What is the hottest website during the National Holiday?


中国铁路客户服务
www.12306.cn是中国铁路客户服务中心唯一网站。截止目前，没
2012年10月2日 星期二
首 页
客运服务
货运服务
站车风采
商务座位
1
2
3
4
动车组席位
更多>>>
最新动态
为保障您顺畅购票，请下载安装根证书。.cn网上售
关于铁路预售期有关事项公告
关于查询互联网购票电子支付等问题的公告 NEW!
铁路快运货物班列试办网上预订服务 NEW!
北京、广州、乌鲁木齐铁路局公告 NEW!
铁路旅客服务质量网上调查问卷
修改实名制车票退票办法
关于武广高铁提供电子客票直接进出站服务的公告
网上购票用户注册
购票/预约
退 票
余 票 查 询
全文搜索:
请输入搜索条件
铁路客运
法律法规及规范性文件
网上购票常见问题
铁路常识


2012年10月2日 星期三
首 页
客运
车票预订
车票预约
余票查询
列车时刻表查询
余票查询
日期: 2012 年 10 月 07 日
发站: 常州北
到站: 北京南
全部 始发 终到 始发终到 过路
全部 动车 直特 特快 快速 普快
余票信息每10分钟更新一次。
'--'代表无此席别，'0'代表票已售完。

余票查询　　提供信息为准。（www.12306.cn）

日期：2012 年 10 月 07 日

发站：常州北　到站：北京南　车次：

全部　始发　终到　始发终到　过路

全部　动车　直特　特快　快速　普快　普客　临客　验证码：

余票信息每10分钟更新一次。

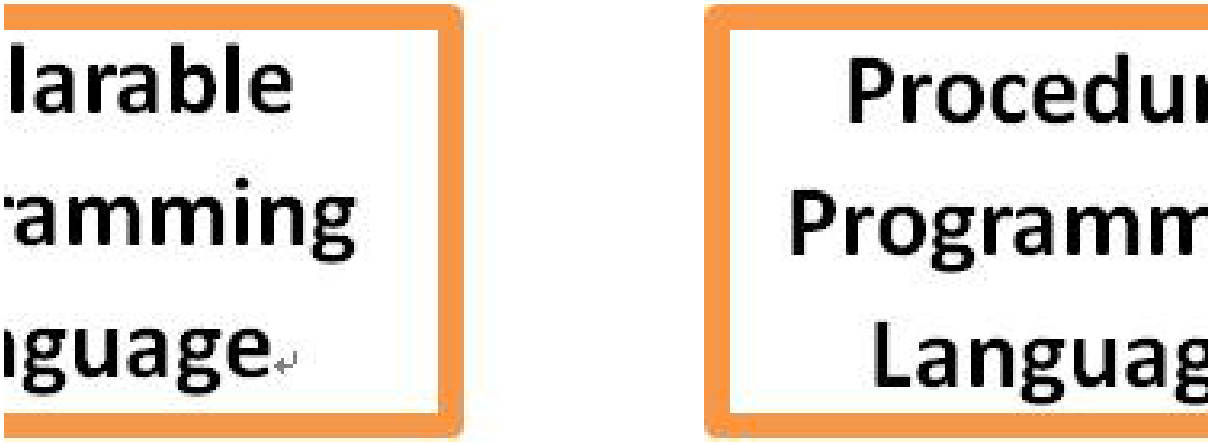
'--'代表无此席别，'0'代表票已售完。

日期：**20121007**；**常州北→北京南**列车全部余票信息15条：

序号	车次	查询区间		区间运行时刻				
		发站	到站	发时	到时	历时	商务座	特等座
1	G104(上海虹桥->北京南)	常州北	北京南	07:52	12:40	04:48	--	0
2	G110(上海虹桥->北京南)	常州北	北京南	08:54	13:28	04:34	0	--
3	D318(上海虹桥->北京南)	常州北	北京南	09:23	17:23	08:00	--	--
4	G34(杭州->北京南)	常州北	北京南	09:40	14:15	04:35	0	--
5	G36(杭州->北京南)	常州北	北京南	10:04	14:44	04:40	0	--
6	G120(上海虹桥->北京南)	常州北	北京南	11:04	15:42	04:38	0	--
7	G122(上海虹桥->北京南)	常州北	北京南	11:40	16:22	04:42	--	0
8	G128(上海虹桥->北京南)	常州北	北京南	12:28	17:03	04:35	--	0
9	G132(上海虹桥->北京南)	常州北	北京南	13:12	17:46	04:34	0	--
10	G136(上海虹桥->北京南)	常州北	北京南	13:37	18:25	04:48	0	--
11	G40(杭州->北京南)	常州北	北京南	13:54	18:36	04:42	0	--

