

数据库

01

数据库概述

04

数据库查询

02

数据定义语言 (DDL)

05

查询操作符与子查询

03

数据操作语言 (DML)

06

函数

数据库概述

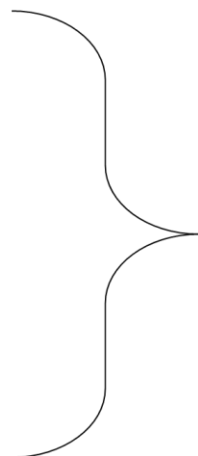
01

关系型数据库管理系统（RDBMS）与结构化查询语言（SQL）

数据库是存储、调用、分析数据的仓库，主要分为关系型数据库与非关系型数据库两类

- 关系数据库是数据库应用的主流
- 关系型数据库以行和列的形式存储数据，这一系列的行和列被称为表，一组表组成了数据库。
- 当前主流的关系型数据库有Oracle、DB2、Microsoft SQL Server、MySQL等。

操作关系型数据库时使用的语言叫结构化查询语言，简称SQL结构化查询语言。

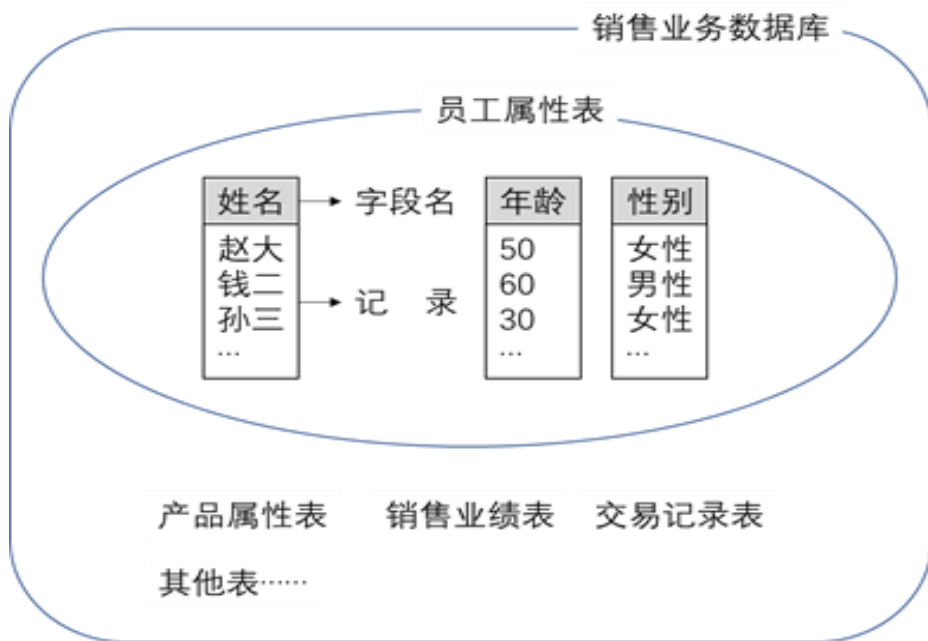


MySQL是最流行的关系型数据库管理系统之一

数据库：数据库是表的集合，带有相关的数据

- 表：一个表是多个字段的集合
- 字段：一个字段是一列数据，由字段名和记录组成

- 在商业数据分析中使用的绝大部分数据都来自于企业数据库
- 数据库负责数据收集、数据整合、数据调用等工作
- 几乎所有数据分析类工具都支持调用数据库中数据



数据定义语 言DDL

02

数据定义语言是用来对数据库管理系统中的对象进行“增删改查”操作的SQL语句，一般应用DDL语言对数据库不同对象进行定义操作时，需要提前具备相应数据库管理权限。

内 容：

1. 针对数据库对象的操作语言
2. 针对数据表对象的操作语言

操作关系型数据库

- 创建数据库：create database 数据库名称;
 - 例：创建名为test的测试数据库
create database test;
- 查看创建好的数据库：show create database 数据库名称;
 - 例：查看创建好的test数据库
show create database test;
- 查看所有数据库列表：show databases;
- 使用数据库：use 数据库名称;
 - 例：使用创建好的test数据库
use test;
- 删除数据库：drop database 数据库名称;
 - 例：删除创建好的test数据库
drop database test;



