目录

[一、数据库的基本概念 2](#_Toc137330393)

[二、MySQL软件的介绍 7](#_Toc137330394)

[三、MySQL语法-------不区分大小写 9](#_Toc137330395)

[1、基础语法 9](#_Toc137330396)

[2、SQL语句的介绍 10](#_Toc137330397)

[3、完整性约束--------针对表格 14](#_Toc137330398)

[4、MySQL的数据类型 20](#_Toc137330399)

[5、函数 22](#_Toc137330400)

[6、单表查询--------“筛选” 25](#_Toc137330401)

[7、多表查询 26](#_Toc137330402)

[8、子查询（嵌套查询） 30](#_Toc137330403)

[9、事物 34](#_Toc137330404)

[10、视图 36](#_Toc137330405)

[11、自定义函数创建 38](#_Toc137330406)

一、数据库的基本概念

数据（data）：指对客观事物进行描述并可以鉴别的符号。这些符号是可识别的，抽象的。

比如数字、图片、音频等。

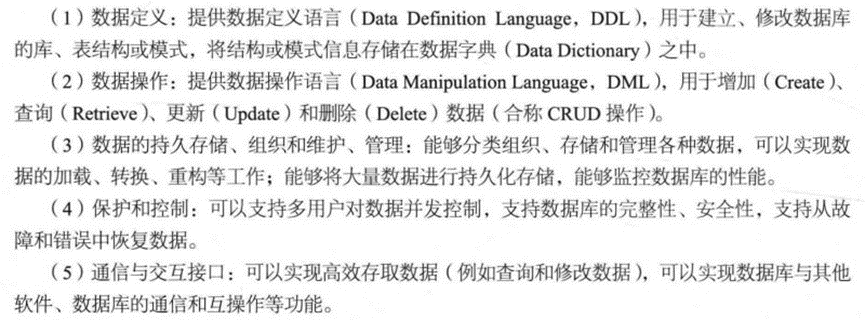
数据库管理系统（DBMS）：数据库极其管理它的软件组成。

数据库系统（DBS）：数据库管理系统以及软硬件、操作人员等其它系统组成。

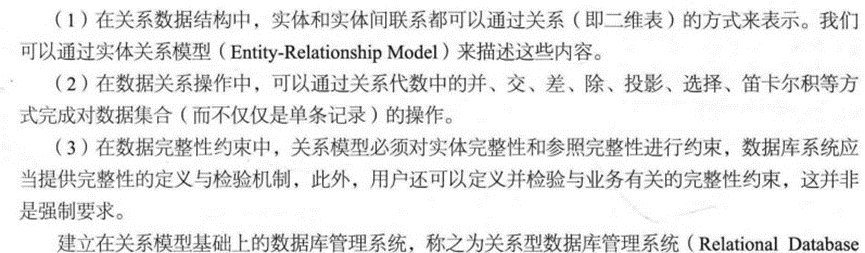
数据库的分类：NoSQL（非关系型数据库） RDBMS（关系型数据库）

数据库应用系统（DBAS）：数据库管理系统下支撑建立的计算机应用系统。

\*数据库管理系统的功能：



\*关系型数据库



\*事务（Transation）：一组数据操作必须组成一个整体，要么全部完成，要么全部取消。

\*事物的特性：原子性Atomicity、一致性Consistency、隔离性Isolation、持久性Durability

\*ACID特性

（1）原子性：事物是一个不可再分的逻辑执行体。（要么全部完成，要么都不完成）

（2）一致性：事务必须使数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态。

所谓一致性是数据库处理前后结果应与其所抽象的客观世界中真实状况保持一致。这种一致性是一种需要管理员去定义的规则。管理员如何指定规则，数据库就严格按照这种规则去处理数据。