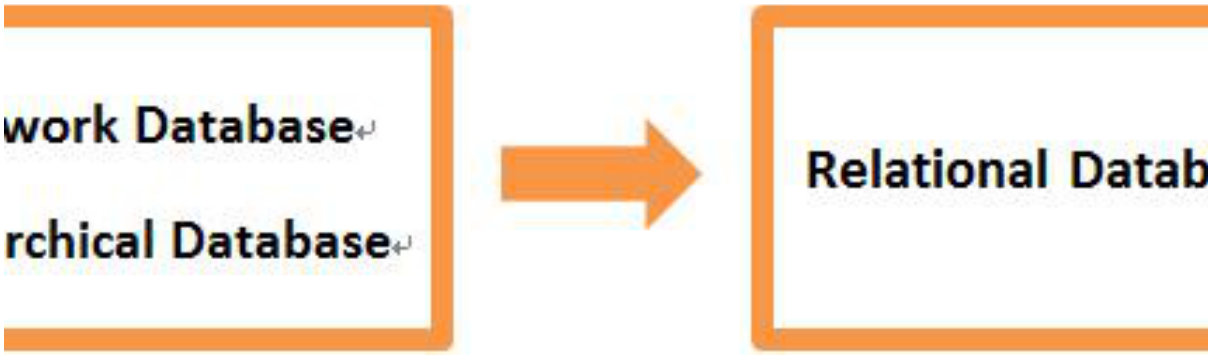
How Does It Work?

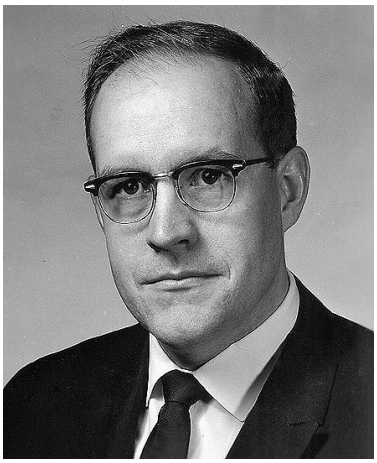
Dance with



SQL **C,**

C++, Java, ...



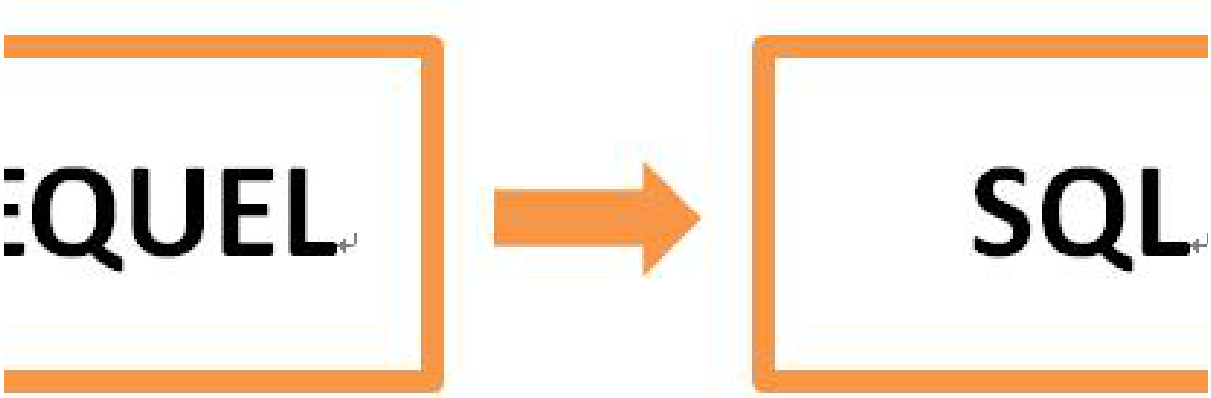


Dr. E. F. Codd

《A Relational Model of Data for Large Shared Data Banks》

Donald D. Chamberlin

Raymond F. Boyce



Structured English Query Language
Structured Query Language

Three Stages of Building Learning SNS

-Formulation Grammar
-More Users Optimization
-More and More Users Nosql

Structured Query Language
For Relational Database

-Data Definition
-Data Manipulation
-Data Query
-Data Control

Category	Verbs	Functions
DDL	CREATE, DROP	Create or drop table, view, index
DML	SELECT	Select from tables
DQL	INSERT, DELETE, UPDATE	Manipulation of data
DCL	GRANT, REVOKE	Give or revoke the authority to manipulate a ch

