

MySQL数据库

第一章：初步认识Mysql及其客户端

一.知识点：

- 1.数据库概念
- 2.表概念
- 3.使用mysql工具
- 4.建库建表

二.重点：

- 2.1建库建表

三.内容：

- 3.1数据库概念

现在社会，是一个信息社会，北上广深或者硅谷归来，创业，成功性比较高，为什么？因为大城市、发达地方，信息比较发达！孙正义：时光穿梭机理论，信息，所存储的东西就是数据。也就是“数据就是资产”。

数据存到哪里呢？存到数据库里面。

数据库是什么呢？数据的仓库。厂房仓库能看，数据库看不见，所以有工具模拟（mysql）。影视：贪官，纪检委要查他，手机卡销毁，银行卡销毁。通话记录，存到移动、联通、电信等等运营商的数据库。

定义：数据库就是一些数据的集合。

- 3.2表概念

数据库是由若干张数据表组成，及其表的关系。

表--》表格：特征：由行和列

数据表：也是由行和列组成的二维集合

行：表示一个人，一条数据，这里的概念叫实体。

列：表示一个人的属性信息，这里叫属性、字段。

表是由行和列组成的二维数据集合。

- 3.mysql服务端：

- 3.1安装完毕之后，进入到mysql命令行客户端；

MySQL 8.0 Command Line Client

Enter password:

3.2 输入密码，进入到操作界面：

MySQL 8.0 Command Line Client

Enter password: ****

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 30

Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

3.3 使用show databases;显示mysql下面有哪些数据库 (show: 显示; database:数据库; s: 复数)

```
mysql> show databases;
+-----+
| Database |
+-----+
| dormdb   |
| information_schema |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
7 rows in set (0.00 sec)
```

3.4 use world(use +数据库名称);进入到相应的数据库下：

```
mysql> use world;
Database changed
mysql> _
```

3.5使用show tables,来查看当前数据库下的所有的数据表;

```
mysql> show tables;
+-----+
| Tables_in_world |
+-----+
| city             |
| country          |
| countrylanguage  |
+-----+
3 rows in set (0.00 sec)
```

3.6: 这个select * from country;来查看所有国家信息; 到后面还会继续讲。

```
mysql> select * from country;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Code | Name | LifeExpectancy | GNP | GNP01d | LocalName | Region | SurfaceArea | IndepY |
| ear | Population | | HeadOfState | | | Capital | Code2 | GovernmentF |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ABW | Aruba | 78.4 | 828.00 | 793.00 | Aruba | North America | Caribbean | 193.00 | N |
| ULL | 103000 | | Beatrix | | 129 | AW | | Nonmetropol |
| itan Territory of The Netherlands |
| AFG | Afghanistan | 45.9 | 5976.00 | NULL | Afganistan/Afqanestan | Asia | Southern and Central Asia | 652090.00 | 1 |
| 919 | 22720000 | | Mohammad Omar | | 1 | AF | | Islamic Emi |
| rate |
| AGO | Angola | 38.3 | 6648.00 | 7984.00 | Angola | Africa | Central Africa | 1246700.00 | 1 |
| 975 | 12878000 | | Jos?? Eduardo dos Santos | | 56 | AO | | Republic |
| |
| AIA | Anguilla | 76.1 | 63.20 | NULL | Anguilla | North America | Caribbean | 96.00 | N |
| ULL | 8000 | | Elisabeth II | | 62 | AI | | Dependent T |
| erritory of the UK |
| ALB | Albania | 71.6 | 3205.00 | 2500.00 | Shqip??ria | Europe | Southern Europe | 28748.00 | 1 |
| 912 | 3401200 | | Rexhep Mejdani | | 34 | AL | | Republic |
| ANP | Antigua and Barbuda | 75.6 | 199.00 | 199.00 | Antigua | North America | Caribbean | 196.00 | 1 |
| 298 | 68000 | | | | 13 | AG | | Commonwealth of the |
| ANS | Antigua and Barbuda | 75.6 | 199.00 | 199.00 | Antigua | North America | Caribbean | 196.00 | 1 |
| 298 | 68000 | | | | 13 | AG | | Commonwealth of the |
```

查看服务是否在运行:

计算机管理

文件(F) 操作(A) 查看(V) 帮助(H)