（3）隔离性：多个事物执行互不干扰。

（4）持久性：事物执行完后的结果永远保持不变。

\*事物并发（多个事物一起操作）的问题

（1）脏读（dirty read）

图形用户界面, 应用程序

描述已自动生成（2）不可重复读（unrepeatable read）



（3）幻象读（phantom read）

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

横向（水平）扩展：增加硬件的数量来保证计算机系统的数据运算处理能力。

纵向（垂直）扩展：升级更换计算机里的CPU，硬盘，内存等，提高计算机的数据运算能力。



NoSQL数据库一般会弱化关系，即弱化模式、表结构、完整性约束，取消事务机制，目的去掉关系模式的约束实现强大的分布式部署能力（分区容错性、伸缩性、可用性）

\*NoSQL的应用：

（1）海量的日志数据、业务数据或监控数据的查询和管理

（2）特殊或复杂的数据模型简化处理

（3）作为数据仓库、数据挖掘系统或OLAP系统的后台数据支撑

\*其它的非关系型数据库：Cassandra、MongoDB、Neo4j、HBase（Hadoop分布式数据库）

\*常见的关系型数据库：Oracle（甲骨文）、DB2、SQL server、MySQL（开源免费-----现在除外）

\*SQL语言分为：

（1）数据查询语言（DQL）：用于对数据的查询

（2）数据操作语言（DML）：用于对数据库中的数据的修改，删除等操作

（3）数据定义语言（DDL）：针对与数据库对象（索引、列表、数据库、函数和图像等）

（4）数据控制语言（DCL）：赋予和回收权限

（5）事务控制语言（TCL）：对数据库的事物管理

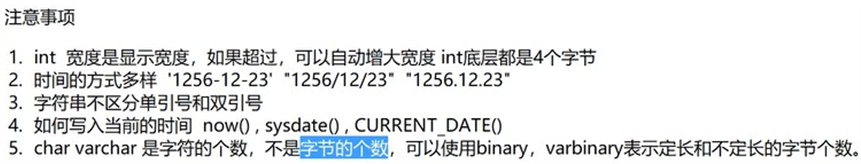
\*在关系型数据库中，数据库中表中的行被称为记录，表中的列被称为是这些记录的字段。

关系型数据库===（表格）行列

\*数据库列表的类型--------整数类型、浮点类型、字符串类型、日期类型（见笔记）

score double(4,1)小数部分为1位，总宽度4位

\*表格插入数据时的操作



\*表格删除数据时的操作

文本, 信件

描述已自动生成

\*主键（primary key）的完整称呼是“主键约束”，是 MySQL 中使用最为频繁的约束。一般情况下，为了便于 DBMS 更快的查找到表中的记录，都会在表中设置一个主键。

主键又分为单字段主键和多字段联合主键（复合主键------多个单字段组合为一个主键）。

（1）每个表中只能定义一个主键。

（2）主键值必须唯一标识表中的每一行，且不能为 NULL，即表中不可能存在有相同主键值的两行数据。这是唯一性原则。

（3）一个字段名只能在联合主键字段表中出现一次。

（4）联合主键不能包含不必要的多余字段。当把联合主键的某一字段删除后，如果剩下的字段构成的主键仍然满足唯一性原则，那么这个联合主键是不正确的。这是最小化原则。

\*索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定信息。（以空间换时间）索引往往是存储在磁盘上的文件中的----------索引太占空间



//关系型数据库的约束

（1）约束实际上就是表中数据的限制条件

（2）约束目的就是为了保证表中的记录完整性和有效性，比如用户表有些列的值（手机号）不能为空，有些列的值（身份证号）不能重复。

（3）约束分类：

主键约束(primary key) PK

自增长约束(auto\_increment)

非空约束(not null)

唯一性约束(unique)

默认约束(default)

零填充约束(zerofill)

外键约束(foreign key) FK

//多个表之间的查询-----------连接查询

图示

描述已自动生成

自连接表一定要起别名，对于自连接查询，可以是内连接查询，也可以是外连接查询。

注意：这是表格查询，查询后的结果是两个表的，不是一个表的。

比如student表里的sno和class里的id做自然连接查询，最终结果会显示sno和id的

这是表格，不是集合。

自动连接默认匹配所有，内连接可以有选择性的匹配。

全外连接：展示左右表全部不匹配信息。

交集：两个表格所有内容组合在一起

并集（UNION）：两个表格相同的部分

差集：A表格减去与B表格相同的部分

MySQL没有交集、并集和差集功能，甲骨文有。

//笛卡尔积

笛卡尔积是指在数学中，两个集合X和Y的笛卡尓积（Cartesian product），又称直积，表示为X × Y，第一个对象是X的成员而第二个对象是Y的所有可能有序对的其中一个成员。

简单的说就是两个集合相乘的结果。

假设集合A={a, b}，集合B={0, 1, 2}，则两个集合的笛卡尔积为{(a, 0), (a, 1), (a, 2), (b, 0), (b, 1), (b, 2)}。