Grammar of SQL

-Create&Drop
-Insert, Delete, Update
-Select
-Grant&Revoke

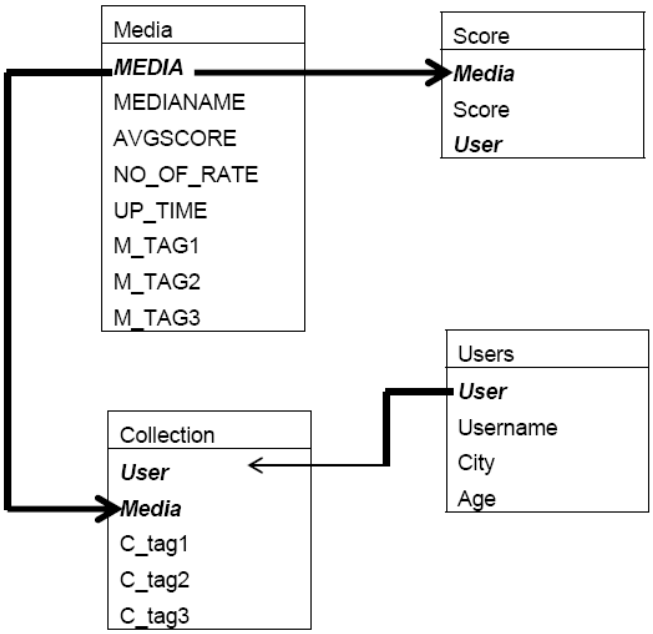
Create media table

```
SQLExecute[conn,
"CREATE  TABLE media (
  media INT NOT NULL ,
  medianame VARCHAR(45) NULL ,
  AVGscore FLOAT NULL ,
  up_time DATETIME NULL ,
  no_of_rate INT Null,
  M_tag1 VARCHAR(45) NULL ,
  M_tag2 VARCHAR(45) NULL ,
  M_tag3 VARCHAR(45) NULL ,
  PRIMARY KEY (media) )"]
```

```
Needs["DatabaseLink`"]
conn = OpenSQLConnection["demo"]
```

```
SQLConnection[demo, 1, Open, TransactionIsolationLevel → ReadCommitted]
```

Create table



TAG1,TAG2,TAG 3 ⟶ TAGS?

NO!! 1NF!!

AVGSCORE IN TABLE SCORE?

NO!! 2NF!!

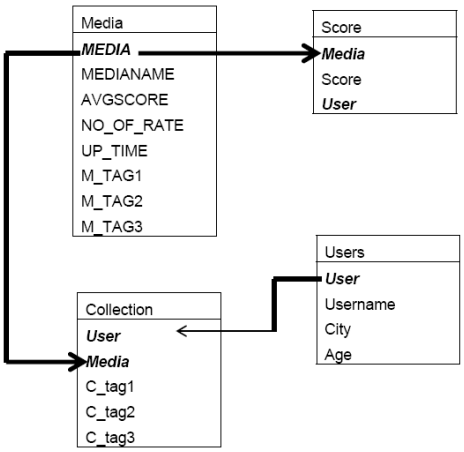
DROP

SQLExecute[conn, “DROP TABLE media “]

Gramma of SQL

-Create&Drop
-Insert, Delete, Update
-Select
-Grant&Revoke

Insert



SQLExecute[conn, “INSERT INTO user (user,username,city,age) VALUES (98,’Penny Liu’,’ShangHai’,’19’)”];
SQLExecute[conn, “INSERT INTO user (user,username,city,age) VALUES (3,’David Wu’,’BeiJing’,’21’)”];
SQLExecute[conn, “INSERT INTO user (user,username,city,age) VALUES (105,’John Green’,’New York’,’18’)”];
SQLExecute[conn, “INSERT INTO user (user,username,city,age) VALUES (78,’Amy Liang’,’Xi an’,’23’)”];
SQLExecute[conn, “INSERT INTO user (user,username,city,age) VALUES (893,’Suki Hsu’,’Hong Kong’,’17’)”]

```
SQLExecute[conn, "SELECT * FROM user"] // TableForm
```

```
3      David Wu      BeiJing      21
78     Amy Liang     Xi an        23
98     Penny Liu     ShangHai     19
105    John Green    New York     18
893    Suki Hsu      Hong Kong    17
```

UPDATE

```
1045   "relational model"          0.    0   SQLDateTime[{2012, 9, 23, 8, 0,
1057   "Oral Practicing"           0.    0   SQLDateTime[{2012, 8, 31, 21, 2
1347   "Pro/E design"              0.    0   SQLDateTime[{2012, 8, 23, 0, 53
1489   "Ergonomics 1st Class"      0.    0   SQLDateTime[{2012, 9, 17, 12, 0
1578   "Statics"                   0.    0   SQLDateTime[{2012, 9, 7, 6, 35,
1734   "World War 2"               0.    0   SQLDateTime[{2012, 8, 31, 9, 35
1984   "How to forecast weather"   0.    0   SQLDateTime[{2012, 9, 19, 19, 3
```

...

