快捷键：

win+D:显示/隐藏脑桌面

win+E:打开资源管理器

win+R:打开运行窗口(cmd命令行/calc计算器/mspaint画图)

Ctrl+S:保存

Ctrl+C:复制

Ctrl+V:粘贴

Ctrl+X:剪切

Ctrl+Z:撤销当前操作

Ctrl+A:全选

Ctrl+F:查找

Ctrl+R:替换

Shift+回车:换行

Alt+F2:选择浏览器运行

Alt+Tab:切换当前窗口(向右)

Alt+Shift+Tab:切换窗口(向左)

F5:刷新

Ctrl+B:在浏览器中查看网页

Ctrl+Shift+N:新建模板网页

Alt+F4:关闭当前的窗口

Lorem:生成随机文本

Backspace:回到上一级

在cmd中：

control+C:强制停止

cls:清零

2019-4-29

1.软件工程

程序设计阶段->软件设计阶段->软件工程阶段

软件工程：研究和应用如何系统性的、规范化的、

可定量的过程化方法去开发和维护软件

\*\*软件过程/软件生命周期\*\*

(1)软件定义期

1)可行性研究阶段--《可行性研究报告》

技术、人员、资金、时间、政策法规、风俗、回报率

2)需求分析阶段--《软件需求说明书》

功能性需求、非功能性需求

(2)软件开发期

3)概要设计阶段--架构师

技术选型、功能模块划分、数据结构设计

4)详细设计阶段--设计师

模块实现细节、属性、方法、算法

5)编码实现阶段

UI设计师--网页效果图

前端工程师--把网页效果图转化为HTML、css、js

后端工程师--提供项目中数据

6)软件测试阶段

软件测试工程师

(3)软件维护期

7)项目部署阶段

将项目部署到服务器

8)项目维护阶段

2.学子商城功能型需求

前台：www.codeboy.com

后台：www.codeboy.com/admin/login.html

前台子系统/APP子系统：

商品模块：首页、搜索、列表、详情

用户模块：注册、登录、收藏夹、用户中心

购物车模块：加入购物车、修改购物车、下单

后台子系统

商品模块：添加、列表、删除、修改

用户模块：列表、删除、详情

订单模块: 列表、修改

3.web项目中的服务器

Server:为客户端提供各种服务的功能强大的计算机

访问服务器

(1)服务器地址:域名/地址

查看域名对应的IP

ping www.codeboy.com

(2)对应的服务：端口号

(3)提供该服务所使用的协议

4.访问自己电脑上的web服务器

启动web服务器 /WebServer/start.cmd

http://127.0.0.1:80 通过自己电脑的IP地址

http://localhost:80 通过自己电脑服务器的域名

获取自己电脑在局域网的IP地址

ipconfig 以太网适配器 以太网 ->IPv4

http://10.1.180.152:80 通过局域网的IP地址访问

课后任务：

(1)复习今天的内容

(2)启动服务器，同桌之间使用局域网IP地址相互访问

(3)预习MySQL数据库

2019-4-30

复习：

软件生命周期/软件过程

软件定义期

可行性研究阶段/需求分析阶段/详细设计阶段/编码实现阶段/

项目测试阶段

软件维护期

项目部署阶段/项目维护阶段

服务器

域名/IP地址

端口号

协议

1.数据的存储形式

内存/特定文件/第三方服务器/数据库服务器

2.什么是数据库

数据库是按照一定的形式来组织、存储数据，目的是为了操作数据

--增删改查

(1)数据库发展历史

网状数据库->层次型数据库->关系数据库->非关系型数据库

(2)关系型数据库逻辑结构

Server(数据库服务器)->Database(数据库)->Table(数据表)->Row(行)->Cloumn(列)

(3)常见的关系型数据库

SQLite mysql sql server oracle

Linux/windows

3.使用MySQL数据库

Mysql->sun->oracle https://www.mysql.com/

Matin->MariaDB https://www.mysql.com/

服务器套装——XAMPP

下载地址：https://www.apachefriends.org/download.html

(1)mysql部署结构

服务器端：负责存储/维护数据--银行总行机房

C:/xampp/mysql/bin/mysqld.exe 启动文件

确保端口3306不被占用

客户端：负责向数据库发起增删改查--ATM机

C:/xampp/mysql/bin/mysql.exe 客户端工具

(2)使用客户端连接服务器端

mysql.exe -h127.0.0.1 -P3306 -uroot

-h host 主机(IP地址/域名)