- 数据库是由多个数据表构成的
- 每张数据表存储多个字段
- 每个字段由不同的字段名及记录构成，每个字段有自己的数据结构及约束条件

商品颜色表： GoodsColor

字段名	字段描述	数据类型	主键	外键	非空	唯一	自增	初始值
ColorID	颜色ID	VARCHAR(4)	N	N	Y	N	N	-
ColorNote	颜色注释（对应商品主表的商品属性）	VARCHAR(20)	N	N	Y	N	N	-
ColorSort	颜色排序	INT	N	N	Y	N	N	0
pt	更新时间	VARCHAR(9)	N	N	Y	N	N	-

商品颜色表：存储商品的颜色信息

创建、查看及删除数据表

创建数据表：create table 表名(...);

– 例：用SQL语句创建以下员工信息表

部门ID	部门名称	员工数
p01	财务部	20
p02	销售部	100
p03	内审部	15
...

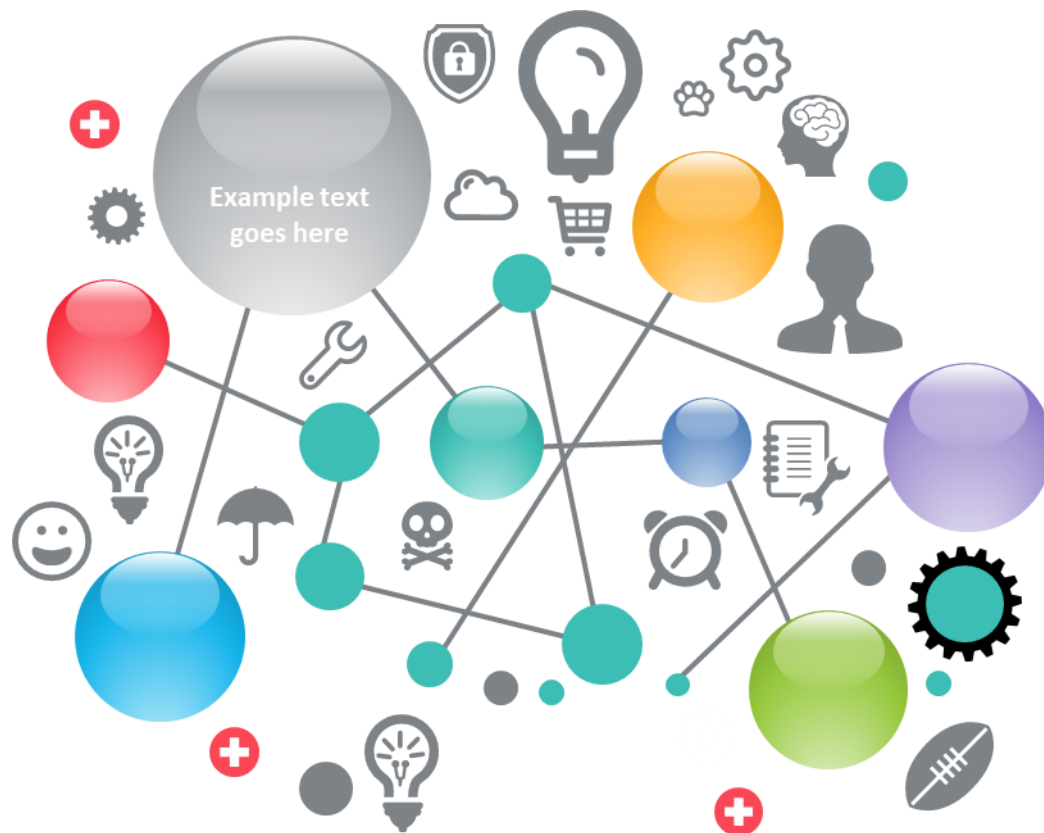
1. 使用test数据库: use test;

2. 创建员工信息表:

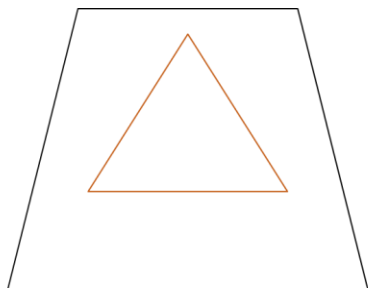
```
create table emp(  
    depid char(3),  
    depname varchar(20),  
    peoplecount int  
);
```