SQLExecute[conn, “insert into score(media,score,user)values (1984,6,893)”];

SQLExecute[conn, “update media set avgscore
=(avgscore*no_of_rate+6)/(no_of_rate+1),no_of_rate=no_of_rate+1
where media=1984”]

UPDATE

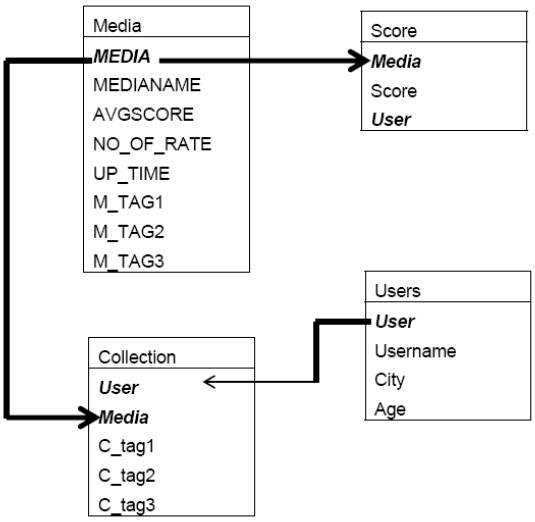
```
SQLExecute[ conn, "SELECT * FROM score"] // TableForm
```

```
1489    8.     3
1984    3.     3
1347    10.    78
1045    9.     98
1057    7.     98
1578    8.     105
1984    6.     105
1489    10.    111
1984    7.     111
1578    6.     893
1734    9.     893
1984    6.     893
```

```
SQLExecute[ conn, "SELECT * FROM media"] // TableForm
```

```
1045    relational model          9.     1    SQLDateTime[{2012, 9, 23, 8, 0, 0.}]      ele
1057    Oral Practicing           7.     1    SQLDateTime[{2012, 8, 31, 21, 23, 0.}]    hur
1347    Pro/E design              10.    1    SQLDateTime[{2012, 8, 23, 0, 53, 0.}]     eng
1489    Ergonomics 1st Class      9.     2    SQLDateTime[{2012, 9, 17, 12, 0, 0.}]     eng
1578    Statics                   7.     2    SQLDateTime[{2012, 9, 7, 6, 35, 0.}]      nat
1734    World War 2               9.     1    SQLDateTime[{2012, 8, 31, 9, 35, 0.}]     hur
1984    How to forecast weather   5.5    4    SQLDateTime[{2012, 9, 19, 19, 31, 0.}]    nat
```

DELETE



SQLExecute[conn, “delete from table media where **avgscore<6**”]

```
1045   "relational model"          9.    1   SQLDateTime[{2012, 9, 23, 8, 0, 0.
1057   "Oral Practicing"           7.    1   SQLDateTime[{2012, 8, 31, 21, 23,
1347   "Pro/E design"              10.   1   SQLDateTime[{2012, 8, 23, 0, 53, 0
1489   "Ergonomics 1st Class"      9.    2   SQLDateTime[{2012, 9, 17, 12, 0, 0
1578   "Statics"                   7.    2   SQLDateTime[{2012, 9, 7, 6, 35, 0.
1734   "World War 2"               9.    1   SQLDateTime[{2012, 8, 31, 9, 35, 0
1984   "How to forecast weather"   5.5   4   SQLDateTime[{2012, 9, 19, 19, 31,
```

```
1489    8.     3
1984    3.     3
1347    10.    78
1045    9.     98
1057    7.     98
1578    8.     105
1984    6.     105
1489    10.    111
1984    7.     111
1578    6.     893
1734    9.     893
1984    6.     893
```

Gramma of SQL

-Create&Drop
-Insert, Delete, Update
-Select
-Grant&Revoke

SELECT

Select <attribute1>, <attribute2>(if*,we will select all)
From <tablename>
Where <constraints>
Group by <attribute x>
Having <constraints>
Order by <attribute>

SELECT

Adding constriants in WHERE

Selection $Tour = \sigma_{Y/N=Y \wedge Age \geq 20}(IErs)$

```
3      David Wu      BeiJing      21
78     Amy Liang     Xi an        23
98     Penny Liu     ShangHai     19
105    John Green    New York     18
893    Suki Hsu      Hong Kong    17
```

-AND,OR

```
SQLExecute[ conn, "select user,username,city,age from user
where age > 20 and city = 'BeiJing'"] // TableForm
```

```
SQLExecute[ conn, "select user,username,city,age from user
where city = 'BeiJing' or city = 'ShangHai'"] // TableForm
```

Pay Attention!

```
SQLExecute[ conn, "select user,username,city,age from user
where city = 'BeiJing' or city = 'Xi an' and age > 22" ] // TableForm
```

```
3     David Wu     BeiJing    21
78    Amy Liang    Xi an      23
```

```
SQLExecute[ conn, "select user,username,city,age from user
where (city = 'BeiJing' or city = 'Xi an') and age > 22" ] // TableForm
```

```
78    Amy Liang    Xi an    23
```