-P port 端口

-u user 用户名

-p password 密码

MySQL -uroot 简写形式

4.MySQL常用得管理命令

show databases; 显示服务器上当前所有的数据库

quit 退出服务器的连接

use 数据库名;进入指定的数据库

show tables; 显示当前数据库中所有的数据表

desc 表名;描述表中都有哪些列(表头)

练习：进入数据库MySQL,查看有多少个数据表。

练习：进入到数据库xuezi,查看表users下有哪些列

5.SQL命令

SQL:Structured Query Language,结构化查询语言,

用于操作关系型数据库服务器，对数据执行增删改查。

两种执行方式：

(1)交互模式:客户端输入一行，点击回车，服务器执行一行，

适用于临时性的查看数据。

(2)脚本模式:客户端把要执行的命令编写在一个脚本文件中，

一次性的提交给服务器执行，适用于批量的操作数据。

mysql -uroot<C:/User/UID/Desktop/邱婷/达内/正课/2019-4-30/01.sql

练习：编写脚本文件02.sql,查看所有的数据库，进入到xuezi数

据库，查看users下有哪些列

SQL语句规范：

(1)每条SQL命令必须以英文的分号作为结束，一条命令可以跨越

多行。

(2)假如某一条命令出现语法规范，则这条命令以及后边所有的

命令不会再执行。

(3)SQL不区分大小写，习惯上数据库关键字用大写，非关键字用

小写。

(4)SQL命令中可以使用单行注释(#...)和多行注释(/\*...\*/),

注释的内容不会被服务器执行。

\*\*\*常用得SQL命令：

(1)丢弃数据库，如果存在

DROP DATABASE IF EXISTS jd;

(2)创建新的数据库

CREATE DATABASE jd(数据库名称);

(3)进入jd数据库

USE jd;

(4)创建保存数据的表

CREATE TABLE student(

sid INT,#integer

name VARCHAR(16),#variable character

sex VARCHAR(1),

score INT

);

(5)向数据表中插入数据

INSERT INTO student VALUES('1','tom','M','82');

(6)查询数据

SELECT \* FROM student;

(7)修改数据

UPDATE student SET sex='M',score='100' WHERE sid='2';

(8)删除数据

DELETE FROM student WHERE sid='3';

练习：编写脚本文件04\_tedu.sql,先丢弃数据库tedu,如果存在；

再创建数据库tedu

练习：进入tedu数据库，创建保存员工数据的表emp,包含员工编号eid,姓名ename,地址addr,电话phone

练习：往数据表emp中插入5条数据，查询最终的结果。

练习：编写脚本文件05\_dang.sql,先丢弃再创建数据库dang,进入该数据库，创建保存图书数据的表book,包含编号bid,标题title,价格price;插入数据，删除其中一条，修改其中1条，查询最终结果。

课后任务：

(1)复习内容

(2)练习：编写脚本文件xz.sql,先丢弃再创建数据库xz,进入该数据库，创建保存用户数据的表user,包含编号uid,用户名uname,密码upwd,邮箱email,电话phone,真实姓名userName,注册时间regTime，是否在线isOnline;插入数据，删除其中一条，修改其中1条，查询最终结果。

2019-5-5

标准SQL语句分类

DDL:Data Define Language 定义数据结构

CREATE/DROP/ALTER

DML:Data Manipulate Language 操作数据

INSERT/UPDATE/DELETE

DQL:Data Query Language 查询数据

SELECT

DCL:Data Control Language 控制用户权限

GRANT(授权)/REVOKE(收权)

1.计算机存储字符

(1)如何存储英文字符

ASCII:总共有128个，对所有的英文字母和符号进行了编码

Latin-1:总共有256个，兼容ASCII，同时对欧洲的符号进行了编码

(2)如何存储中文字符

GB2312:对常用的6千多汉字进行了编码，兼容ASCII码

GBK:对两万多汉字进行了编码，兼容GB2312

BIG5:台湾繁体字编码，兼容 ASCII码

Unicode:对世界上主流国家常用的语言进行了编码，兼容ASCII码，不兼容GB2312，GBK，BIG5；具体的存储方案有UTF-8、UTF-16、UTF-32

中文乱码产生的原因：mysql默认使用Latin-1编码。

(3)解决MySQL中文乱码

Sql脚本文件另存为的编码

客户端连接服务器端使用的编码(SET NAMES UTF8)

