**Mysql**

**一、数据库的类型**

数据库的字段类型和java的数据类型差不多

**1、整型**

我们一般使用int就可以了，其他的类似于tinyint，就是取值范围不一样

|  |  |
| --- | --- |
| **MySQL数据类型** | **含义（有符号）** |
| tinyint(m) | 1个字节  范围(-128~127) |
| smallint(m) | 2个字节  范围(-32768~32767) |
| mediumint(m) | 3个字节  范围(-8388608~8388607) |
| int(m) | 4个字节  范围(-2147483648~2147483647) |
| bigint(m) | 8个字节  范围(+-9.22\*10的18次方) |

**2.浮点型：**

我们一般使用double

|  |  |
| --- | --- |
| **MySQL数据类型** | **含义** |
| float(m,d) | 单精度浮点型    8位精度(4字节)     m总个数，d小数位 |
| double(m,d) | 双精度浮点型    16位精度(8字节)    m总个数，d小数位 |

设一个字段定义为float(5,3)，如果插入一个数123.45678,实际数据库里存的是123.457，但总个数还以实际为准，即6位。

1. **字符串(char,varchar,\_text)**

|  |  |
| --- | --- |
| **MySQL数据类型** | **含义** |
| char(n) | 固定长度，最多255个字符 |
| varchar(n) | 固定长度，最多65535个字符 |
| tinytext | 可变长度，最多255个字符 |
| text | 可变长度，最多65535个字符 |
| mediumtext | 可变长度，最多2的24次方-1个字符 |
| longtext | 可变长度，最多2的32次方-1个字符 |

char和varchar：  
1.char(n) 若存入字符数小于n，则以空格补于其后，查询之时再将空格去掉。所以char类型存储的字符串末尾不能有空格，varchar不限于此。   
2.char(n) 固定长度，char(4)不管是存入几个字符，都将占用4个字节，varchar是存入的实际字符数+1个字节（n<=255）或2个字节(n>255)，所以varchar(4),存入3个字符将占用4个字节。   
3.char类型的字符串检索速度要比varchar类型的快。  
  
varchar和text：   
1.varchar可指定n，text不能指定，内部存储varchar是存入的实际字符数+1个字节（n<=255）或2个字节(n>255)，text是实际字符数+2个字节。   
2.text类型不能有默认值。   
3.varchar可直接创建索引，text创建索引要指定前多少个字符。varchar查询速度快于text,在都创建索引的情况下，text的索引似乎不起作用。

**4.二进制数据(\_Blob)**

这种格式主要是用来存储图片和视频，大家知道这种类型就可以了

1.\_BLOB和\_text存储方式不同，\_TEXT以文本方式存储，英文存储区分大小写，而\_Blob是以二进制方式存储，不分大小写。

2.\_BLOB存储的数据只能整体读出。

**5.日期时间类型**

|  |  |
| --- | --- |
| date | 日期 '2008-12-2' |
| time | 时间 '12:25:36' |
| datetime | 日期时间 '2008-12-2 22:06:44' |
| timestamp | 自动存储记录修改时间 |

若定义一个字段为timestamp，这个字段里的时间数据会随其他字段修改的时候自动刷新，所以这个数据类型的字段可以存放这条记录最后被修改的时间。

所以我们如果数据库里面定义的修改时间是timestamp就可以不用手动的插入用户修改时间，这样稍稍减少一些写代码的工作量，减少一些忘记录入修改时间的bug；

**6.数据类型的属性**

|  |  |
| --- | --- |
| **MySQL关键字** | **含义** |
| NULL | 数据列可包含NULL值 |
| NOT NULL | 数据列不允许包含NULL值 |
| DEFAULT | 默认值 |
| PRIMARY KEY | 主键 |
| AUTO\_INCREMENT | 自动递增，适用于整数类型 |

定义主键和外键主要是为了维护关系数据库的完整性，总结一下：  
主键是能确定一条记录的唯一标识，比如，一条记录包括身份正号，姓名，年龄。身份证号是唯一能确定你这个人的，其他都可能有重复，所以，身份证号是主键。   
外键用于与另一张表的关联。是能确定另一张表记录的字段，用于保持数据的一致性。比如，A表中的一个字段，是B表的主键，那他就可以是A表的外键。

**二、  主键、外键和索引的区别**

主键、外键和索引的区别？

|  |  |  |  |
| --- | --- | --- | --- |
|  | 主键 | 外键 | 索引 |
| 定义： | 唯一标识一条记录，不能有重复的，不允许为空 | 表的外键是另一表的主键, 外键可以有重复的, 可以是空值 | 该字段没有重复值，但可以有一个空值 |
| 作用： | 用来保证数据完整性 | 用来和其他表建立联系用的 | 是提高查询排序的速度 |
| 个数： | 主键只能有一个 | 一个表可以有多个外键 | 一个表可以有多个惟一索引 |

