

SQL

1970年6月，IBM的研究员Edgar Frank Codd在San Jose发表了《大型共享数据库中的数据的关系模型》（A Relational Model of Data for Large Shared Data Banks）的论文，提出了关系模型的概念，奠定了关系模型的理论基础。使用**行、列**组成的**二维表**来组织数据和关系，表中**行（记录）**即可以描述数据**实体**，也可以描述实体间**关系**。

SQL (Structured Query Language)：1976年，IBM实验室System R项目，通过实现数据结构和操作来证明关系模型实用性，并直接产生了**结构化查询语言SQL**。1987年，SQL被ISO组织标准化。它是一种对关系型数据库进行查询、更新、管理的可编程语言。

语言规范

- SQL语句大小写不敏感
 - 一般建议，SQL的关键字、函数等大写
- SQL语句末尾应该使用分号结束
- 注释
 - 多行注释 `/*注释内容*/`
 - 单行注释 `-- 注释内容`
 - MySQL 注释可以使用`#`
- 使用空格或缩进来提高可读性
- 命名规范
 - 必须以字母开头
 - 可以使用数字、#、\$和_
 - 不可使用关键字

MariaDB 安装

以CentOS安装为例

```
1 # yum list | grep mariadb
2 mariadb-libs.x86_64                1:5.5.60-1.el7_5 @anaconda
3 mariadb.x86_64                    1:5.5.60-1.el7_5 base
4 mariadb-bench.x86_64              1:5.5.60-1.el7_5 base
5 mariadb-devel.i686                1:5.5.60-1.el7_5 base
6 mariadb-devel.x86_64              1:5.5.60-1.el7_5 base
7 mariadb-embedded.i686             1:5.5.60-1.el7_5 base
8 mariadb-embedded.x86_64           1:5.5.60-1.el7_5 base
9 mariadb-embedded-devel.i686       1:5.5.60-1.el7_5 base
10 mariadb-embedded-devel.x86_64     1:5.5.60-1.el7_5 base
11 mariadb-libs.i686                 1:5.5.60-1.el7_5 base
12 mariadb-server.x86_64             1:5.5.60-1.el7_5 base
13 mariadb-test.x86_64               1:5.5.60-1.el7_5 base
14
15 安装mariadb 服务，会自动安装mairadb
16 # yum install mariadb-server
```

```

17
18 # systemctl start mariadb.service
19 # ss -tanl
20 State          Recv-Q Send-Q Local Address:Port          Peer Address:Port
21 LISTEN         0      50          *:3306                    *:
22
23 开机启动
24 # systemctl enable mariadb.service
25
26 为了安全设置MySQL服务
27 # mysql_secure_installation
28
29 数据库密码登录
30 # mysql -u root -p
31 mysql> show databases;
32 +-----+
33 | Database          |
34 +-----+
35 | information_schema |
36 | mysql              |
37 | performance_schema |
38 +-----+
39 3 rows in set (0.00 sec)
40
41 # 创建并授权用户
42 mysql> grant all on *.* to 'wayne'@'%' identified by 'wayne';
43 mysql> flush privileges;

```

导入测试脚本

```

1 # mysql -u root -p < test.sql

```

SQL语句

SQL语句分为

- DDL数据定义语言，负责数据库定义、数据库对象定义，由CREATE、ALTER与DROP三种语句组成
- DML数据操作语言，负责对数据库对象的操作，CRUD增删改查
- DCL数据控制语言，负责数据库权限访问控制，由GRANT和REVOKE两个指令组成
- TCL事务控制语言，负责处理ACID事务，支持commit、rollback指令

DCL

GRANT授权、REVOKE撤销

```

1 GRANT ALL ON employees.* TO 'wayne'@'%' IDENTIFIED by 'wayne';
2 REVOKE ALL ON *.* FROM wayne;

```

* 为通配符，指代任意库或者任意表。*. * 所有库的所有表；employees.* 表示employees库下所有的表 % 为通配符，它是SQL语句的通配符，匹配任意长度字符串

DDL

删除用户（慎用）

```
1 DROP USER wayne;
```

创建数据库 库是数据的集合，所有数据按照数据模型组织在数据库中。

```
1 CREATE DATABASE IF NOT EXISTS test CHARACTER SET utf8mb4 COLLATE
  utf8mb4_general_ci;
2 CREATE DATABASE IF NOT EXISTS test CHARACTER SET utf8;
```