SELECT

Adding constriants in WHERE(cont'd)

-NOT

```
SQLExecute[conn,
  "select user,username,city,age from user where not(city = 'BeiJing')"] // TableForm
```

```
78    Amy Liang    Xi an       23
98    Penny Liu    ShangHai    19
105   John Green   New York    18
893   Suki Hsu     Hong Kong   17
```

```
SQLExecute[conn,
  "select user,username,city,age from user where city <> 'BeiJing'"] // TableForm
```

```
78    Amy Liang    Xi an       23
98    Penny Liu    ShangHai    19
105   John Green   New York    18
893   Suki Hsu     Hong Kong   17
```

-Between,In,Like

BETWEEN value1 AND value2 ⟷ >= value1 AND <= value2

IN('Beijing','Shanghai') ⟷ ='Beijing'OR='Shanghai'

LIKE' Bei%' , LIKE'Bei_ _ _ _'

SELECT

Group By

```
3     1578   "Math"           "Markov"    Null
78    1347   "Spindle"        "Pro/e"     "CAM"
78    1489   "Illumination"   Null        Null
98    1045   "database"       "model"     Null
105   1057   "English"        "Test"      "TOEFL"
893   1347   "Car"            "Spindle"   Null
893   1734   "Japan"          "Germany"   "History"
```

```
SQLExecute[conn,
  "select user,count(media) from collection group by user"] // TableForm
```

```
3     1
78    2
98    1
105   1
893   2
```

```
SQLExecute[conn,
  "select user,count(media) from collection group by user having count(media)=1"] //
 TableForm
```

```
3     1
98    1
105   1
```

SELECT

Using data from several tables

Natural Join

$$[Detail+] = Detail \bowtie Duration$$

```
SQLExecute[ conn, "select c.user, u.username,c.media,medianame,C_tag1,C_tag2,C_tag3
from media m,user u,collection c
where m.media=c.media and u.user=c.user
order by user DESC"] // TableForm
```

893	Suki Hsu	1734	World War 2	Japan	Germany	History
893	Suki Hsu	1347	Pro/E design	Car	Spindle	Null
105	John Green	1057	Oral Practicing	English	Test	TOEFL
98	Penny Liu	1045	relational model	database	model	Null
78	Amy Liang	1489	Ergonomics 1st Class	Illumination	Null	Null
78	Amy Liang	1347	Pro/E design	Spindle	Pro/e	CAM
3	David Wu	1578	Statics	Math	Markov	Null

SELECT

Subquery select

```
SQLExecute[conn, "select media.medianame,media.media,media.avgscore
from media
where avgscore<
(select avg(avgscore) from media)
"] // TableForm
```

```
Oral Practicing           1057    7.
Statics                   1578    7.
How to forecast weather   1984    5.5
```

```
SQLExecute[conn, "select media.medianame,media.media,media.avgscore
from media
where avgscore<avg(avgscore)
"] // TableForm
```

```
JDBC::Error: No aggregate in this context.
from media
where avgscore < avg(avgscore)
]>>
$Failed
```

SELECT

Subquery select

-Using any, all

```
3     1578   "Math"           "Markov"    Null
78    1347   "Spindle"        "Pro/e"     "CAM"
78    1489   "Illumination"   Null        Null
98    1045   "database"       "model"     Null
105   1057   "English"        "Test"      "TOEFL"
893   1347   "Car"            "Spindle"   Null
893   1734   "Japan"          "Germany"   "History"
```

```
SQLExecute[conn, "select user
from collection
where collection.media=any(select media from collection where user=78)
and user<>78
"] // TableForm
```

```
893
```

Grammar of SQL

-Create&Drop
-Insert, Delete, Update
-Select
-Grant&Revoke

GRANT SELECT ON media to ADMIN1
Grant all on media to admin1 (all=select, insert,delete, update)

Testing on INDEX

. Effect of INDEX

0.1s⟶0.003s 30times

. Where needs optimization

We may only want to see few pages

. Consideration of other factors

Disk space, time spent in updating index

Example:

Our Learning SNS System is going to celebrate the tenth anniversary. Our department in Beijing wants to search the users who collect and score a same media material in last 180days, and thank them with rewards.
There are there tables

user:

userid ⌗	username	city	age

collection:

userid ⌗	username ⌗

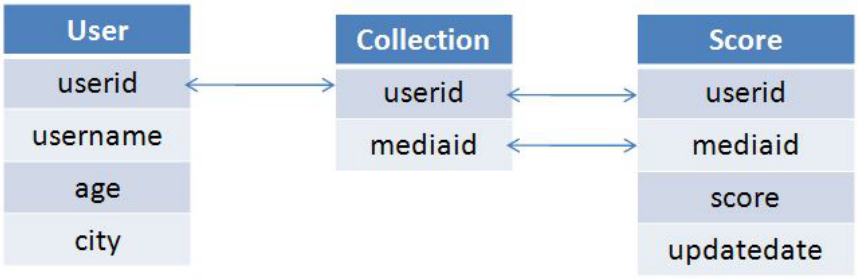
score:

userid ⌗	username ⌗	score	updatedate

How to find their '**usrname**' by using SQL statement.