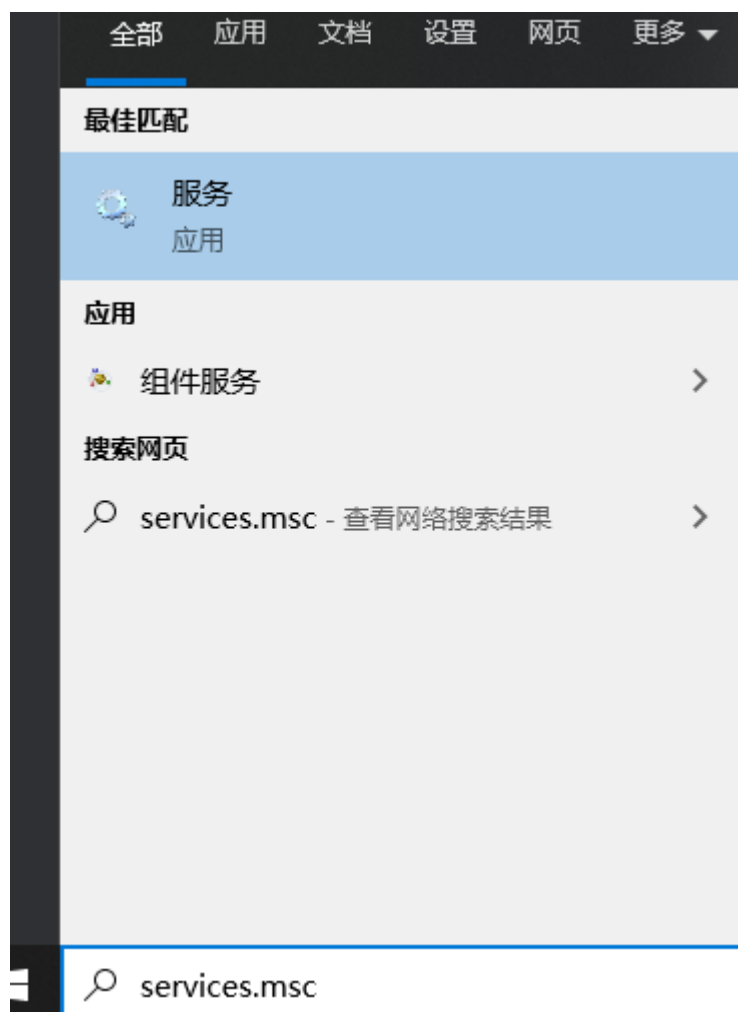
服务

名称	描述	状态	启动类型	登录为
MessagingService_4d30e67	支持...		手动(触发...	本地系统
Microsoft (R) 诊断中心标准收集器服务	诊断...		手动	本地系统
Microsoft Account Sign-in Assistant	支持...		手动(触发...	本地系统
Microsoft iSCSI Initiator Service	管理...		手动	本地系统
Microsoft Passport	为用...		手动(触发...	本地系统
Microsoft Passport Container	管理...		手动(触发...	本地服务
Microsoft Software Shadow Copy Provider	管理...		手动	本地系统
Microsoft Storage Spaces SMP	Micr...		手动	网络服务
Microsoft Store 安装服务	为 M...	正在运行	手动	本地系统
Microsoft Windows SMS 路由器服务	根据...		手动(触发...	本地服务
Mozilla Maintenance Service	Moz...		手动	本地系统
MySQL80		正在运行	自动	网络服务
Net.Tcp Port Sharing Service	提供...		禁用	本地服务
Netlogon	为用...		手动	本地系统
Network Connected Devices Auto-Setup	网络...	正在运行	手动(触发...	本地服务
Network Connection Broker	允许 ...	正在运行	手动(触发...	本地系统
Network Connections	管理...		手动	本地系统
Network Connectivity Assistant	提供 ...		手动(触发...	本地系统
Network List Service	识别...	正在运行	手动	本地服务

一定要查看mysql服务是否在运行状态;

命令行查看服务启动:

在搜索框输入：services.msc或“服务”



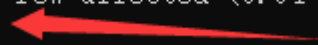
也可以出来计算机上的服务。

重点：


3.7 建库建表

3.7.1：建库：create database 数据库名;

```
mysql> create database school;  
Query OK, 1 row affected (0.04 sec)  
mysql>
```




```
mysql> show databases;
+-----+
| Database |
+-----+
| dormdb   |
| information_schema |
| mysql    |
| performance_schema |
| sakila   |
| school   |
| sys      |
| world    |
+-----+
```



3.8:建表

```
mysql> use school ;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```



```
mysql>
```

```
mysql> create table student(
    -> xuehao int,
    -> xingming varchar(20)
    -> );
Query OK, 0 rows affected (0.20 sec)

mysql> show tables;
+-----+
| Tables_in_school |
+-----+
| student          |
+-----+
1 row in set (0.00 sec)
```

语法格式:

create table 表名

(

列名 数据类型,

列名 数据类型

);

重点：

3.9mysql数据类型 int：数字型；这个是整型0 1 2

varchar(20):字符串string,这个和字符串一样；20是字符串的长度；

1、使用建议**

1、在指定数据类型的时候一般是采用从小原则，比如能用TINY INT的最好就不用INT，能用FLOAT类型的就不用DOUBLE类型，这样会对MYSQL在运行效率上提高很大，尤其是大数据量测试条件下。

2、不需要把数据表设计的太过复杂，功能模块上区分或许对于后期的维护更为方便，慎重出现大杂烩数据表

3、数据表和字段的起名字也是一门学问,不要和关键字一样。

5、数据库的最后设计结果一定是效率和可扩展性的折中，偏向任何一方都是欠妥的。

作业：

课后作业：网上书店及其三张表

第二章：完整性约束

一.知识点

1.1完整性约束的概念

1.2 实体完整性约束

1.3域完整性约束

1.4引用完整性约束

二.重点

实体完整性约束、域完整性约束、引用完整性约束

三.内容

约束概念：

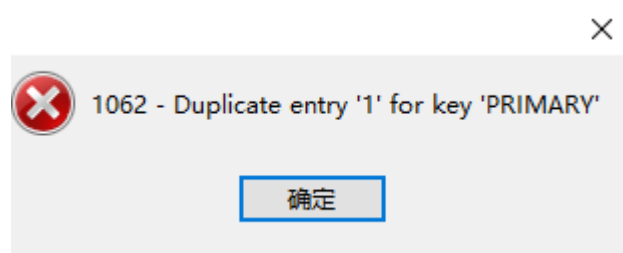
数据也是需要增加约束的，就是对数据的限制，数据类型就是限制。

3.1 实体完整性约束

总结：实体完整性约束，限制的是一行数据（表示一个实体）。

实体不可以重复，比如说我们班学生的学号不能重复，工号、银行卡号等不能，有且只有一个，钥匙限制，专业术语：主键。

无标题 @网上书店 (aa... x emp @hr (aaa) - 表 x 出版社表 @网上书店 (... x					
文件 编辑 窗口 帮助					
新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移 下移					
栏位 索引 外键 触发器 选项 注释 SQL 预览					
名	类型	长度	小数点	允许空值 (
publd	int	11	0	<input checked="" type="checkbox"/>	
pubName	varchar	30	0	<input checked="" type="checkbox"/>	
Address	varchar	40	0	<input checked="" type="checkbox"/>	



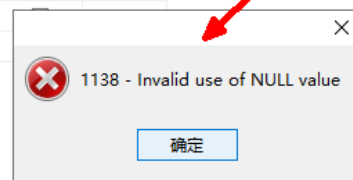
当我们增加了钥匙的时候，你再输入重复数据，会报错。系统不允许使用重复数据。

主键约束：有且只有一个；

自动递增：

当系统里面，有类似身份证号的时候，又加钥匙报错：

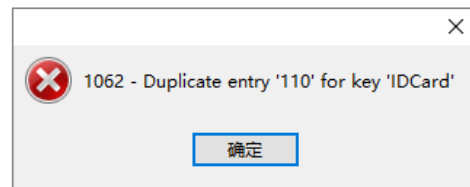
文件 编辑 窗口 帮助					
新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移 下移					
栏位 索引 外键 触发器 选项 注释 SQL 预览					
名	类型	长度	小数点	允许空值 (
EmpNo	int	11	0	<input type="checkbox"/>	1
EmpName	varchar	20	0	<input checked="" type="checkbox"/>	
Sex	char	2	0	<input checked="" type="checkbox"/>	
Age	int	11	0	<input checked="" type="checkbox"/>	
EmpAddress	varchar	20	0	<input checked="" type="checkbox"/>	
EmpPhone	varchar	11	0	<input checked="" type="checkbox"/>	
Birthday	datetime	0	0	<input checked="" type="checkbox"/>	
IDCard	varchar	18	0	<input checked="" type="checkbox"/>	



增加一个Unique的约束（唯一约束）：

在索引里面，增加

EmpNo	EmpName	Sex	Age	EmpAddress	EmpPhone	Birthday	IDCard
1	赵党	男	28	河南商丘	13888888809	2019-11-20	1 110
2	泰丽	人妖	20	泰国	(Null)	2019-11-06	1 110
3	老朽	男	500	花果山	(Null)	(Null)	(Null)



实体：一行数据的限制；

3.2域完整性约束

域：区域的意思；就是列：

域：列--》列完整性约束；

Not Null：非空约束，加完之后，如果你不填写，报错：

emp @hr (aaa) - 表

EmpNo	EmpName	Sex	Age	EmpAddress	EmpPhone	Birthday	IDCard
(Null)	(Null)	女	(Null)	(Null)	(Null)	(Null)	(Null)
1	赵党	男	28	河南商丘	13888888809	2019-11-20	1 110
3	老朽	男	500	花果山	(Null)	(Null)	(Null)
4	范耀辉	男	(Null)	(Null)	(Null)	(Null)	(Null)

1364 - Field 'EmpName' doesn't have a default value

确定

默认值约束：

default:


```
-- 设置默认值约束；
ALTER TABLE emp
change COLUMN empaddress address varchar(30)
default '河南郑州'
-- 更简单的
ALTER TABLE emp
ALTER COLUMN sex set DEFAULT '男'
```

总结：

针对一行设置：

主键约束--》让行有且只有一个；

唯一约束：

自动增长

针对一列设置：

Not Null:非空约束

default:默认值约束

四.作业

4.1 熟练掌握上述5个约束，其中默认值需要敲代码。建议组长及其能力可以挑战一把；

4.2 page51,page52 建表，熟练（两种方式）录屏作业

4.3 page65 Books表建一下，放数据进去

第三章：表的增删改操作

一.知识点

1.insert:增加

2.delete:删除

3.update:修改

二.重点

2.1 insert delete update的语法格式

三.内容

3.1 insert(增加)

```
# 增删改操作
# 1.insert插入数据;
# 语法格式: insert into 表名(列名1,列名2,...,列名n)
#           values(值1,值2,...,值n)
#注意: 值1--值n和列名1--列名n有对应关系, 个数一致; 类型一致;
-- select * from `图书类别表`
insert into 图书类别表(cateId,cateName)values(4,'计算机')
```

```
insert into 图书类别表(cateId,cateName)values(4,'计算机')
```

受影响的行: 1

时间: 0.140ms

```
insert into 图书类别表(cateId,cateName) values(5,'经济')
```

cateId	cateName
1	文学
2	小说
3	艺术
4	计算机
5	法律

当已经存在5的时候, 不能再增加, 因为cateId已经设置过主键了, 主键特征: 有且只有一个。

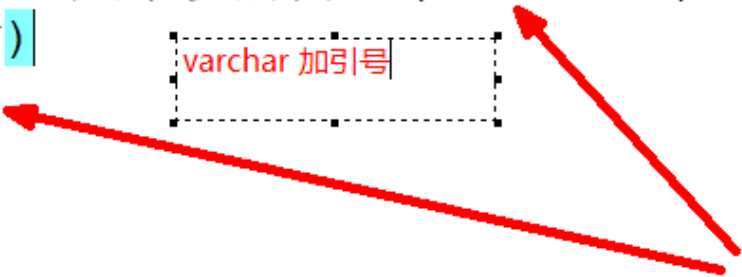
```
insert into 图书类别表(cateName) values('动画')
-- 自动增长类??? auto_increment
```

总结:

1.当数据表里面有自动增长列的时候, 可以省略自动增长列, 则系统自动插入数据。

2.varchar 日期类型两边需要加"

```
1 # varchar类型
2 insert into `图书类别表` (cateName)
3 VALUES (美食)
```



信息

概况

状态

[SQL] # varchar类型

insert into `图书类别表` (cateName)

VALUES (美食)

[Err] 1054 - Unknown column '美食' in 'field list'

```
INSERT into `图书表` (名称,作者,出版日期)values('我爱玩跳棋','田磊','2019-8-8')
```

3.插入多条语句

```
INSERT into `图书表` (名称,作者,出版日期)values
('我爱玩军棋','田大磊','2019-8-8'),
('我爱玩象棋','田大磊','2019-8-8'),
('我爱玩游戏','田大磊','2019-8-8')
```

3.2 delete操作 (删除)

