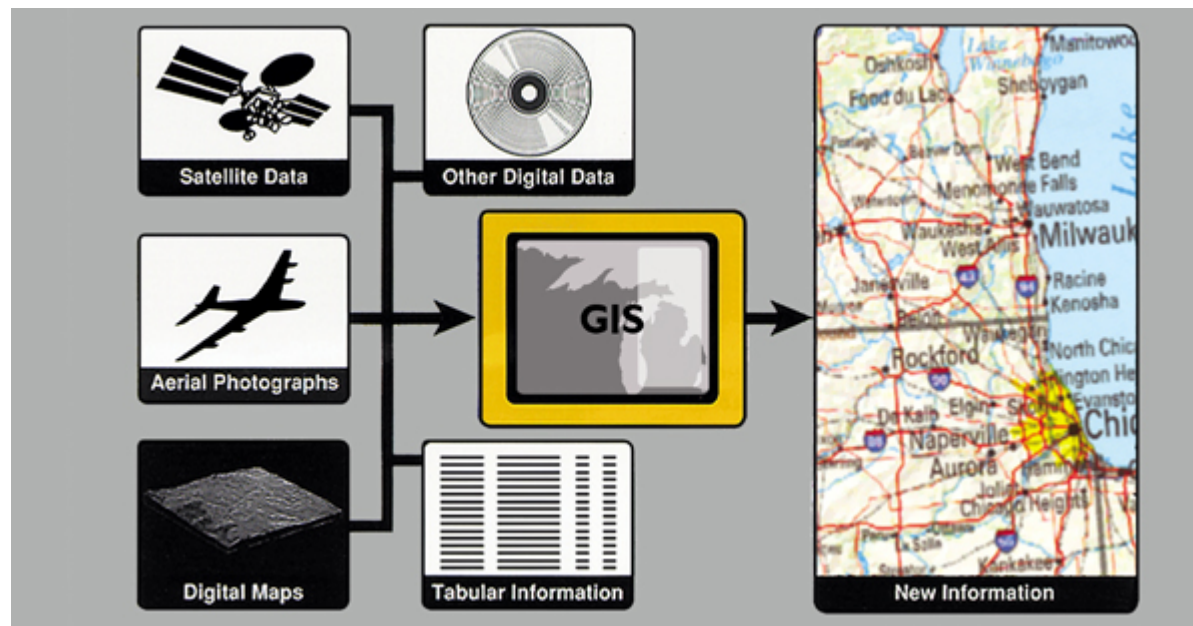
```
mysql> ALTER TABLE user ADD user_name varchar(128)
-> GENERATED ALWAYS AS (json_extract(data,'$.name')) VIRTUAL;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select user_name from user;
+-----+
| user_name |
+-----+
| "Amy" |
| "David" |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> alter table user add index idx_username (user_name);
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

MySQL 5.7——更好的GIS支持

- InnoDB原生GIS类型支持
- GeoHash索引功能支持
- 更多的GIS函数支持



MySQL 5.7——InnoDB原生GIS类型支持

- MySQL 5.5版本前
 - MyISAM表支持GIS类型
 - MyISAM表支持Spatial Index
- MySQL 5.6版本
 - InnoDB表支持GIS类型
 - 但是GIS类型以Blob类型存储
 - InnoDB表不支持支持Spatial Index
- MySQL 5.7版本
 - GIS类型 & Spatial Index支持

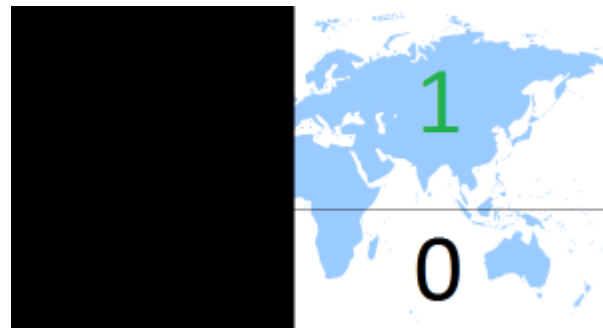
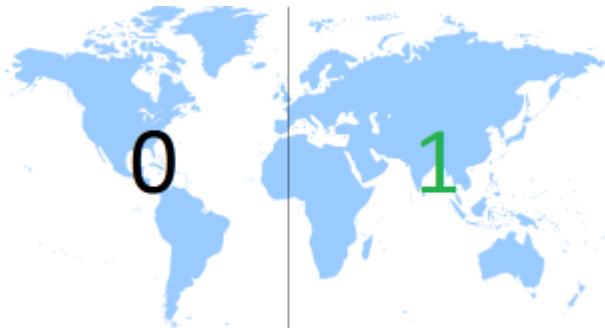


MySQL 5.7 —— GeoHash索引功能支持

- What is GeoHash
 - Geohash is a latitude/longitude geocode system
 - u5r2vty0

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 32	0	1	2	3	4	5	6	7	8	9	b	c	d	e	f	g
Decimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Base 32	h	j	k	m	n	p	q	r	s	t	u	v	w	x	y	z

u5r2vty0 = 11010 00101 10111 00010 11011 11001 11110 00000



MySQL 5.7 —— GeoHash索引功能支持

□ GeoHash 函数

- ST_GeoHash(longitude, latitude, max_length),
- ST_GeoHash(point, max_length)
- ST_LatFromGeoHash(geohash_str)
- ST_LongFromGeoHash(geohash_str)
- ST_PointFromGeoHash(geohash_str, srid)