二、MySQL软件的介绍

现在MySQL已被甲骨文收购

MySQL优势：（社区版免费）

（1）开放源代码

（2）跨平台

（3）轻量级

（4）成本低

MySQL的官网：



MySQL分为服务端、远程连接、客户端等。初学者只需要安装服务端。

//在Windows系统中的MySQL

（1）程序运行文件的位置：C:\Program Files\MySQL。

（2）数据文件夹，存放MySQL的基础数据和以后新增的数据：C:\ProgramData\MySQL

（3）MySQL的配置文件：C:\ProgramData\MySQL\MySQL Server 8.0\my.ini（端口号、基础文件夹）

//Windows的MySQL启动

第一步、写入环境变量配置。

第二步、在cmd窗口中输入命令



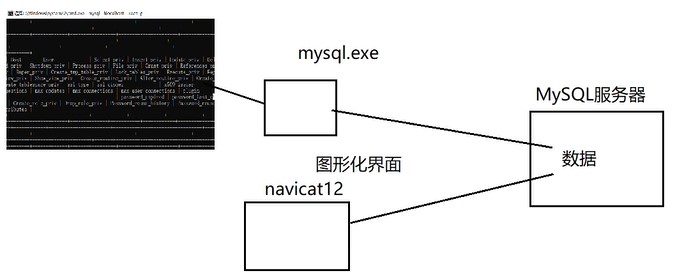
（localhost服务端的位置，可以用127.0.0.1------环回地址代替，本机访问可以默认不加h参数）

MySQL数据存储结构：库>>>表>>>数据

//Windows里的MySQL的卸载

关闭端口-----点击卸载程序------删除文件残留------删除环境变量

//图形客户端



数据库管理人员（DBA）可以利用SQL通过数据库管理系统（DBMS）对数据库（DBM）里的数据（DB）进行管理。

SQL是一种非过程话语言。

SQL是IBM公司研发的。

我们学的SQL语言是99版的。

\*navicat12的启动错误：

不同版本的MySQL加密方式不同。

文本

描述已自动生成

MySQL里的字符集“utf-8”其实是“utf8mb4”。

\*navicat12的使用

（1）注释

单行注释： 多行注释：

文本

描述已自动生成

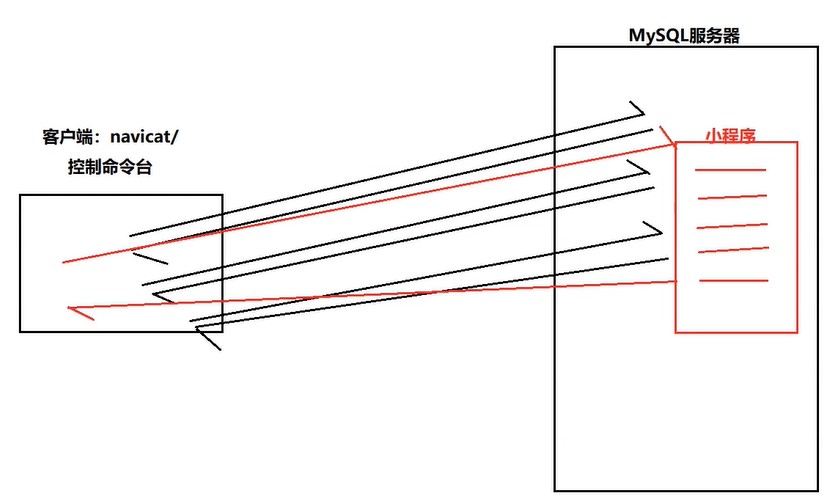
\*在cmd窗口中输入关闭MySQL命令中出现情况：“发生系统错误 5。 拒绝访问。”

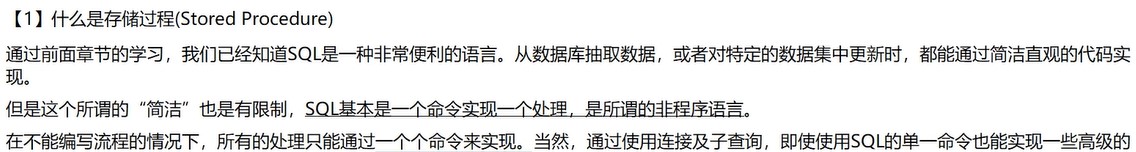
给对应的文件赋予管理员权限（C:\Windows\System32\cmd.exe）32指的是操作系统的位数