```
-- 2.删除表中数据;
-- delete from 表名 where 条件
-- delete from `图书类别表` where cateId=5
```

#删除 多行语句的2种格式

```
DELETE from `图书表` where 图书编号=1 or 图书编号=3 or 图书编号=4
```

```
DELETE from `图书表` where 图书编号 IN(5,6,7)
```

#删除整个表

```
delete from 图书表
```

产出表中所有数据另外一个格式:

truncate table 表名

区别: delete from 表名 相当于拿个水瓢, 来删除, 在系统有一个登记记录。记录多次。

truncate table 表名, 举起水桶, 直接倒了。记录一次;

3.3 update操作 (更新)

```
#语法格式: update 表名 set 列名1=值1,...,列名n=值n [where 条件]
-- 3.1不加条件的时候
-- update 图书表 set 名称='国际象棋从入门到放弃'
-- 3.2一般更新都需要加条件;
update 图书表 set 名称='大学生学王者农药' where 图书编号=9
```

3.4 外键--》外籍是主键外面的一个键, 在另外一个表中, 主要作用就是保证数据的一致性, 多个表的数据保持一致。

比如说: 班级表: 主键班级编号 1 2 3

学生表:

学生编号 学生姓名 班级编号

1 田磊 1

2 李鹏飞 3

加的时候, 在外键表进行增加外键

-- 出问题了??? 图书类型编号只有1-7; 但是现在出现一个新的图书类型编号, 并且数据也进到数据库了, 这个就是无效数据。 -- 对于无效数据, 不应该进去, 最起码报错。

-- 出问题了??? 图书类型编号只有1-7; 但是现在出现一个新的图书类型编号, 并且数据也进到数据库了, 这个就是无效数据。 -- 对于无效数据, 不应该进去, 最起码报错。

```
update 图书表 set 图书类型编号=5 where 图书编号=13
```

```
update 图书类别表 set cateId=9 where cateId=5
```

-- CASCADE 级联更新|级联删除, 使用级联更新但是不要设置级联删除。 -- 在数据库里面, 系统真正跑起来都是逻辑删除, 更改的是记录的状态; -- 外键表也叫从表; 主键--》主表; 构成了主从表关系;

```
SQL> INSERT 图书表(名称,作者,图书类型编号)VALUES
抖音实战2','文直撞2',9)
```

Err] 1452 - Cannot add or update a child row: a foreign key constraint fails ('网上书店','图书表', CONSTRAINT '图书表_ibfk_1' FOREIGN KEY ('图书类型编号') REFERENCES '图书类别表' ('cateid'))

update 图书表 set 图书类型编号=5 where 图书编号=13

```
29 update '图书类别表' set cateId=9 where cateId=5
30
```

信息 概况 状态

[SQL] update '图书类别表' set cateid=9 where cateid=5

[Err] 1451 - Cannot delete or update a parent row: a foreign key constraint fails ('网上书店','图书表', CONSTRAINT '图书表_ibfk_1' FOREIGN KEY ('图书类型编号') REFERENCES '图书类别表' ('cateid'))

外键:

1.外键是建立在外键表上的。图书表;

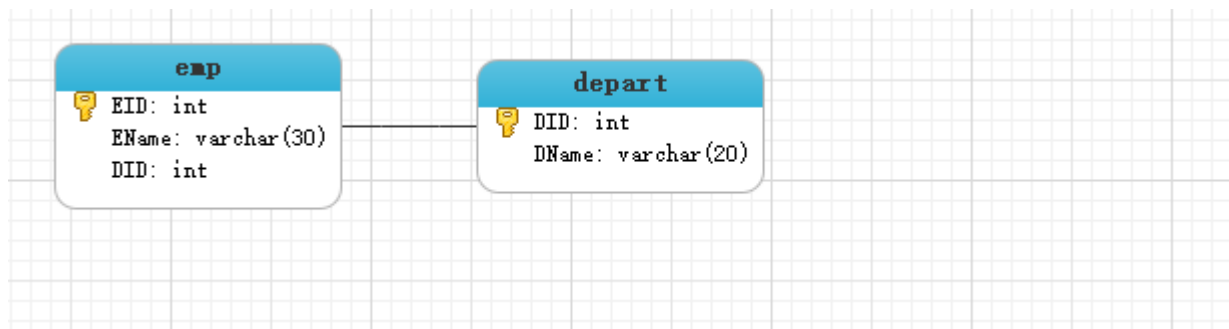
2.设置外键的时候, 前面是外键表的外键列, 后面是主键表和主键列

主键表	索引	外键	触发器	选项	注释	SQL 预览	主键表
名	索引	外键	触发器	选项	注释	SQL 预览	主键表
图书表_ibfk_1	图书类型编号	网上书店	图书类别表	cateid	RESTRICT	CASCADE	
图书表_ibfk_2	出版社编号	网上书店	出版社表	pubid	RESTRICT	CASCADE	

外键列
前面是外键表
图书表

当前
数据
库

主键
表的
主键



四.作业

- 4.1 给图书表增加外键（图书类型编号、出版社编号）
- 4.2 page63页，给成绩表设置外键。
- 4.3 针对之前数据表进行的增删改操作，每个表进行3条数据以上的操作。

第四章：基础查询

一.知识点

- 1.1 基本查询的语法格式
- 1.2 计算列、函数等

二.重点

- 2.1 查询的语法格式
- 2.2 函数的使用

三.内容

基础表构建：

```

-- 建立主键表（部门表）先有主键表，再有外键表
-- Create TABLE depart
-- (
--   DID int auto_increment PRIMARY key,
--   DName varchar(20)
-- )
# 插入三条语句
-- insert depart(DName)VALUES('技术部'),('人事部'),('财务部')
-- select * from depart
#2.建立外键表（员工表）
-- Create Table Emp(
--   EID int Auto_Increment PRIMARY Key,
--   EName varchar(30)not null,
--   DID int -- 2.2注意 外键表的外键类型一定要和主键表的主键类型一致;
-- )
-- 2.3 在外键表上建立外键
# 插入语句
-- INSERT Emp(EName,DID)values('李牛牛',1),('王晓',3),('李丽',2)
-- select * from emp