CHARACTER SET指定字符集。utf8mb4是utf8的扩展，支持4字节utf8mb4，需要MySQL5.5.3+。COLLATE指定字符集的校对规则，用来做字符串的比較的。例如a、A谁大？

删除数据库

```
1 DROP DATABASE IF EXISTS test;
```

创建表

表分为行和列，MySQL是行存数据库。数据是一行行存的，列必须固定多少列。行Row，也称为记录Record，元组。列Column，也称为字段Field、属性。字段的取值范围叫做域Domain。例如gender字段的取值就是M或者F两个值。

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	1	1986-06-26
10002	1964-06-02	Bezalel	Simmel	2	1985-11-21
10003	1959-12-03	Parto	Bamford	1	1986-08-28
10004	1954-05-01	Chirstian	Koblick	1	1986-12-01
10005	1955-01-21	Kyoichi	Maliniak	1	1989-09-12
10006	1953-04-20	Anneke	Preusig	2	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	2	1989-02-10
10008	1958-02-19	Saniya	Kalloufi	1	1994-09-15
10009	1952-04-19	Sumant	Peac	2	1985-02-18
10010	1963-06-01	Duangkaew	Piveteau	2	1989-08-24

列、字段、Field、Column

```
1 CREATE TABLE `employees` (  
2   `emp_no` int(11) NOT NULL,  
3   `birth_date` date NOT NULL,  
4   `first_name` varchar(14) NOT NULL,  
5   `last_name` varchar(16) NOT NULL,  
6   `gender` smallint(6) NOT NULL DEFAULT '1' COMMENT 'M=1, F=2',  
7   `hire_date` date NOT NULL,  
8   PRIMARY KEY (`emp_no`)  
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

反引号标注的名称，会被认为是非关键字，使用反引号避免冲突。

删除表

```
1 DROP TABLE [IF EXISTS] `employees`;
```

DESC 查看列信息 {DESCRIBE | DESC} tbl_name [col_name | wild]