或者以管理员的方式启动cmd窗口。

\*SQL语法的网站：https://www.runoob.com/sql/sql-alias.html

\*MySQL的存储过程





我们可以通过函数代替许多的查询语句，以达到我们的要求-------------------脚本、自定义函数

优点：

（1）提高执行性能 （2）减轻网络负担 （3）可将数据库的处理黑匣子化

三、MySQL语法-------不区分大小写

1、基础语法

mysql -h【主机名】 -u【用户名】 -p //登录MySQL

exit或quit; //退出MySQL

show database; //查看MySQL下的所有数据库

use 【数据库名】; //进入数据库

show tables; //查看数据库下的所有表名

\*MySQL的启动和关闭

（1）Windows系统

NET START MYSQL80 net start mysql80 //mysql80是MySQL服务的名字

NET STOP MYSQL80 net stop mysql80

（2）Linux系统

service mysql start

service mysql stop

service mysql restart //重启MySQL服务

\*Windows系统下的MySQL使用命令（了解）

create user 'user01' @'localhost' identified by 'user01'; //创建用户“identifie鉴定”

drop user ‘user01’@’localhost’; //删除用户

select user,host from mysql.user; //查看所有用户

select current\_user(); select user(); //查看当前用户

rename user 'user01'@'localhost' to 'User01'@'localhost'; //将用户名user01 改为User01

alter user 'User01'@'localhost' identified by 'User01'; //将密码user01改为User01

show grants for ‘User01’@’localhost’; //查看当前登录用户所属的条件

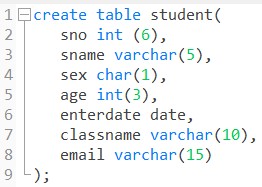
grant all on test.score to 'User01'@'localhost'; //这里的all是指将test库里的score表的所有权限都给User01

revoke all on test.score from 'User01'@'localhost'; //撤销权限

flush privileges; //刷新系统权限表，即时生效

2、SQL语句的介绍

\*创建表



文本

描述已自动生成

describe student; desc student //显示详细的表的信息结构

select \* from student; //显示表中所有的信息

show create table student; //显示创建该表的SQL语句

drop table student; //删除整张表

truncate table student; //删除表

delete from user; //删除表中的记录

delete from user where user\_id = 1; //有条件的删除表中的记录

insert into student (id,name,age) values(null,'zhangsan',23); //在表中插入数据

select name,id from student where id=20201304023; //有条件的查询表中的内容

update student set sex = ‘男’ where sno = ‘20201304023’; //有条件地在表格修改数据

update student set sex = ‘男’; //把表格中选定的整列数据都给替换了

alter table 表名 rename to 新表名; //修改表名

alter table 表名 drop column 列名; //删除表中的具体列

alter table 表名 change 列名 新列名 varchar(30); //修改表中列的名字

alter table 表名 modify 列名 varchar(22); //修改表中具体列的属性

alter table 表名 add 字段名 字段类型 first; //将新增加的列设定为第一列

alter table 表名 add 字段名 字段类型 after 字段名; //将新增加的列设置在某个列的后面（字段名：列名 字段属性：列里面存储的数据类型）

show columns from 表名; //查看表中所有的列名

alter table student engine= myisam; //修改数据表的类型，将其表的类型修改为myisam

create database 数据库名; //创建数据库

show databases; //查看所有的数据库

drop database 数据库名; //删除具体的数据库

//快速添加一张表，使其与已存在的表student数据和结构一致

create table student2 as select \* from student;

//快速添加一张表，使其与已存在的表student结构一致，但里面没有任何数据

create table student3 as select \* from student where 1 = 2;