-- INSERT emp(EName,DID)values('路卫卫',5)  1 2 3

```

3.1查询 全表 select * from 表名

语法：

*：所有列名；from：从哪里来？

select 列名1, 列名2, 列名3,...,列名n

from 表名

指定列 别名 单条件 多条件

```

# 1.查询所有
select * from depart
select * from emp
# 2.查询员工表的部分列
select EName,Gender,Address from Emp
# 3.别名;姓名 性别 地址都是别名，不是真正的列名
select EName '姓名',Gender As '性别',Address as '地址'
from Emp
# 4.单条件查询;where 列名=值
select EName '姓名',Gender As '性别',Address as '地址'
from Emp
where EName='李丽'
# 5.单条件查询;where 列名=值
select EName '姓名',Gender As '性别',Address as '地址'
from Emp
where EName='李丽' Or Gender='男'

```

3.2.计算列

#6.year(列名) ,month(列名) 计算出来的, 查询使用函数求出来的值, 就是计算列
select EName 姓名,year(Birth)出生年份 , MONTH(Birth)出生月份 from emp

3.3.消除重复行

#7. 查询Emp (雇员表) 中的数据;来自于几个城市? ? ? ? 去除了重复行数据;
select DISTINCT Did 城市编号 from Emp

3.4 查询条数;

9.查询条数;LIMIT:可以出来前几条数据; MSSQL:TOP 3; Oracle:伪列;
-- select * from Emp LIMIT 3 (开始号,条数)
-- select * from Emp LIMIT 0,3 --0:开始的记录号, 不是表里面编号,3:条数
select * from Emp LIMIT 3,3 --
select * from Emp LIMIT 6,3

	EID	EName	DID	gender	Age	address	Birth
1	1	李牛牛	1	男	18	河南省郑州市	2001-07-01
2	2	王晓	3	男	19	北京市东城区	2000-01-01
3	3	李丽	2	女	17	上海市浦东新区	2002-08-01
4	5	方秀秀	3	女	88	德国	(Null)
5	6	靳博	2	男	16	日本	(Null)
6	7	赵党	1	男	17	商丘	(Null)
7	8	魏子波	2	男	18	安阳	(Null)
8	9	王双宜	1	女	18	洛阳	(Null)
9	10	马怡	3	女	15	南阳	(Null)

左边数据表有一列空白的就是记录号

3.5.排序

8.查询排序;Order By 待排序的列名 ;默认升序; ASC; 降序: DESC
select EName 姓名,Age 年龄 from emp ORDER BY Age DESC

3.6.是否为空查询

10.查询年龄是空NULL;IS NOT NULL
SELECT * from emp where age IS null

四.作业

4.1课后作业（除函数外全做）

班级表、学生表主外键关系：

```
# 创建主键表（主表）
-- Create Table Class(
--   id int not null PRIMARY Key,
--   ClassName varchar(50)not null
-- )
# 创建外键表（从表）
Create TABLE Student(
    id int not null PRIMARY Key,
    StuName varchar(50)not null,
    gender char(2),
    age SMALLINT,
    CID int,
    CONSTRAINT FOREIGN Key(CID) REFERENCES Class(ID)
)
```

第五章：函数

一.知识点

1.1字符串函数

1.2日期函数

1.3数学函数

1.4系统函数

二.重点

2.1字符串函数

2.2日期函数

2.3 函数+字段查询

三.内容

3.1字符串函数

concat(s1,s2...sn)：连接s1到sn

length(str)：求字符串长度

```
select LENGTH('我们在学习mysql')-- 20:5+5*3个字节
```

Upper('abcd'):变大写


```
select UPPER('abcD'),LOWER('abcD')
```

Ltrim(' abc')

RTrim (' bbc ')

```
# left:L;R:Right
-- select LENGTH(lTrim(' abc')),length(RTRIM(' abc '))
```

Replace('abcd','d','eee')

```
select replace('你是个大傻帽','傻帽','**')
UPDATE replyInfo set Content=replace(content,'法轮功','**')
```

Locate('a','abcddfdfd')

```
-- select LOCATE('价','涨价了在哪里? ')
# locate('子串','母串')查子串在母串中的位置
```

SubString('Hondfdf',2,3)

3.2日期函数

Now()

```
select Now() 当前日期和时间;Date(日期格式)time(日期时间格式)
```

date(date) time(date)

```
select Date(Now())'日期',Time(Now())'时间'
```

DateAdd()

```
select DATE_ADD(Now(),INTERVAL 1 day)
```

DateDiff(d1,d2)

```
DateDiff(d1,d2):前面的日期时间-后面的日期时间;差值: 天;
select DATEDIFF('2019-8-8','2018-8-8')
```

TimeStampDiff(unit,d1,d2);Unit:Second,Minute,month,year,day,week,quarter

```
-- 求出年龄
-- TimestampDiff(unit,d1,d2):后面减前面;d2-d1
-- select TIMESTAMPDIFF(SECOND,'2000-8-8','2019-12-4')
```

dayofweek

```
select DAYOFWEEK(now())
```

计算教师表中教员教龄：

```
select * from teacher
-- 如何算教龄;
select TeacherID,TeacherName,
TIMESTAMPDIFF(year,teachTime,now()) '教龄'
from teacher
```

3.3数学函数

ABS(数字)

Ceiling (小数) 地板

Floor (小数) 天花板

Power(2,3)

Round (58.32,1),四舍五入

Sqrt (4) :