3. 查看表是否创建成功: show tables;

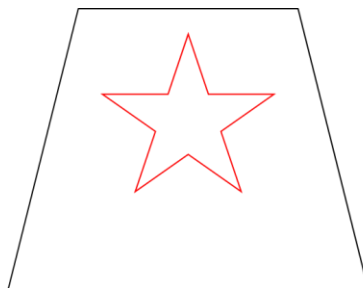
4. 删除数据表: drop table emp;



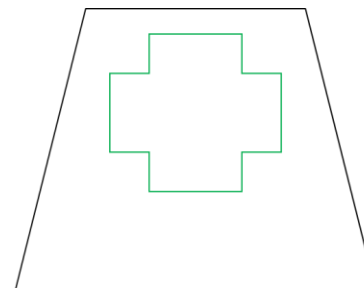
会员注册信息表



注册时间 日期型



用户名 文本型



充值金额 小数型

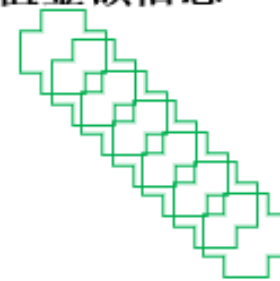
会员注册信息



用户名信息



充值金额信息



“位”与“字节”

- 位: "位(bit)"是电子计算机中最小的数据单位。每一位的状态只能是0或1。
- 字节: 8个二进制位构成1个"字节(Byte)", 它是存储空间的基本计量单位。
- 数据类型: 不同的数据类型具有不同的字节长度。字节长度越长能取值的数值区间也就越大。

一个INT整数型数据占用四个字节长度, 有符号大小-2147483648~2147483647, 无符号大小0~4294967295

1	1	0	0	1	0	1	1
1	0	1	1	1	0	1	1
1	0	1	0	0	0	0	1
1	0	1	0	1	0	1	0

- 数值类型：
 - INT: 有符号的和无符号的。有符号大小-2147483648~2147483647, 无符号大小0~4294967295。
最大显示宽度为11- int(11)
 - TINYINT:有符号的和无符号的。有符号大小-128~127, 无符号大小为0~255。
最大显示宽度为4- tinyint(4)
 - SMALLINT:有符号的和无符号的。有符号大小-32768~32767, 无符号大小为0~65535。
最大显示宽度为6- smallint(6)
 - MEDIUMINT:有符号的和无符号的。有符号大小-8388608~8388607, 无符号大小为0~16777215。
最大显示宽度为9- mediumint(9)
 - BIGINT:有符号的和无符号的。最大显示宽度为20- bigint(20)

注：在指定字段数据类型时如果在数据类型后加上UNSIGNED属性可以禁止负数

- 数值类型：
 - **FLOAT(M,D)**:只能为有符号的。4字节，默认为(10,2)
 - **DOUBLE(M,D)**:只能为有符号的。8字节，默认为(16,4)
 - **DECIMAL(M,D)**:只能为有符号的。

- 日期和时间类型

- DATE:YYYY-MM-DD格式, 在1000-01-01和9999-12-31之间。例如: 1973-12-30
- DATETIME:YYYY-MM-DD HH:MM:SS格式, 位于1000-01-01 00:00:00和9999-12-31 23:59:59之间。例如: 1973-12-30 15:30:00
- TIME: 以HH:MM:SS格式, -838:59:59~838:59:59
- TIMESTAMP:称为时间戳, 在1970-01-01 00:00:00和2037-12-31 23:59:59之间。例如, 1973年12月30日下午15:30, 则在数据库中存储为:19731230153000
- YEAR(2|4): 以2位或4位格式存储年份值。如果是2位, 1970~2069; 如果是4位, 1901~2155。默认长度为4

- 字符串类型

- **CHAR(M)**: 固定长度字符串，长度为1-255。如果内容小于指定长度，右边填充空格。如果不指定长度，默认为1
- **VARCHAR(M)**: 可变长度字符串，长度为1-255。定义该类型时必须指定长度
- BLOB 或 TEXT: 最大长度65535。存储二进制大数据，如图片。不能指定长度。两者区别：BLOB 大小写敏感
- TINYBLOB 和 TINYTEXT: 最大长度255。不能指定长度。
- MEDIUMBLOB 或 MEDIUMTEXT: 最大长度16777215 字符
- LONGBLOB 或 LONGTEXT: 最大长度4294967295 字符
- ENUM: 枚举。例如: ENUM('A' , ' B' , ' C')。NULL 值也可