//快速添加一张表，只要部分列、部分数据

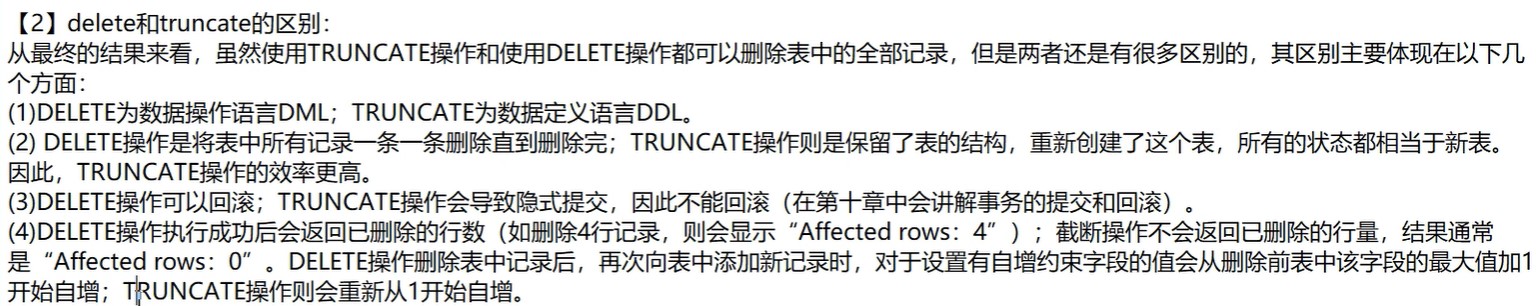
create table student3 as select sno,sname,age from student where sno =2;

//删除数据

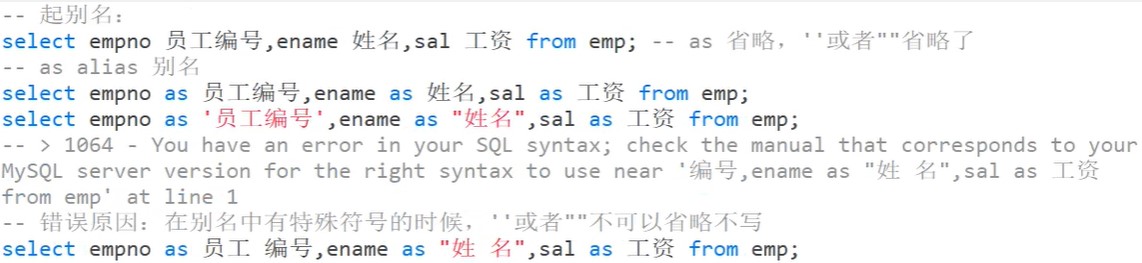
（1）drop删除整张表

（2）delete只删除表中的数据，保留表中的结构

（3）truncate只删除表中的数据，保留表中的结构



//起别名的注意事项“AS”



select column\_name(s) from table\_name AS alias\_name; //给表格起别名

SELECT column\_name AS alias\_name FROM table\_name; //给列起别名

\*使用别名的好处

（1）在查询中涉及超过一个表

（2）在查询中使用了函数

（3）列名称很长或者可读性差

（4）需要把两个列或者多个列结合在一起

\*给表取别名的语法

SELECT w.name, w.url, a.count, a.date

FROM Websites AS w, access\_log AS a

WHERE a.site\_id=w.id and w.name="菜鸟教程"; //起了别名

SELECT Websites.name, Websites.url, access\_log.count, access\_log.date

FROM Websites, access\_log

WHERE Websites.id=access\_log.site\_id and Websites.name="菜鸟教程"; //没起别名的

（表：Websites和access\_log）

（表：Websites里的列为name,url）

（表：access\_log里的列为count,date）

//有条件的查询



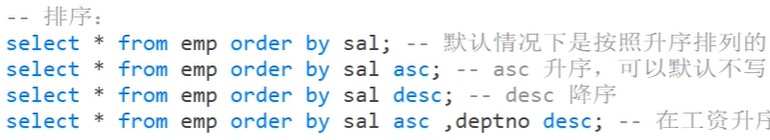
//去重查询

表格中的有些列里的元素重复了，想要显示该列中的所有元素



（对多个列进行去重查询，这些列组成的数组是不相同的。） distinct不同的，区别

//排序



（先对sal进行升序排序。如果某几行的sal值相同，则会按deptno进行降序排序。）

//有条件的查询“where”

（1）where +“算数运算符”