服务器端创建数据库使用的编码(CHARSET=UTF8)

练习：编码脚本文件01\_sohu.sql,先丢弃在创建数据库sohu,设置

存储的编码为UTF8，进入该数据库，创建保存新闻数据的表news,包含编码nid,标题title,发表时间pubTime,来源origin,内容content;插入数据,删除1条，修改1条

2.MySQL中的列类型

创建数据表的时候，指定的列可以存储的数据类型

CREATE TABLE t1(nid 列类型);

(1)数值型--引号可加可不加

TINYINT 微整形，占1字节(8位)，范围-128~127

SMALLINT 小整形，占2个字节(16位)，范围-32768~32767

INT 整形，占4个字节(32位)，范围-2147483648~2147483647

BIGINT 大整形，占8个字节(64位)

FLOAT 单精度浮点型，占4个字节，范围3.4E–38～3.4E+38(E38:10的38次方)，比INT范围大得多，可能产生计算误差

DOUBLE 双精度浮点型，占8个字节，范围1.7E–308～1.7E+308，范围比BIGINT大的多，可能产生计算误差

DECIMAL(M,D) 定点小数，不会产生计算误差，M代表总的有效位数，D代表小数点后的有效位数

BOOL 布尔型，只有两个值TRUE/1、FALSE/0，TRUE和FALSE不能加引号，最终存储的值是1和0，会变成TINYINT来存储。常用于存储只有两个状态的值，例如性别、是否在线、是否为会员...

(2)日期时间型--必须加引号

DATE 日期型 2019-12-25

TIME 时间型 14:32:50

DATETIME 日期时间型 2019-12-25 14:32:50

(3)字符串类型--必须加引号

VARCHAR(M) 变长字符串，不会产生空间浪费，操作速度相对慢，M最大值是65535，常用于存储姓名、文章标题、文章的内容

CHAR(M) 定长字符串，可能产生空间浪费，操作速度相对快，M最大值是255，常用于存储手机号码、身份号码等固定长度的数据

TEXT(M) 大型变长字符串，M最多是2G

练习1：编写脚本文件02\_xuezi.sql,设置存储的编码UTF8，创建

数据库xuezi,进入该数据库，创建保存笔记本数据的表laptop,包含编号lid,标题title,价格price,库存量stockCount,上架时间pubTime,是否显示在首页isIndex;插入4条数据。

练习2：编写脚本文件03\_tedu.sql,设置存储的编码为UTF8,创建

数据库tedu,进入该数据库，创建保存部门数据的表emp,包含部门编号did,名称dname,员工数量empCount,插入以下数据

10 研发部 3

20 市场部 2

30 运营部 2

创建保存员工数据的表emp,包含员工编号eid,姓名ename,性别sex,生日birthday,工资salary,所属部门编号deptId,插入数据。

3.列约束

MySQL可以对插入的数据进行特定的验证，只有满足条件才允许插入到数据表中，否则被认为是非法的插入。

例如：一个人的性别只能是男或者女，一个人的年龄0~100

CREATE TABLE t1(id 列类型 列约束);

(1)主键约束--PRIMARY KEY

声明了主键约束的列上不允许插入重复的值，一个表中只能有一个主键，通常添加在编号列。设置了主键就不能再插入NULL值。

表中查询的记录会按照主键由小到大

NULL表示空，在插入数据时，无法确定要保存的值。例如无法确定员工的性别、生日、工资、所属部门都可以使用NULL；NULL和任何值比较都不等，包括两个NULL比较。

注意：不能加引号

(2)唯一约束--UNIQUE

声明了唯一约束的列上，不允许插入重复的值，允许插入NULL，而且允许插入多个NULL

课后任务：

(1)复习今天内容

(2)练习

编写脚本文件xz.sql,设置存储的编码utf8,创建数据库xz,进入该数据库，创建保存笔记本家族的表laptop\_family，包含fid,家族名称fname,笔记本数量laptopCount;创建保存笔记本数据的表laptop，包含lid,title,price,规格spec,详情details,上架时间shelfTime,是否在售isOnsale,所属家族编号familyId，插入数据。