Example:

- **Is it a solution?**




```
SELECT DISTINCT u.username
FROM user u
JOIN collection c
ON c.userid = u.userid
JOIN score s
ON s.mediaid = c.mediaid
AND s.userid = c.userid
WHERE u.city = 'Beijing'
AND s.updatedate >= CURRENT_Date-INTERVAL 180 DAY
```

Or without "DISTINCT" ?

Example:

- DISTINCT & No DISTINCT
- EXSITS & IN
- The order of conditions

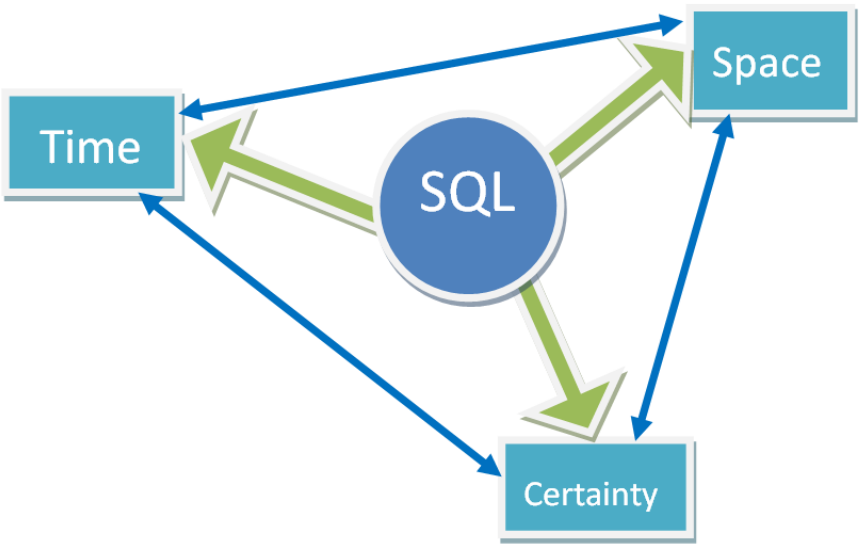
Example:

- **Using EXISTS statement**

```
SELECT u.username
FROM user u
WHERE EXISTS (SELECT *
FROM collection c, score s
WHERE c.userid = u.userid
AND s.mediaid = c.mediaid
AND s.userid = c.userid
AND s.updatedate >= CURRENT_Date - INTERVAL 180 DAY
AND u.city =  'Beijing')
```

It is a good statement?

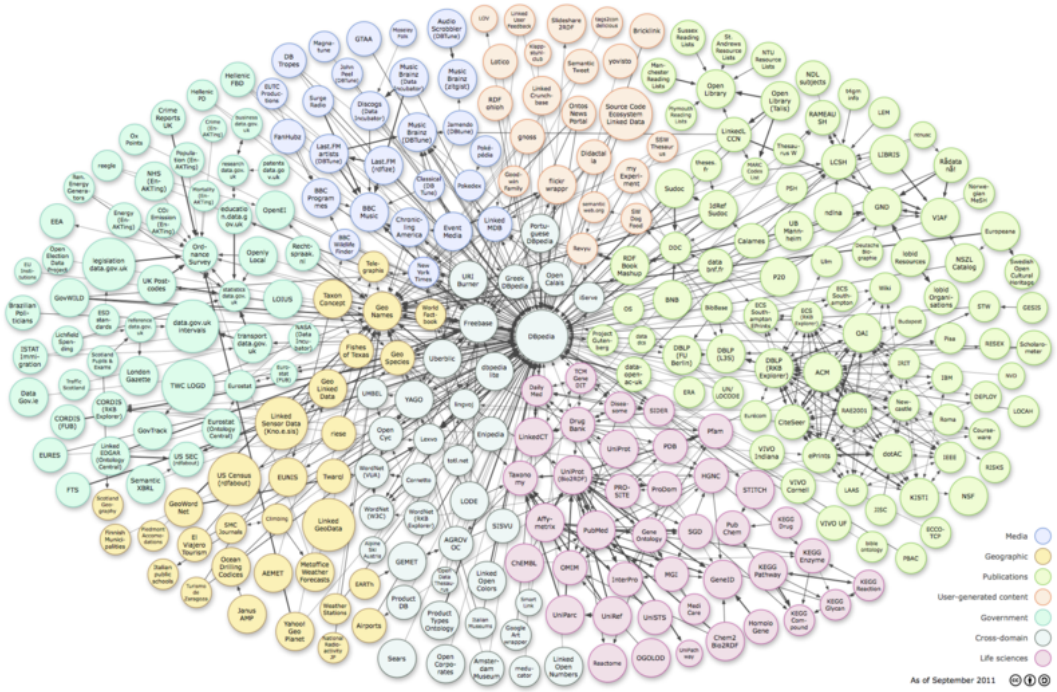
Relation between SPACE, TIME and CERTAINTY



SQL = magic language

But, can you sure this language perfect?