```
1 DESC employees;  
2 DESC employees '%name';
```

关系

在关系数据库中，关系就是二维表，由行和列组成。

行Row，也称为记录Record，元组。列Column，也称为字段Field、属性。字段的取值范围叫做域Domain。例如gender字段的取值就是1或者2两个值。

维数：关系的维数指关系中属性的个数 基数：元组的个数

注意在关系中，属性的顺序并不重要。理论上，元组顺序也不重要，但是由于元组顺序与存储相关，会影响查询效率。

候选键

关系中，能唯一标识一条元组的属性或属性集合，称为候选键。候选键，表中一列或者多列组成唯一的key，通过这一个或者多个列能唯一的标识一条记录。表中可能有多个候选键。

PRIMARY KEY主键

从候选键中选择出主键。

主键的列不能包含空值null。主键往往设置为整型、长整型，可以为自增AUTO_INCREMENT字段。表中可以没有主键，但是，一般表设计中，往往都会有主键，以避免记录重复。

Foreign KEY外键

严格来说，当一个关系中的某个属性或属性集合与另一个关系（也可以是自身）的候选键匹配时，就称作这个属性或属性集合是外键。

索引Index

可以看做是一本字典的目录，为了快速检索用的。空间换时间，显著提高查询效率。可以对一列或者多列字段设定索引。

主键索引，主键会自动建立主键索引，主键本身就是为了快速定位唯一记录的。唯一索引，表中的索引列组成的索引必须唯一，但可以为空，非空值必须唯一。普通索引，没有唯一性的要求，就是建了一个字典的目录而已。联合索引，多个字段组合创建索引，使用条件查询时，先匹配左边字段。

全文索引，MyISAM使用，对Char、Varchar、TEXT类型使用。空间索引，SPATIAL，基本不用。

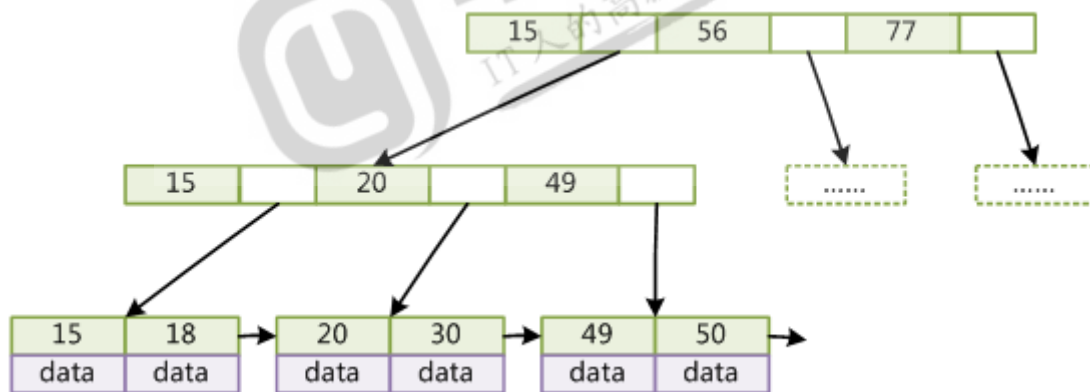
在MySQL中，InnoDB和MyISAM的索引数据结构可以使用Hash或BTree，innodb默认是BTree。

Hash时间复杂度是 $O(1)$ ，但是只能进行精确匹配，也就是Hash值的匹配，比如范围匹配就没办法了，hash值无序所以无法知道原有记录的顺序。Hash问题较多。

BTree索引，以B+树为存储结构。

虽然，索引可以提高查询效率，但是却影响增删改的效率，因为需要索引更新或重构。频繁出现在where子句中的列可以考虑使用索引。要避免把性别这种字段设索引。

B+树



B+树，是一个m叉树，也是一种平衡树，兼顾文件系统存储和操作的考虑，同时减少了树的深度。

B+树节点组织成一棵树。节点分为内部节点和叶子节点。

内部节点不存储数据，叶子节点不存储指针。

叶子节点深度一致。叶子节点包含所有索引字段值。

每个leaf node保存数据，所有的leaf node组织成**链表**。假设读取16到22的数据，找到18后（大于16的第一个数据是18），顺着链表往后遍历读取即可。

InnoDB中，数据文件本身就是按主键索引存储的，叶子节点中保存的就是数据记录。

如果在其他字段上定义B+Tree索引，这个索引树的叶子节点的数据记录的是主键，这种称为辅助索引。

InnoDB靠主键才能组织数据存储的，所以一定要定义主键，否则MySQL帮你找一个候选键作为主键，找不到会自动增加一个主键。这个主键建议使用整数或自增整数。整数比较方便快捷，自增减少B+树分裂再平衡。

InnoDB中主键索引和数据存在一起，称为**聚簇索引**、聚集索引、聚类索引、簇集索引。MyISAM索引和数据是不同文件存储，这是非聚集索引。聚簇索引效率一般高于非聚簇索引。

约束Constraint

为了保证数据的完整正确，数据模型还必须支持完整性约束。

“必须有值”约束 某些列的值必须有值，不许为空NULL。

域约束Domain Constraint

限定了表中字段的取值范围

实体完整性Entity Integrity

PRIMARY KEY约束定义了主键，就定义了**主键约束**。主键不重复且唯一，不能为空。UNIQUE KEY同理。

引用完整性Referential Integrity

也叫参照完整性。

外键定义中，可以不是引用另一张表的主键，但是，往往实际只会关注引用主键。

外键：在表B中的列，引用了表A中的主键，表B中的列就是外键。

A表称为主表（父表），B表称为从表（子表）。

插入规则：不需要指定。如果在表B插入一条记录，B的外键列插入了一个值，这个值必须是表A中存在的主键值。

更新规则：定义外键约束时指定该规则。

删除规则：定义外键约束时指定该规则。

外键约束的操作

设定值	说明
CASCADE	级联，从父表删除或更新会自动删除或更新子表中匹配的行
SET NULL	从父表删除或更新行，会设置子表中的外键列为NULL 必须保证子表列没有指定NOT NULL，也就是说子表的字段可以为NULL才行
RESTRICT（默认）	如果从父表删除主键，如果子表引用了，则拒绝对父表的删除或更新操作
NO ACTION	标准SQL的关键字，在MySQL中与RESTRICT相同。拒绝对父表的删除或更新操作

外键约束，是为了保证数据完整性、一致性，杜绝数据冗余和错误。