```
mysql> SELECT ST_GeoHash(180,0,10);
```

```
+-----+  
| ST_GeoHash(180,0,10) |  
+-----+  
| xbpbbpbpb           |  
+-----+
```

举例：

Uber查找附近的车辆

但是不需要根据距离排序

MySQL 5.7 —— 更多空间函数支持



- GeoJSON
 - a format for encoding a variety of geographic data structures
 - makes it easy to connect MySQL with other GeoJSON enabled software and services
 - [Google Maps Javascript API](#)

```
{  
  "type": "Feature",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [125.6, 10.1]  
  },  
  "properties": {  
    "name": "Dinagat Islands"  
  }  
}
```

```
mysql> SELECT ST_AsText(ST_GeomFromGeoJson(  
-> '{  
>   "type": "Feature",  
>   "properties": {"Location": "Oracle HQ"},  
>   "geometry":  
>   {  
>     "type": "Point",  
>     "coordinates": [-122.262289, 37.530518]  
>   }  
> }'  
> ));
```

```
+-----+  
| SELECT ST_AsText(ST_GeomFr ... |  
+-----+  
| POINT(-122.262289 37.530518) |  
+-----+
```

MySQL 5.7 —— 更多空间函数支持

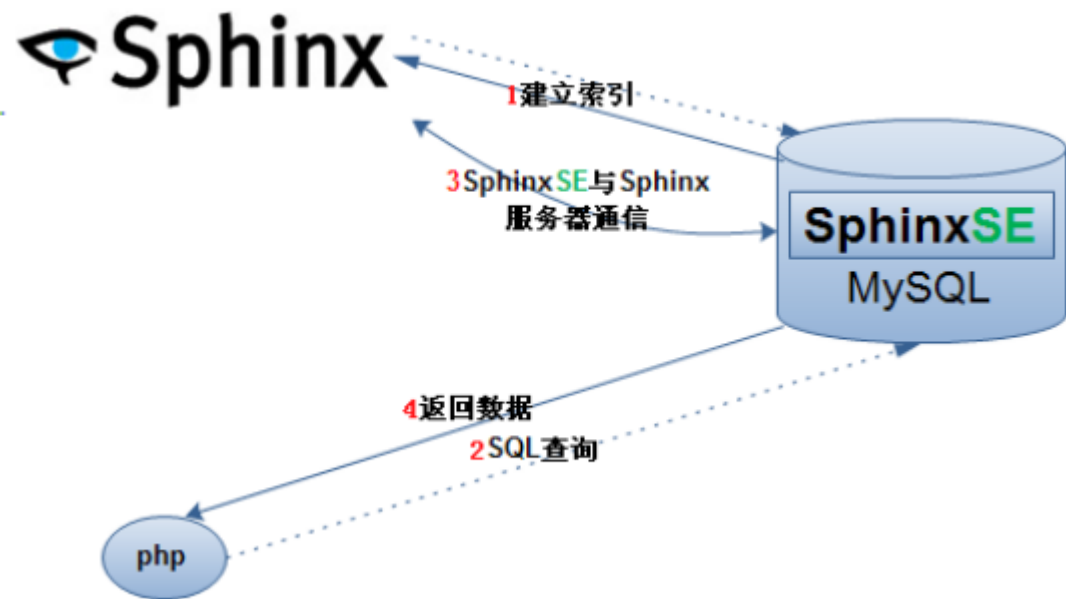
- ST_Distance_Sphere()
 - Returns the minimum spherical distance between two points and/or multipoints on a sphere, in meters
- ST_MakeEnvelope()
 - Returns the rectangle that forms the envelope around two points.

```
mysql> SELECT id, ST_Distance_Sphere(Point(30.18,120.19
), geom) as distance_in_meters, tags, ST_AsText(geom)
FROM nodes
WHERE ST_Contains( ST_MakeEnvelope(
    Point((30.18
+(20/111)), (120.19+(20/111))),
    Point((30.18,120.19
-(20/111)), (120.19-(20/111)))
), geom )
    AND match(tags) against (“+泰国 +粤菜+餐厅” IN BOOLEAN
MODE)
ORDER BY distance_in_meters LIMIT 10
```

可以先通过之前的GeoHash进行过滤，然后再计算距离

MySQL 5.7 —— 全文索引中文支持

- MySQL 5.5版本之前
 - 仅MyISAM表支持全文索引
 - 不支持中文分词
- MySQL 5.6版本
 - InnoDB表支持全文索引
 - 但不支持中文分词
- MySQL 5.7
 - 支持**无分隔符字符分词**
 - n-gram parser



MySQL 5.7 —— 全文索引中文支持