Even if the data become huge that you can't imagine?



NoSQL(Not only SQL)

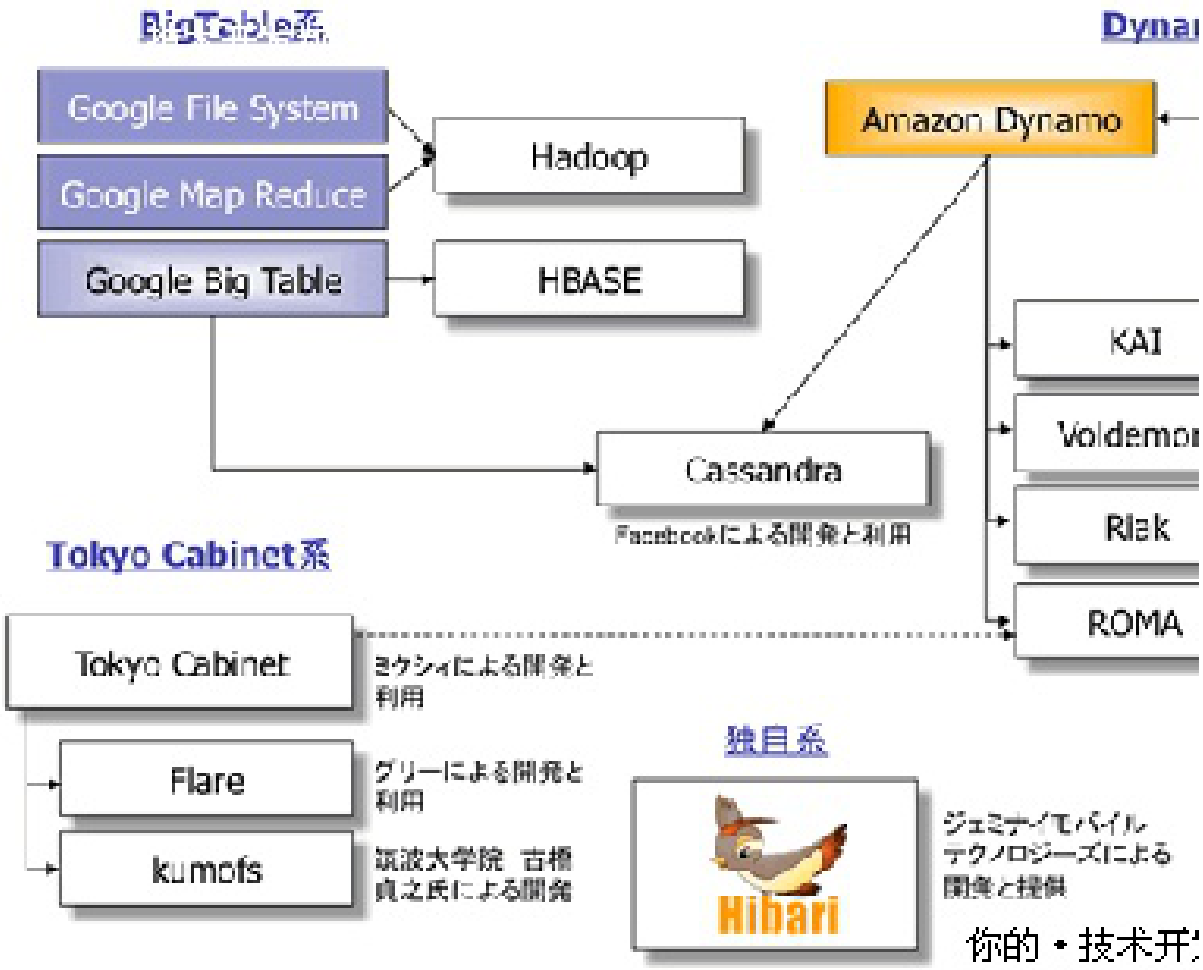
What's that mean???
A broad class of database management systems identified by non-adherence to the widely used relational database management system model (RDBMS).