图片包含 表格

描述已自动生成

（2）where + “逻辑运算符”（与、或、非---------“and”“or”“!”）

文本

低可信度描述已自动生成

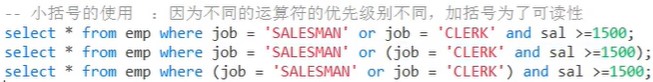
（3）模糊查询--------（“%”代表任意字符，“\_”代表单个字符）



（4）非空判断



（5）小括号-------划分判断语句的优先级

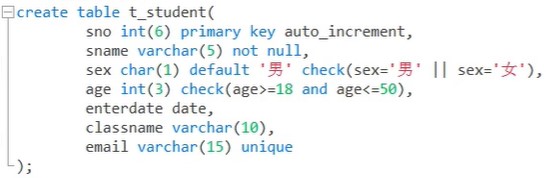


（SQL解析器在处理操作时会优先处理and操作，在处理or）

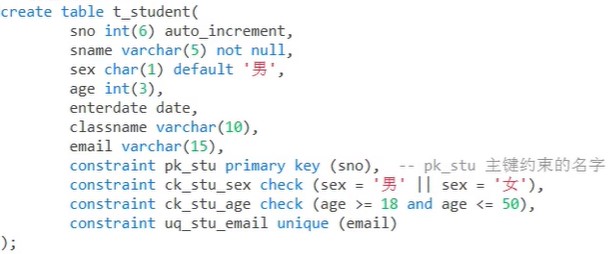
3、完整性约束--------针对表格

3.1、完整性约束的用法

（1）列级约束（针对某一个特定的列）



（2）表级约束（用于对多个列一起的约束）

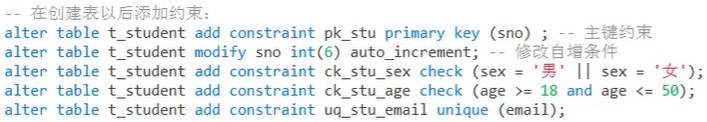


\*注意

表级约束：可以约束表中任意一个或多个字段。与列定义相互独立，不包含在列定义中；与定义用‘，’分隔；必须指出要约束的列的名称；

列级约束：包含在列定义中，直接跟在该列的其它定义之后 ，用空格分隔；不必指定列名；

（3）在创建表后添加约束



表格

描述已自动生成

（4）外键约束--------------------（难）

//有些表格之间有相同的列，当我们修改了其中一列时，其它的表要挨着修改吗？

我们可以引入外键约束，用来在两个表数据之间建立链接，其中一张表的一个字段被另一张表中对应的字段约束。也就是说，设置外键约束至少要有两种表，被约束的表叫做从表（子表），另一张叫做主表（父表），属于主从关系。