```
# 3.数学函数;
select ABS(-1),ABS(0),ABS(1) -- 绝对值函数;
select FLOOR(24.3),CEILING(24.3)
-- Floor():小于24.3的最大整数;
-- CEILING(X): 大于x的最小整数;
select POWER(2,4)-- POWER(X,Y)x的y次幂
select ROUND(24.345,2) -- round(x,d):x的小数位为d
select sqrt(4) -- 算数平方根
```

3.4系统函数

Convert(1234,char(4))

Ifnull (expr,value):如果表达式为null，则返回表达式的值，否则返回null。

```
select user,host from mysql.user
```

```
select Current_User
```

```
-- #4.系统函数;
select CONVERT(1234,char(4)) -- 1234 ->char型; concat
select newsTitle,IFNULL(NewsContent,'无内容') from newsinfo
-- 查询的是newsContent,有内容, 显示的原来的内容; 没有内容显示后面的'无内容'
# 单列 ;mysql数据库表user中, 有哪些用户; root;localhost: 本机;
SELECT user,host from mysql.`user`
select CURRENT_USER -- 这时候, 显示的用户名@主机
```

四.作业

4.1 课后作业

4.2 **重点**是使用函数+字段的查询; (课前测)

第六章：模糊查询与分组查询

一.知识点

- 1.模糊查询
- 2.关键字: Like、Between、Group By
- 3.聚合函数的使用
- 4.正则表达式的使用

二.重点

- 2.1 like关键字模糊查询
- 2.2 正则是个小难点

三.内容

- 3.1 模糊查询like 之%和_的使用

```

select * from studentinfo where StudentName='曲灵风'
UPDATE studentinfo set StudentName='PD_TOM' where StudentID='2011001010'
# 名字里面以风结尾的名字; like:像;%:代表任意个字符;
SELECT * from studentinfo where StudentName like '%风'
#名字里面含有处, 如何查?
SELECT * from studentinfo where StudentName like '%处%'
#查名字姓刘???
SELECT * from studentinfo where StudentName like '刘%'
# _下划线代表单个字符;
SELECT * from studentinfo where StudentName like '刘__'
# 假设名字里面PD_TOM_KeLUsi;出了问题??? ESCAPE:逃避; 避开;
#ESCAPE 查询的时候, _不起作用;
SELECT * from studentinfo where StudentName like '%/_%' ESCAPE '/';

```

3.2 between and in 正则表达式使用

```

# BETWEEN A and B;介于之间 [A,B]等价于 >=A and <=B
select * from studentinfo where age between 18 and 24
select * from studentinfo where age >=18 and age<=24
select * from studentinfo where age BETWEEN 24 and 18 # 这个是有问题
# in 在某某地方找;in的效率比字段=值要低;
select * from studentinfo where city = '周口' or city='洛阳' or city='商丘'
select * from studentinfo where city in('周口','洛阳','商丘')
# 正则表达式;JS
select * from studentinfo where StudentNO like '412%' or
StudentNO like '413%'
select * from studentinfo where StudentNO regexp '^[4][1][2,3]'

```

3.3 聚合函数

```

# 聚合函数: 求和函数sum();平均值函数avg();max()最大值函数
# min()最小值函数; count():统计函数;
select examId 考试编号,StudentID 学生编号,SubjectId 科目号,exam 成绩
from exam
select * from studentinfo
#现在要查看2011001002的最高分???
select StudentID 学生编号,max(exam)最高分,min(exam)最低分,avg(exam) 平均分,sum(exam)总分
from exam
where StudentID='2011001002'
# 查看一下当前学生表, 有多少学生;
select count(*) 学生总数 from studentinfo

```

3.4 分组

```
# 分组:Group By
select city 城市,count(*)总数 from studentinfo
group By city
#多列分组;
select ClassID 班级编号,Gender 性别,Count(*) from studentinfo
GROUP BY Gender,ClassID
# 计算一下人数超过2个? 针对分组后的结果进行比较判断, 不能用where 了。having
#查的目标是某个班有多少个男生, 多少个女生; 逻辑上是有问题的, 即使有结果, 也不对!
select ClassID 班级编号,Gender 性别,Count(*) from studentinfo
GROUP BY Gender,ClassID
HAVING count(*)>2
# where -> GROUP BY ->Having
```

StudentID	StudentName	Gender	Age	Birthday	StudentNO	ClassID	BeginTime	Phone	Province	City	Email
2011001002	马钰	男	20	1991-02-21 00:00:00	107021991022102	1	2011-5-6	15125741526	河南	商丘	my91@263.com
2011001003	谭处端	男	21	1990-06-21 00:00:00	103031990062102	1	2011-5-6	15890172541	河南	洛阳	tcd@163.com
2011001004	黄蓉	女	22	1990-02-11 00:00:00	127011990021106	1	2011-5-6	18637165241	河南	周口	huangrong@26
2011001006	刘处玄	男	18	1990-10-14 00:00:00	101021990101403	1	2011-5-6	15568852141	河南	郑州	LCX90@hotmail
2011001008	丘处机	男	24	1991-06-21 00:00:00	101031991062106	1	2011-5-6	15142153256	河南	郑州	qcj@163.com
2011001009	王处一	男	20	1991-11-15 00:00:00	107021991111505	1	2011-5-6	15125741529	河南	洛阳	王99@sina.com
2011001010	PD_TOM	男	19	1992-05-06 00:00:00	137021991022102	1	2011-5-6	15125471535	河南	洛阳	hdt@sohu.com
2011001011	孙不二	男	20	1991-08-19 00:00:00	127021991080901	1	2011-5-6	15125471539	河南	周口	sun1991@163.c
2011003001	刘瑛	女	25	1991-11-07 00:00:00	101061991110704	3	2011-5-6	18854215874	河南	郑州	liuying@sohu.c
2011003002	曲灵风	女	25	1990-02-01 00:00:00	113031989060401	3	2011-5-6	18878452365	河南	南阳	qlf@taohuadao
2011003003	陈玄风	男	24	1990-02-01 00:00:00	105221990052902	3	2011-5-6	18878452361	河南	安阳	cxf@163.com
2011003004	梅超风	女	24	1990-02-01 00:00:00	101041990091604	3	2011-5-6	18878452397	河南	郑州	mcf@sohu.com
2011003007	裘千尺	女	24	1990-02-01 00:00:00	101021992010703	3	2011-5-6	13852417412	河南	郑州	jueqingdan@16

城市	总数
商丘	1
洛阳	3
周口	2
郑州	5
南阳	1
安阳	1

四.作业

4.1 课后作业

第七章：多表数据连接与查询

一.知识点

1.1 多表连接的概念

1.2 笛卡尔乘积、左外连接、右外连接、内连接、自身连接

1.3 查询合并

二.重点

2.1 左外连接、右外连接、内连接、自身连接

三.内容

3.1 在实际业务中，我们很多时候，都需要用到多表连接。设计数据库，时候，比如：教师授课表，这个是一个中间表；

我们先在要看到哪位老师，讲的什么课程？教师表 课程表，会发现这两个表没有关系。教师授课表这个表跟教师表和课程表都有关系。现在就需要3个表的数据关联。就需要多表连接查询。

3.2 笛卡尔乘积：最终是两个表条数相乘；