Features?
It does not use SQL as its query language
It has a distributed, fault-tolerant architecture

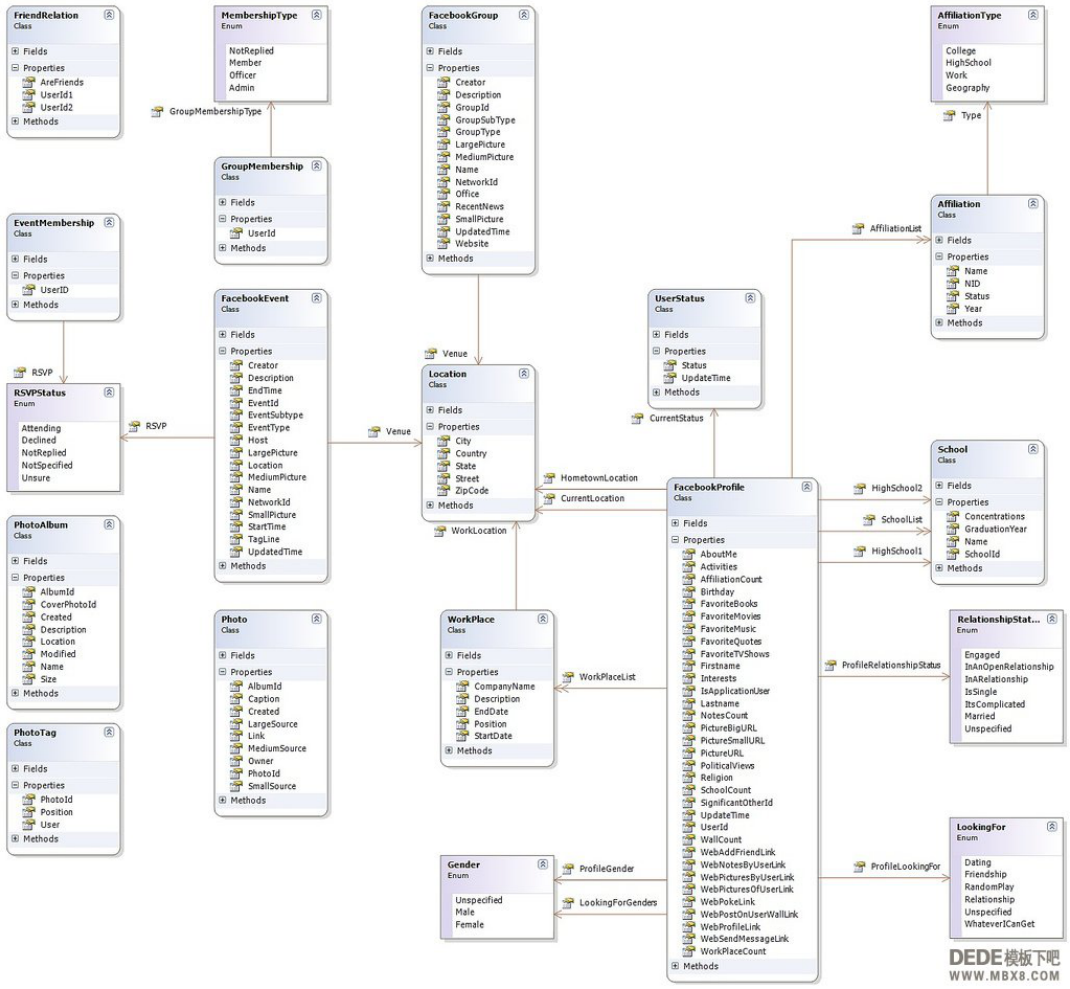
Why NoSQL?
High performance, huge storage, high Scalability and Availability

Representative example :

Slide


BigTable系
Google File System
Google Map Reduce
Google Big Table
Hadoop
HBASE
Amazon Dynamo
KAI
Cassandra
Facebookによる開発と利用
Riak
ROMA
Tokyo Cabinet系
Tokyo Cabinet
ミクシィによる開発と利用
Flare
グリーによる開発と利用
kumofs
筑波大学院 古橋貞之氏による開発
独自系
Hibari
ジェミナイモバイルテクノロジーズによる開発と提供

DDBMS (Distribute Database management system)



Popularity

1.	**Oracle**	**RDBMS**
2.	**Microsoft SQL Server**	**RDBMS**
3.	**MySQL**	**RDBMS**
4.	**Microsoft Access**	**RDBMS**
5.	**DB2**	**RDBMS**
6.	**PostgreSQL**	**RDBMS**
7.	**MongoDB**	**Document store**
8.	**SQLite**	**RDBMS**
9.	**Cassandra**	**Wide-column sto**
10.	**Memcached**	**Key-value store**
11.	Redis	Key-value store
12.	HBase	Wide-column store
13.	CouchDB	Document store
14.	Riak	Key-value store
15.	Neo4j	Graph database
16.	Berkeley DB	Key-value store
17.	MariaDB	RDBMS
18.	Oracle NoSQL	Key-value store

Summary

NoSQL is a ‘movement’ or ‘trend’!
It is not a unique database

If we want learn more about these,
Please learn SQL first!