```
mysql> CREATE TABLE articles
(
  FTS_DOC_ID BIGINT UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
  title VARCHAR(100),
  FULLTEXT INDEX ngram_idx(title) WITH PARSER ngram
) Engine=InnoDB CHARACTER SET utf8mb4;
Query OK, 0 rows affected (0.06 sec)

mysql> # ALTER TABLE articles ADD FULLTEXT INDEX ngram_idx(title) WITH PARSER ngram;
```

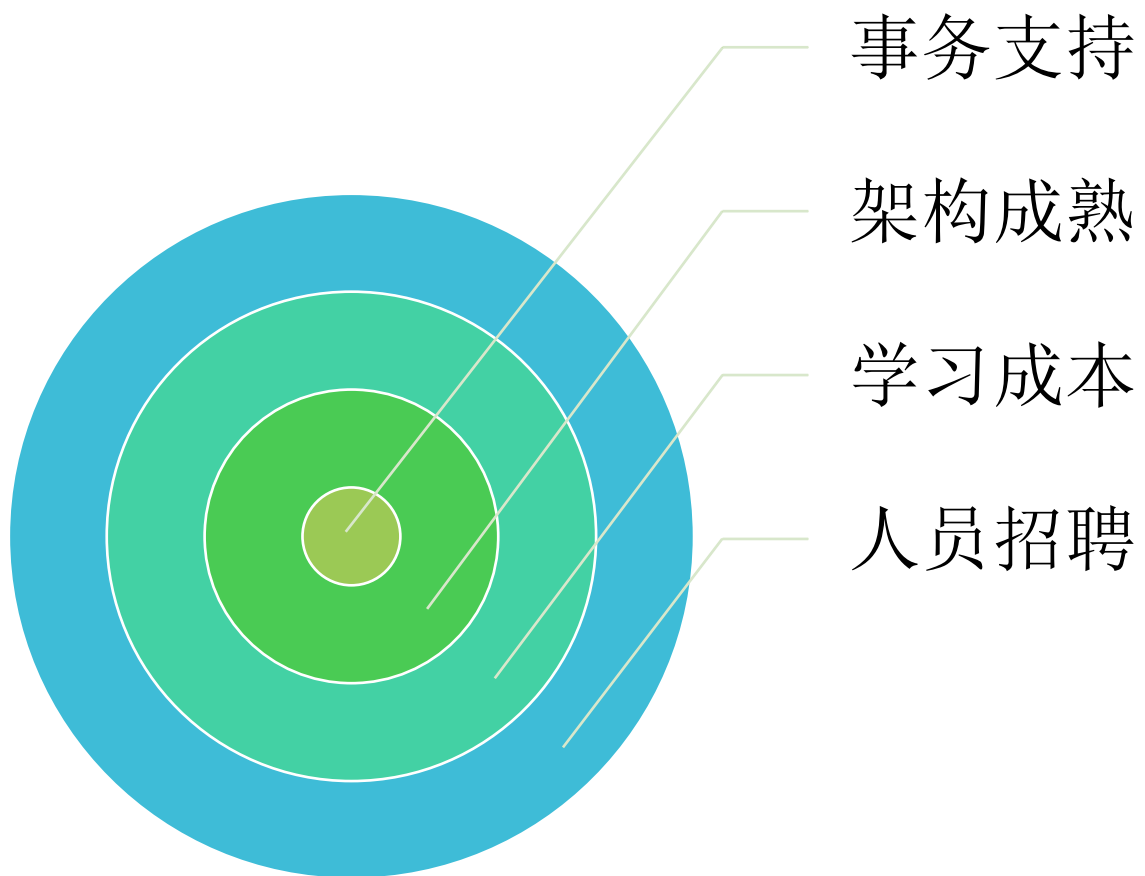
```
mysql> INSERT INTO articles (title) VALUES ('信息系统');
Query OK, 1 row affected (0.01 sec)

mysql> SET GLOBAL innodb_ft_aux_table="test/articles";
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM INFORMATION_SCHEMA.INNODB_FT_INDEX_CACHE;
+-----+-----+-----+-----+-----+-----+
| WORD  | FIRST_DOC_ID | LAST_DOC_ID | DOC_COUNT | DOC_ID | POSITION |
+-----+-----+-----+-----+-----+-----+
| 信息  | 1            | 1            | 1         | 1      | 0        |
| 息系  | 1            | 1            | 1         | 1      | 3        |
| 系统  | 1            | 1            | 1         | 1      | 6        |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM articles
-> WHERE MATCH (title) AGAINST ('信息*' IN BOOLEAN MODE);
+-----+-----+
| FTS_DOC_ID | title          |
+-----+-----+
| 1          | 信息系统        |
| 2          | 信息 系统      |
| 3          | 信息的系统    |
| 4          | 信息           |
+-----+-----+
4 rows in set (0.00 sec)
```

统一MySQL架构的优势



Q & A