2019-5-6

复习：

Mysql存储

ASCII/Latin-1/GB2312/GBK/BIG5/Unicode

脚本文件/客户端连接服务器/服务器创建数据库统一UTF8

列类型

数值型

TINYINT/SMALLINT/INT/BIGINT

FLOAT/DOUBLE/DECIMAL

BOOL 性别 1/0

日期时间型

DATE/TIME/DATETIME

字符串型

VARCHAR/CHAR/TEXT

1.列约束

CREATE TABLE t1(id 列类型 列约束);

(1)主键约束——PRIMARY KEY

(2)唯一约束——UNIQUE

(3)非空约束——NOT NULL

声明了非空约束的列上不允许插入NULL

(4)默认值约束——DEFAULT

可以使用DEFAULT关键字设置默认值，有两种方式可以使用默认值

INSERT INTO laptop\_family VALUES(40,'小米',DEFAULT);

INSERT INTO laptop\_family(fid,fname) VALUES(50,'戴尔');

练习：设置笔记本的默认价格为6999，分别使用两种默认值插入数据

(5)检查约束——CHECK

检查约束可以对要插入的数据进行自定义的验证

CREATE TABLE student(

score TINYINT CHECK(score>=0 AND score<=100)

);

mysql不支持检查约束，会降低数据的插入速度

(6)外键约束——FOREIGN KEY

声明了外检约束的列上，取值必须在另一个表的主键列上出现过，两者的列类型保持一致，允许使用NULL

FOREIGN KEY(列名) REFERENCES 数据表(主键列)

2.mysql的自增列

AUTO\_INCREMENT:自动增长，假如一个列上声明了自增列，无需手动赋值，直接赋值为NULL，就会获取当前的最大值，然后加1插入

注意：只适用于整数型的主键列

自增列允许手动赋值

练习：编写脚本文件01\_tedu.sql,创建数据库tedu,进入该数据库；创建保存部门数据的表dept(did,dname)

10 研发部 20 市场部 30 运营部 40测试部

创建保存员工数据的表emp(eid,ename,sex,birthday,工资salary,所属部门编号deptId),插入15条数据。

3.简单查询

(1)查询特定的列

示例：查询出所有员工的姓名和工资

SELECT ename,salary FROM emp;

练习：查询出所有员工的编号，姓名，性别，生日

SELECT eid,ename,sex,birthday FROM emp;

(2)查询所有的列

SELECT \* FROM emp;

SELECT eid,ename,sex,birthday,salary,deptId FROM emp;

(3)给列起别名

示例：查询所有员工的姓名和工资，使用中文别名

SELECT ename AS 姓名,salary AS 工资 FROM emp;

练习：查询所有员工的编号，姓名，性别，生日，使用中文别名

SELECT eid AS 编号,ename AS 姓名,sex AS 性别,birthday AS 生日 FROM emp;

练习：查询所有员工的编号和姓名，使用一个字母别名

SELECT eid AS a,ename AS b FROM emp;

在给列起别名的时候，AS关键字可以省略的，保留空格

(4)显示不同的记录/合并相同项

示例：查询都有哪些性别的员工

SELECT DISTINCT sex FROM emp;

DISTINCT(合并)

练习：查询出员工都分布在哪些部门

SELECT DISTINCT deptId FROM emp;

(5)查询时执行计算

示例：计算2+5-6.3\*7.9+4\*5.4

SELECT 2+5-6.3\*7.9+4\*5.4;

练习：查询所有员工的姓名及其年薪

SELECT ename,salary\*12 AS 年薪 FROM emp;

练习：假设每个员工工资加500，年终奖加5000，查询出所有员工的姓名及其年薪，给列起中文别名。

SELECT ename AS 姓名,(salary+500)\*12+5000 AS 年薪 FROM emp;

(6)查询的结果集排序

示例：查询所有的部门，结果集按照编号从小到大排序

SELECT \* FROM dept ORDER BY did ASC;#ascendant升序

示例：查询所有的部门，结果集按照编号从大到小排序

SELECT \* FROM dept ORDER BY did DESC;#Describe描述 Descendant降序

练习：查询所有的员工，结果集按照工资降序排列

SELECT \* FROM emp ORDER BY salary DESC;

练习：查询所有的员工，结果集按照年龄从小到大排列

SELECT \* FROM emp ORDER BY birthday ASC;

练习：查询所有的员工，结果集按照姓名升序排序

SELECT \* FROM emp ORDER BY ename ASC;

练习：查询所有的员工，结果集按照工资升序排序，如果工资相同按照姓名升序排列

SELECT \* FROM emp ORDER BY salary ASC,ename ASC;