```
# 笛卡尔乘积;t_a:4条; t_b: 4条; 16=4*4;
select * from t_a,t_b
```

目标：哪些老师上课？？？

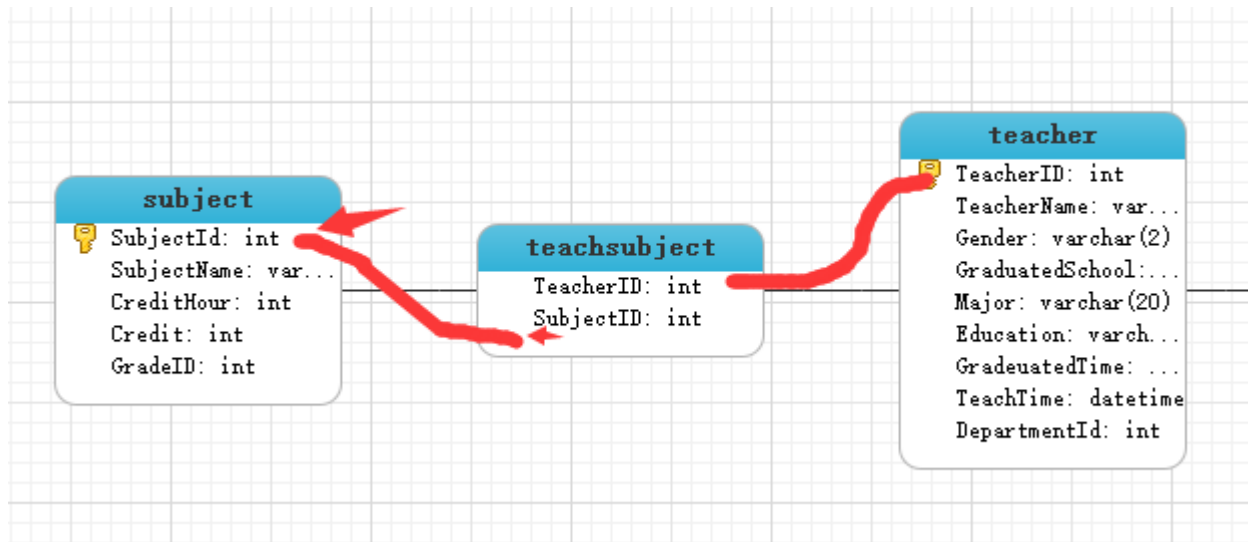
4 select * from teacher,subject

信息	结果1	概况	状态			
TeacherID	TeacherName	Gender	GraduatedSchool	Major	Education	Grad ^
1	王重阳	男	全真研究院	阵法	博士	2000
2	黄药师	男	桃花岛大学	格斗	博士	1999
3	段智兴	男	大理皇家学院	格斗	博士	1998

只读 查询时间: 0.016s 第 1 条记录 (共 132 条)

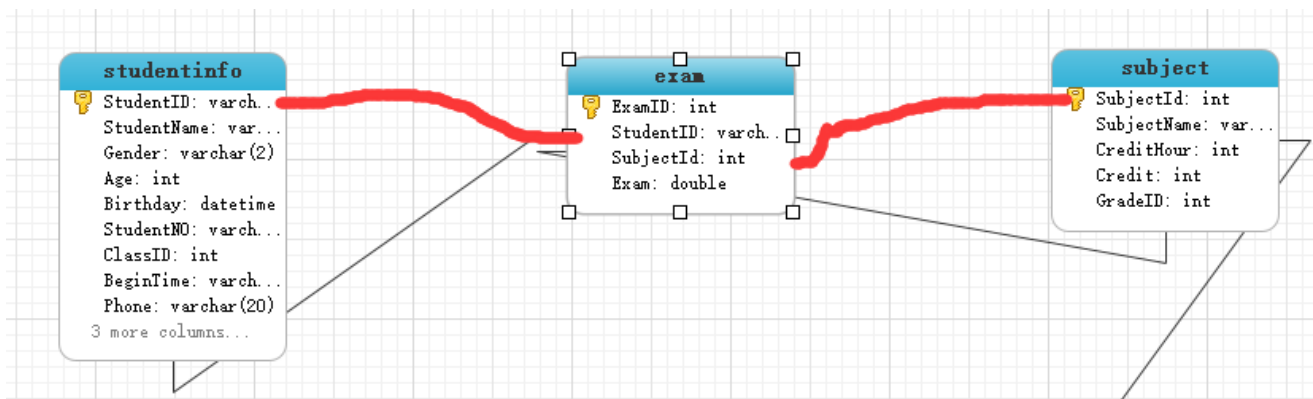
3.3 加上查询条件;