//其中主表中的关联字段叫做主键，子表中的关联字段叫做外键。

//外键约束主要作用就是能够让表与表之间的数据建立关联，使数据更加完整，关联性更强。

//外键创建规则

（1）必须有主表才可以设置从表。

（2）主表必须实际存在。

（3）必须为主表定义主键。

（4）外键列的数据类型必须和主键列的数据类型相同。

（5）外键列的数量必须和主键列的数量相同。

（6）外键可以不是外表中的主键，但必须和主表关联字段相对应。

//语法（创建外键约束的）

第一步、建立主从表（划分外键，主键）-------加约束

文本

低可信度描述已自动生成

class（主表） constraint 约束 reference提及、谈到

图形用户界面, 文本

描述已自动生成student（从表） fk\_stu\_classno（外键名字）

第二步、设置外键约束



//单充建立外键，不加外键约束，可以吗？

从表添加一条主表不存在的数据成功了？

主表删除数据，从表对应的那个数据还存在？

答案：不可以

外键约束只有表级约束，没有列级约束。

先删主表，在删从表。

//参数说明：

constraint ：用于设置外键约束名称（可以省略）

foreign key：外键设置，用于指定外键字段

references：主表及主键设置，用于指定主表和主键

//属性说明：

cascade：主表删除或修改记录时，从表也会对关联记录的外键字段进行修改。

restrict：删除或修改主表记录，子表中若有关联记录，则不允许主表删除或修改。

set null：主表删除或修改主表记录时，从表会将关联记录的外键字段设为null。

on update cascade：主表修改记录时，从表关联记录的外键字段也会修改。（将cascade改为restrict，意思相反）

on delete cascade：主表删除记录时，从表关联记录的外键字段也会删除。（将cascade改为restrict，意思相反）

//语法

constraint 外键名称foreign key 外键字段references class 主键字段

on update cascade on delete cascade //外键约束创建（在表格里写该语法）

alter table 表名 drop foreign key 外键名称; //删除外键（在表格外面写的）

//主从表之间的关系--------------当约束条件为（on update cascade on delete cascade）

（1）从表中删除记录不会对主表造成影响。（从删主不删）

（2）主表的修改根据属性会对从表造成影响。（主删从不见）

（3）从表的修改不影响主表，但主表的修改会影响从表

（4）无法插入主键关联中不存在的记录。

cascade（v.）串联，倾斜，传授，垂下

（n.）瀑布

restrict （v.）限制，控制

//级联操作

on delete cascade:主表中的数据被删，从表中相关数据也会被删

on update cascade:主表中的数据被改后，从表中的相关数据也会被修改

on delete cascade on update cascade:删除和修改有级联操作

on delete set null:主表总数据被删除后，从表中相关数据为null

on update set null：

\*外键策略

在外键约束的情况下，如何合理的删除外键

//不允许操作

手机屏幕截图

描述已自动生成

//级联操作

图形用户界面, 文本, 应用程序

描述已自动生成

//置空操作

图片包含 文本

描述已自动生成

3.2、完整性约束常见的错误

（1）列级约束

文本

中度可信度描述已自动生成

（当我们插入数据时发生报错，插入的数据会丢失，但是引用的主键却不会丢失而会保存）

（2）表级约束



（3）外键约束

//单充建立外键，不加外键约束，可以吗？

从表添加一条主表不存在的数据成功了？

主表删除数据，从表对应的那个数据还存在？

答案：不可以

3.3、约束的分类

primary 私有的 union 单独的 check检查 increment自增的

default默认 foreign外部的

//主键约束(primary key) PK 主键的定义