练习：查询所有的员工，所有的女员工显示在前面，如果性别相同，按照生日降序排列

SELECT \* FROM emp ORDER BY sex,birthday DESC;

ORDER BY 可以按照数值，日期时间，字符串来排序，默认按照ASC升序排列

(7)条件查询

示例：查询出姓名为king的员工的所有的列

SELECT \* FROM emp WHERE ename='king';

练习：查询出编号为5的员工的编号，姓名，工资

SELECT eid,ename,salary FROM emp WHERE eid=5;

练习：查询出20号部门下员工所有的列

SELECT \* FROM emp WHERE deptId=20;

练习：查询出女员工所有的列

SELECT \* FROM emp WHERE sex=0;

练习：查询出工资在5000以上的员工的所有列

SELECT \* FROM emp WHERE salary>=5000;

比较运算符：>= <= > < = !=(不等于)

练习：查询出1991-1-1后出生的员工所有列

SELECT \* FROM emp WHERE birthday>'1991-1-1';

练习：查询出不在10号部门的员工所有列

SELECT \* FROM emp WHERE deptId!=10;

练习：查询出没有明确部门的员工所有列

SELECT \* FROM emp WHERE deptId IS NULL;

练习：查询出有明确部门的员工所有列

SELECT \* FROM emp WHERE deptId IS NOT NULL;

练习：查询出工资在6000以上的男员工所有列

SELECT \* FROM emp WHERE salary>=6000 AND sex=1;

练习：查询出工资在5000~8000之间员工所有列

SELECT \* FROM emp WHERE salary>=5000 AND salary<=8000;

SELECT \* FROM emp WHERE salary BETWEEN 5000 AND 8000;

练习：查询出工资在5000以下和8000以上的员工所有列

SELECT \* FROM emp WHERE salary<=5000 OR salary>=8000;

SELECT \* FROM emp WHERE salary NOT BETWEEN 5000 AND 8000;

练习：查询出1990年之前和1993年之后出生的员工所有列

SELECT \* FROM emp WHERE birthday<'1990-1-1' OR birthday> '1993-12-31';

SELECT \* FROM emp WHERE birthday NOT BETWEEN '1990-1-1' AND '1993-12-31';

练习：查询出1993年出生的员工所有列

SELECT \* FROM emp WHERE birthday>='1993-1-1' AND birthday<='1993-12-31';

SELECT \* FROM emp WHERE birthday BETWEEN '1993-1-1' AND '1993-12-31';

练习：查询出20号部门和30号部门员工所有列

SELECT \* FROM emp WHERE deptId=20 OR deptId=30;

SELECT \* FROM emp WHERE deptId IN(20,30);

练习：查询出不在20号部门和30号部门员工所有列

SELECT \* FROM emp WHERE deptId NOT BETWEEN 20 AND 30;

SELECT \* FROM emp WHERE deptId NOT IN(20,30);

AND/OR

BETWEEN AND/NOT BETWEEN AND

IS NULL/IS NOT NULL

IN()/NOT IN()

(8)模糊条件查询

示例：查询姓名中含有字母e的员工所有列

SELECT \* FROM emp WHERE ename LIKE '%e%';

练习：查询姓名中以e结尾的员工所有列

SELECT \* FROM emp WHERE ename LIKE '%e';

练习：查询姓名中倒数第2个字符为e的员工所有列

SELECT \* FROM emp WHERE ename LIKE '%e\_';

% 可以匹配任意多个字符 >=0

\_ 可以匹配任意1个字符 =1

以上两个匹配符号必须使用LIKE关键字

(9)分页查询

查询的结构集中有太多的数据，一次显示不完，可以分页显示。

需要两个条件：页面的数据量，当前的页码

任意1页开始的值=(当前的页码-1)\*每页的数据量

SELECT \* FROM emp LIMIT start,count;

start:开始的值

count:每页的数据量

注意：start和count的取值必须是数值，不能使用字符串。

练习：假设每页显示5条记录，分别查询前5页

第1页：SELECT \* FROM emp LIMIT 0,5;

第2页：SELECT \* FROM emp LIMIT 5,5;

第3页：SELECT \* FROM emp LIMIT 10,5;

第4页：SELECT \* FROM emp LIMIT 15,5;