```
# 我们现在要看哪个学生在哪个班??? 显示班级名;外键关系;classID,在studentInfo, class表里面; 都可以使用;
select S.StudentID,S.studentName,C.ClassId,C.className
  from studentinfo S,class C
 where S.ClassID=C.ClassID
/* 总结: A B是别名
   select A.列名,B.列名
   from 表A A,表B B
   where A.列=B.列
*/
```



三表连接查询:

```
# 如何实现查看教师授课哪门课程的查询呢???
select T.teacherID,T.teacherName,S.subjectName
  from teacher T,teachsubject TS,subject S
 where T.teacherId=TS.teacherId and TS.subjectID=S.subjectID
# page 142:某个学生, 某门课考了多少分?
SELECT stu.studentID,stu.studentName,s.subjectName,e.exam
  from studentinfo Stu,Subject S,Exam e
 where Stu.StudentID=e.studentId and S.subjectID=e.subjectID
```



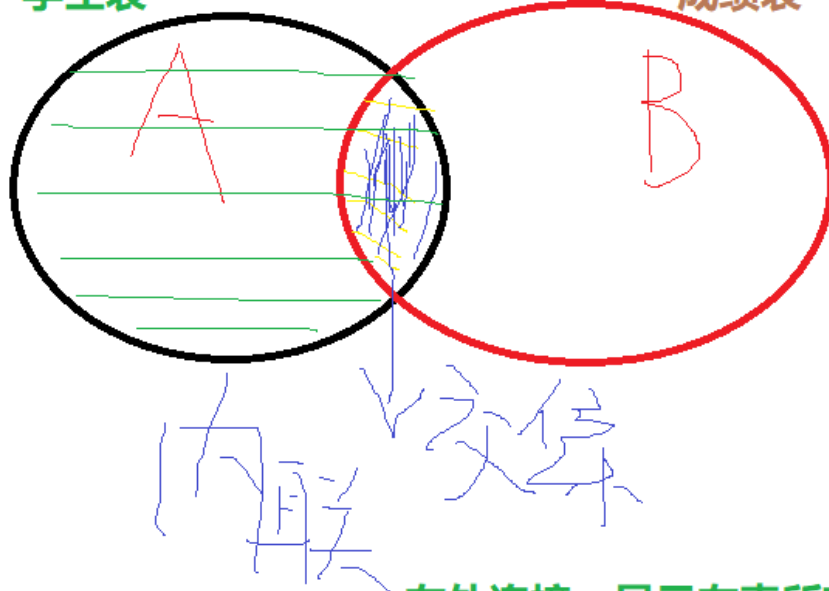
#3.等值连接/内连接

```
-- select 表名.列名1,...,表名.列名n
-- from 表名 A [Inner|Left|Right]Join 表B B
```

```
-- on 表达式
select S.StudentId,S.studentName,C.ClassId,C.className
  from studentinfo S,class C
where S.ClassID=C.ClassID
# 使用内连接; 原来的逗号, inner join ;where ->on 条件
select S.studentId 学号,S.studentName 姓名,C.classId,C.className 班级
from studentinfo S inner Join Class C
on S.classId=C.classId
# 三个表 的内连接
SELECT stu.studentID,stu.studentName,s.subjectName,e.exam
from studentinfo Stu,Subject S,Exam e
where Stu.StudentID=e.studentId and S.subjectID=e.subjectID
# inner join on格式*****
SELECT XS.studentID,XS.studentName,s.subjectName,e.exam
FROM studentinfo XS inner Join Subject S Inner Join Exam e
on XS.studentId=e.studentId and S.subjectId=e.subjectID
#需要注意的是from 表A inner join 表B on a.列=B.列 inner join 表C on 条件
SELECT XS.studentID,XS.studentName,s.subjectName,e.exam 成绩
FROM studentinfo XS inner Join Exam e -- 学生表 成绩表 ;成绩表是一个中间表;
on XS.studentId=e.studentId -- 学生.学号=成绩表.学号
Inner Join Subject S -- 科目表
on S.subjectId=e.subjectID -- 科目表.科目号=成绩表.科目号
-- 等价于from A,B where A.列=B.列
```

学生表

成绩表



左外连接, 显示左表所有数据, 右表只显示, 交集的数据


```
#4.左外连接|右外连接;
SELECT * from studentinfo -- 13个学生

select distinct studentId from exam -- 8个学生 有成绩;
-- 使用内连接来查询成绩;都是考试的学生,但是有的学生没有考试??
select XS.StudentName ,E.exam
FROM studentinfo XS JOIN Exam E -- Inner 也可以省略,一般不省略
on XS.studentId=e.StudentId

-- 左外连接;
select XS.StudentName ,IFNULL(E.exam,'缺考')
FROM studentinfo XS Left Outer JOIN Exam E -- Outer 可以省略
on XS.studentId=e.StudentId
```

5.自身连接:

```
#自身连接;自己表和自己表进行连接;自身表当成两个表来处理。
select * from product
# id:编号; productName:产品名称;parent_id:上级编号;
# 树形菜单;0:顶级;1
# 目标: 显示当前产品,及其上级商品的名称;实现树形菜单的数据库显示;
# 子表(Child): C; 父表: (Parent): P
SELECT C.Id 编号,C.ProductName 名称,P.ProductName 上级名称
from product C INNER JOIN product P
on C.parent_Id=P.id
# 自身连接使用范围: 产品类别表; 部门表 (编号 部门名称 上级编号)
```

编号	名称	上级名称
	3 笔记本	数码产品
	4 智能手机	数码产品
▶	5 电器	家用产品
	6 家具	家用产品
	7 冰箱	电器
	8 洗衣机	电器
	10 别克	汽车品牌
	11 宝马	汽车品牌
	12 雪佛兰	汽车品牌

四.作业

4.1课后作业

4.2录屏