****. cascade方式****  
在父表上update/delete记录时，同步update/delete掉子表的匹配记录   
  
   **. set null方式**  
在父表上update/delete记录时，将子表上匹配记录的列设为null  
要注意子表的外键列不能为not null    
  
   **. No action方式**  
如果子表中有匹配的记录,则不允许对父表对应候选键进行update/delete操作    
  
   **. Restrict方式**  
同no action, 都是立即检查外键约束  
  
   **. Set default方式**

父表有变更时,子表将外键列设置成一个默认的值 但Innodb不能识别

1. **Sql语句部分**

**1.SQL语言的分类**

SQL语言共分为四大类：数据查询语言DQL，数据操纵语言DML，数据定义语言DDL，数据控制语言DCL。

1. 数据查询语言DQL（Data Query Language）

数据查询语言DQL基本结构是由SELECT子句，FROM子句，WHERE 子句组成：

SELECT <列名> FROM <表名> WHERE <查询条件>

2 .数据操纵语言DML（data manipulation language）

数据操纵语言DML主要有三种形式：

1) 插入：INSERT

2) 更新：UPDATE

3) 删除：DELETE

保存、备份数据库的方法---转储或载入。

mysql中的删除语句特别说明：

    如果想在mysql中给想要删除的表起别名，name必须在delete后面起，同时delete后面加上表的别名，

代表要删除这个表的内容，如果delete后面有两个表则代表要删除这两个表中的内容。

//错误

delete from person where p.id=7;

//delete删除，正确  
delete from person where id=7;

delete p from person p where p.id=7;  
delete p from person p LEFT JOIN course on p.courseId=course.courseId where p.id=7;

//只删除person表中的内容  
delete p from person p LEFT JOIN course c on p.courseId=c.courseId where p.id=7;

//同时删除person表和course表中的内容

delete p,c from person p LEFT JOIN course c on p.courseId=c.courseId where p.id=7;

3. 数据定义语言DDL（data definition/describe language）

数据定义语言DDL用来创建数据库中的各种对象-----表、视图、

索引等如：

1、最简单的：

CREATE TABLE t1(  
    id int not null,  
    name char(20)  
);  
2、带主键的：

a：  
CREATE TABLE t1(  
    id int not null primary key,  
    name char(20)  
);  
b：复合主键  
CREATE TABLE t1(  
    id int not null,  
    name char(20),  
    primary key (id,name)  
);  
3、带默认值的：

CREATE TABLE t1(  
    id int not null default 0 primary key,  
    name char(20) default '1'  
);

4. 数据控制语言DCL(Data Control Language)

数据控制语言DCL用来授予或回收访问数据库的某种特权，并控制

数据库操纵事务发生的时间及效果，对数据库实行监视等。如：

1. GRANT：授权。

操作cmd

show databases；

Use testjh；

Select \* from person；

小tips：

在数据库中写sql语句要注意“”和‘’的区别，“”代表起别名同时别名中可以输入空格，‘’代表字符串

接下来主要讲一下sql中的查询语句，也是我没最常用到的sql语句。

**条件查询介绍**

条件查询就是在查询时给出WHERE子句，在WHERE子句中可以使用如下运算符及关键字：

=、!=、<>、<、<=、>、>=；

IS NULL； IS NOT NULL

AND；OR；NOT；

IN(set)；Not in；

BETWEEN…AND；

Like 模糊查询：

\_ 任意一个字符

%：任意0~n个字符

去重distinct

数字可以相加

Ps：任何值与null相加都等于null

这个时候我们就要用到 ifnull

**sql中的聚合函数  sum avg max min count**

COUNT()：统计指定列不为NULL的记录行数；

MAX()：计算指定列的最大值，如果指定列是字符串类型，那么使用字符串排序运算；

MIN()：计算指定列的最小值，如果指定列是字符串类型，那么使用字符串排序运算；

SUM()：计算指定列的数值和，如果指定列类型不是数值类型，那么计算结果为0；

AVG()：计算指定列的平均值，如果指定列类型不是数值类型，那么计算结果为0；

**分组：**

 注：凡和聚合函数同时出现的列名，一定要写在group by 之后

这是规矩。

Having：

注：having与where的区别:

1.having是在分组后对数据进行过滤.

where是在分组前对数据进行过滤

2.having后面可以使用聚合函数(统计函数)

where后面不可以使用聚合函数。

WHERE是对分组前记录的条件，如果某行记录没有满足WHERE子句的条件，那么这行记录不会参加分组；而HAVING是对分组后数据的约束。

Limit x,y：

查询从第x行开始，不包括x行，往后加y行的内容

8.3　查询代码的书写顺序和执行顺序

       查询语句书写顺序：select – from- where- group by- having- order by-limit

       查询语句执行顺序：from - where -group by - having - select - order by-limit

**多表查询，内连接，外链接**

内连接inner join: 只连接匹配的行  
左外连接 left join': 包含左边表的全部行（不管右边的表中是否存在与它们匹配的行），以及右边表中全部匹配的行  
右外连接 right join: 包含右边表的全部行（不管左边的表中是否存在与它们匹配的行），以及左边表中全部匹配的行

Ps：mysql不支持  
全外连接 full join: 包含左、右两个表的全部行，不管另外一边的表中是否存在与它们匹配的行。

**//子查询**

**注意，子查询后面必须跟别名**