课后任务：

(1)复习内容，删除代码保留注释，重新编写SQL语句

(2)查询出工资在8000以上的女员工的姓名、性别、生日和工资，结果集按照工资的降序排列，取出前3条数据。

SELECT ename,sex,birthday,salary

FROM emp

WHERE salary>=8000 AND sex=0

ORDER BY salary DESC

LIMIT 0,3;

2019-5-7

1.复杂查询

(1)聚合查询/分组查询

示例：查询所有员工的数量

SELECT COUNT(eid) FROM emp;

练习：使用所属部门编号计算员工数量

SELECT COUNT(deptId) FROM emp;

练习：使用员工姓名计算员工数量

SELECT COUNT(ename) FROM emp;

SELECT COUNT(\*) FROM emp;

\*\*聚合函数

\*\*函数是一个功能体，提供若干个数据，产出结果——饺子机

\*\*COUNT()/SUM()/MAX()/MIN()/AVG()

练习：查询出所有员工工资的总和

SELECT SUM(salary) FROM emp;

练习：查询出所有男员工的平均工资

SELECT AVG(salary) FROM emp WHERE sex=1;

SELECT SUM(salary)/COUNT(\*) AS 'avg(1)' FROM emp WHERE sex=1;

练习：查询出所有女员工最高工资

SELECT MAX(salary) FROM emp WHERE sex=0;

练习：查询出所有员工的最低工资

SELECT MIN(salary) FROM emp WHERE sex=0;

\*\*分组查询只能查询聚合函数和分组条件

示例：查询出每个部门的员工数量是多少

SELECT deptId,COUNT(\*) FROM emp GROUP BY deptId;

练习：查询出男女员工的平均工资、最低工资、最高工资

SELECT sex,AVG(salary),MAX(salary),MIN(salary) FROM emp GROUP BY sex;

\*\*year() 获取日期中的年份

\*\*month() 获取日期中的月份

示例：查询出1991年出生的员工

SELECT \* FROM emp WHERE YEAR(birthday)=1991;

练习：查询出3月份出生的员工

SELECT \* FROM emp WHERE MONTH(birthday)=3;

(2)子查询

把一个SQL语句的查询结构作为另一个SQL语句的查询条件

示例：查询出研发部员工所有的列

步骤1：查询出研发部的部门编号——10

SELECT did FROM dept WHERE dname='研发部';

步骤2：查询出10号部门员工所有列

SELECT \* FROM emp WHERE deptId=10;

综合：

SELECT \* FROM emp WHERE deptId=(SELECT did FROM dept WHERE dname='研发部');

练习：查询出比Tom工资高的员工的所有列

SELECT \* FROM emp WHERE salary>(SELECT salary FROM emp WHERE ename='Tom');

练习：查询出和tom同一年出生的员工有哪些

SELECT \* FROM emp WHERE YEAR(birthday)=(SELECT YEAR(birthday) FROM emp WHERE ename='tom');

(3)多表查询

示例：查询出所有的员工及其部门名称

SELECT did,ename,dname FROM dept,emp;

错误：笛卡尔积

多表查询避免产生笛卡尔积，给查询添加条件

SELECT did,ename,dname FROM dept,emp WHERE did=deptId;

以上查询方式无法查询出没有部门的员工，也无法查询出没有员工的部门，这是SQL-92中的多表查询语法；在SQL-99中提出了新的查询语法

(1)内连接 INNER JOIN...ON... 和SQL92的结果一样

SELECT ename,dname FROM emp INNER JOIN dept ON deptId=did;

(2)左外连接 LEFT OUTER JOIN...ON...

SELECT ename,dname FROM emp LEFT OUTER JOIN dept ON deptId=did;

查询的结果是左侧表中所有的记录都显示，OUTER可以省略

(3)右外连接 RIGHT OUTER JOIN...ON...

SELECT ename,dname FROM emp RIGHT OUTER JOIN dept ON deptId=did;

查询的结果是右侧表中所有记录都显示，OUTER可以省略

(4)全连接 FULL JOIN...ON...

显示左侧和右侧表中所有记录，MySQL不支持

UNION 合并相同的项

UNION ALL 不合并相同的项

(SELECT ename FROM emp\_us)

UNION

(SELECT ename FROM emp\_cn);

全连接

(SELECT ename,dname FROM emp LEFT OUTER JOIN dept ON deptId=did)

UNION

(SELECT ename,dname FROM emp RIGHT OUTER JOIN dept ON deptId=did);