- 约束是在表上强制执行的数据检验规则
- 用来保证创建的表的数据完整和正确
- MySQL数据库常用约束条件

约束条件	说明	语法
PRIMARY KEY	主键约束	字段名 数据类型 PRIMARY KEY
NOT NULL	非空约束	字段名 数据类型 NOT NULL
UNIQUE	唯一约束	字段名 数据类型 UNIQUE
AUTO_INCREMENT	自增字段	字段名 数据类型 AUTO_INCREMENT
DEFAULT	默认值	字段名 数据类型 DEFAULT 默认值

序号	学号	学生姓名	成绩
1	A001	赵大	50
2	A002	钱二	60
3	A001	赵大	
4	A003	张三	70
5		李四	80
6	A005	王五	90

空置

重复值

- 学号： 有重复值和空值，不符合主键、非空及唯一三项约束条件
- 学生姓名： 有重复值不符合主键、唯一两项约束条件
- 成绩： 有空值，不符合主键、非空两项约束条件
- 序号： 非空、唯一、自增，符合所有约束条件要求

主键约束：保证表中每行记录都不重复

主键，又称为”主码”，是数据表中一列或多列的组合。主键约束要求主键列的数据必须是唯一的，并且不允许为空。使用主键，能够惟一地标识表中的一条记录，还可以加快数据库查询的速度

主键分为两种类型：

-- 单字段主键：

```
create table emp(  
    depid char(3) primary key,  
    depname varchar(20),  
  
    );
```

-- 多字段联合主键：

```
create table emp(  
    depid char(3),  
    depname varchar(20),  
    peoplecount int,  
    primary key(depname,depid)
```

非空约束，指的是字段的值不能为空：

– 语法：字段名 字段类型 not null

```
create table emp(  
    depid char(3) primary key,  
    depname varchar(20) not null,  
    peoplecount int  
);
```

唯一性约束，要求该列的值必须是唯一的：

- 允许为空，但只能出现一个空值;
- 一个表中可以有多个字段声明为唯一的;
- 唯一约束确保数据表的一列或几列不出现重复值;
- 语法：字段名 数据类型 unique

```
create table emp(  
    depid char(3) primary key,  
    depname varchar(20) not null,  
    peoplecount int unique  
);
```

默认约束，指定某个字段的默认值：

- 当插入记录时如果没有明确为字段赋值时，那么系统就会自动为这个字段赋值为默认约束设定的值
- 语法： 字段名 数据类型 default 默认值

```
create table emp(  
    depid char(3) primary key,  
    depname varchar(20) default '-',  
    peoplecount int unique  
);
```

自增字段：一个表只能有一个自增字段，自增字段必须为主键的一部分。默认情况下从1开始自增

例： 创建含各种约束条件的数据表

```
CREATE TABLE example(id INT PRIMARY KEY AUTO_INCREMENT, -- 创建整数型自增主键
                      name VARCHAR(4) NOT NULL, -- 创建非空字符串字段
                      math INT DEFAULT 0, -- 创建默认值为0的整数型字段
                      minmax FLOAT UNIQUE -- 创建唯一约束小数型字段
);
```

练习： 尝试用语言描述以下建表语句

```
create table fruits(  
    f_id char(10) not null,  
    s_id int not null default 0,  
    f_name char(255) not null,  
    f_price decimal(8,2) not null,  
    primary key(f_id)  
);
```



修改数据表（1）

修改表指的是修改数据库中已经存在的数据表的结构：

- MySQL使用alter table语句修改数据表结构，包括：修改表名，修改字段数据类型或字段名，增加和删除字段，修改字段的排列位置等
- 例：将数据表emp改名为empdep
`alter table emp rename empdep;`
- 例：将数据表empdep中depname字段的数据类型由varchar(20)修改成varchar(30)
`alter table empdep modify depname varchar(30);`
- 例：将数据表empdep中depname字段的字段名改为dep
`alter table empdep change depname dep varchar(30);`
- 例：将数据表empdep中dep字段的字段名改回为depname，并将该字段数据类型该会为varchar(20)
`alter table empdep change dep depname varchar(20);`
- 例：为数据表empdep添加新字段maname，新字段数据类型为varchar(10)，约束条件为非空
`alter table empdep add maname varchar(10) not null;`

修改表指的是修改数据库中已经存在的数据表的结构：

- MySQL使用alter table语句修改数据表结构，包括： 修改表名，修改字段数据类型或字段名，增加和删除字段，修改字段的排列位置等
- 例：将数据表empdep中maname字段的排列顺序改为第一位
alter table empdep modify maname varchar(10) first;
- 例：将数据表empdep中maname字段的排列顺序改到depid字段之后
alter table empdep modify maname varchar(10) after depid;
- 例：删除maname字段
alter table empdep drop maname;

数据操作语言DML

03

数据操作语言（DML）是对表中记录进行添加（INSERT）、更新（UPDATE）、删除（DELETE）等操作的语言，与DDL语言相同，在使用DML语言对记录信息进行操作时，也需要提前拥有相应的操作权限

内 容：

1. 数据填充
2. 数据修改
3. 数据删除

填充数据 - 用insert into语句为表插入数据

语法：insert into 表名(字段1,字段2,...) values

-- 插入数据

```
insert into fruits(f_id,s_id,f_name,f_price)
values('a1',101,'apple',5.2),
('b1',101,'blackberry',10.2),
('bs1',102,'orange',11.2),
('bs2',105,'melon',8.2),
('t1',102,'banana',10.3),
('t2',102,'grape',5.3),
('o2',103,'coconut',9.2),
('c0',101,'cherry',3.2),
('a2',103,'apricot',25.2),
('l2',104,'lemon',6.4),
('b2',104,'berry',7.6),
('m1',106,'mango',15.6),
('m2',105,'xbabay',2.6),
('t4',107,'xbababa',3.6),
('b5',107,'xxxx',3.6);
```

练习：创建以下大气质量表

表名： Monthly_Indicator

字段名	字段描述	数据类型	主键	外键	非空	唯一	自增	初始值
city_name	城市名	VARCHAR(20)	Y	N	Y	N	N	-
month_key	月份	DATE	Y	N	Y	N	N	
aqi	AQI	INT(4)	N	N	Y	N	N	0
aqi_range	范围	VARCHAR(20)	N	N	Y	N	N	-
air_quality	质量等级	VARCHAR(20)	N	N	Y	N	N	-
pm25	PM2.5	FLOAT(6, 2)	N	N	Y	N	N	0
pm10	PM10	FLOAT(6, 2)	N	N	Y	N	N	0
so2	SO2	FLOAT(6, 2)	N	N	Y	N	N	0
co	CO	FLOAT(6, 2)	N	N	Y	N	N	0
no2	NO2	FLOAT(6, 2)	N	N	Y	N	N	0
o3	O3	FLOAT(6, 2)	N	N	Y	N	N	0
ranking	排名	INT(4)	N	N	Y	N	N	0

导入外部文本文件：

-- 为Monthly_Indicator表导入外部txt文件

```
load data local infile '文件路径.txt'  
into table Monthly_Indicator  
fields terminated by '\t'  
ignore 1 lines;
```



对导入表中的数据一般从导入内容、导入数据总行数以及表结构三方面进行检查

-- 检查导入内容Monthly_Indicator

```
Select * from Monthly_Indicator;
```

-- 检查导入数据总行数Monthly_Indicator

```
Select count(*) from Monthly_Indicator;
```

-- 检查表结构

```
Desc Monthly_Indicator;
```


为字段赋值update...set 与 删除记录 delete

update...set: 为字段赋值, 语法为update 表名 set 字段名 = 值;

例: 使用concat函数在f_name字段值前添加'fruit_' 信息

```
update fruits set f_name = concat('fruit_',f_name);
```

delete: 删除数据表中的数据, 语法为DELETE FROM 表名 [WHERE Clause], 如果省略where的话则删除表中所有数据记录

例: 删除f_id为'b5'的数据记录

```
delete from fruits where f_id = 'b5';
```