//自增长约束(auto\_increment) 只能搭配主键，键值（整型）自增加一

//非空约束(not null) 键值不能为空

//唯一性约束(unique) 键值必须是唯一的

//默认约束(default) 键值是默认的

//零填充约束(zerofill) 插入数据时，当该字段的值的长度小于定义的长度时，会在该值的前面补上相应的0

//外键约束(foreign key) FK

补充：

列级约束：只能应用于一列上。

列级约束：包含在列定义中，直接跟在该列的其它定义之后 ，用空格分隔；不必指定列名

表级约束：可以应用于一列上，也可以应用在一个表中的多个列上。

表级约束：与列定义相互独立，不包含在列定义中；与定义用‘，’分隔；必须指出要约束的列的名称

（1）外键约束只有表级约束，没有列级约束。

（2）列级约束有六种：主键、外键、唯一、检查、默认、非空/空值

（3）表级约束有四种：主键、外键、唯一、检查

有constraint的是列级约束，没有constraint是表级约束

4、MySQL的数据类型

表中的列有4种数据类型：整型、浮点数型、字符串类型、日期和时间类型。

//补充

一个字节=8个比特位

一个字符占一个字节

1个二进制的计数位为1个比特位 8个比特位 （00000000 11111111）

无符号（0 2\*8-1） 有符号（-2\*8 2\*8-1）

字符（byte） 比特（bit）

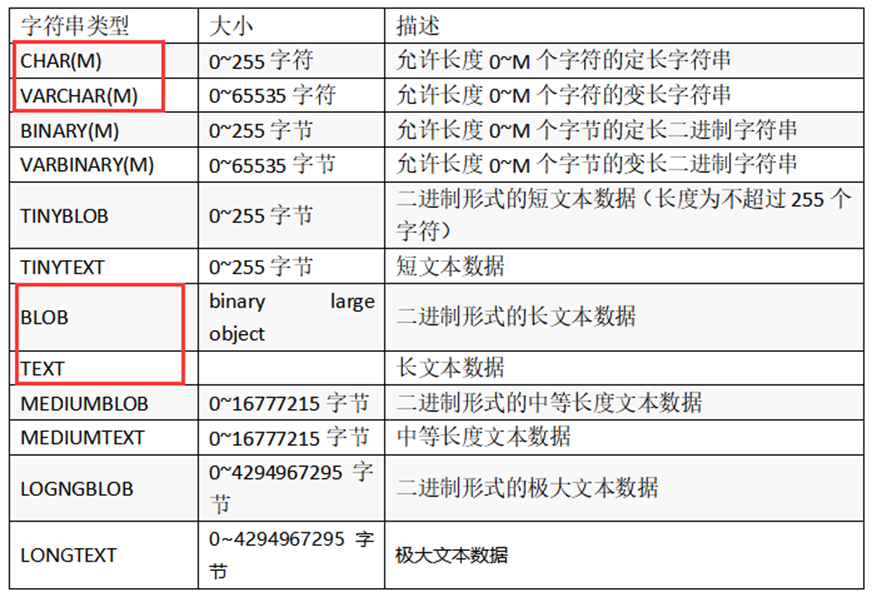
//整型



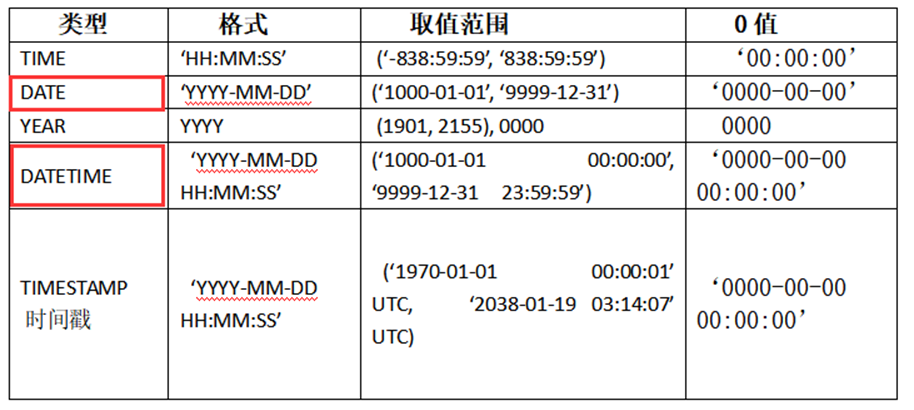
//浮点型



//字符串类型



//日期和时间类型



5、函数

\*MySQL里提供了大量函数，这里的函数相当于功能模块（封装了一些SQL语句），简化了SQL的使用。（Python里的数据库，java里的封装）



//函数分类

单行函数：对单个数据运算，返回一个结果

多行函数（分组函数）：对一组数据运算，返回一个结果

除了多行函数（sum、count、avg、max、min），剩下的都是单行函数

5.1、单行函数



（如果没有where条件话，from dual可以忽略不写）

abs( )绝对值 celi( )向上取整 floor( )向下取整

round( )四舍五入 mod( )取模函数-------取余数



curdate( )年月日 curtime( )时分秒



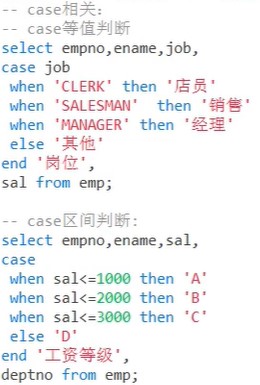
sleep( ) 使计算机程序（进程，任务或线程）进入休眠，使其在一段时间内处于非活动状态。---------C语言里的sleep函数

now( ) 返回当前日期和时间 sysdate( )返回函数执行的日期和时间

\*注意：

SQL语句插入数据时，now( )函数插入的结果最终要受限与对应的表的列的数据类型。





5、JOSN函数（没讲）



（查看当前使用的数据库名称，用户名称，对应的MySQL版本）

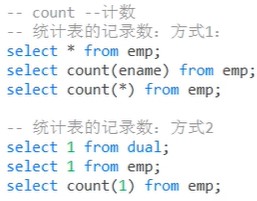
7、lower函数

将所属的字符段转换为小写。

5.2、多行函数

图形用户界面, 文本, 应用程序

描述已自动生成



\*为什么不用列来统计表的记录数？

比如有些列它的一些行的值为空，用count函数计算结果肯定不正确。

“dual”你可以在没有表的情况下指定一个虚拟的表名

count 数量，合计

5.3、分组和筛选

（1）分组

查询一个部门的平均薪资，但这个一个公司有好多部分。如何准确查找不同部门的平均薪资呐？-----------分组“group by”

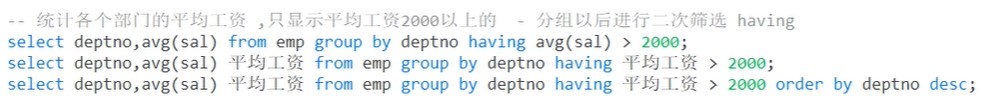
把要分组的列按照其里面的值均分为好几组。



文本, 信件

描述已自动生成

（2）having分组后进行筛选



图形用户界面, 文本

描述已自动生成

6、单表查询--------“筛选”

文本

描述已自动生成

\*查询的大概流程

给查询的具体的列（给列加筛选条件-----“函数”）

where给查询的结果套上相应的要求。

对最终的查询结果进行分组、排序

7、多表查询

7.1、内连接、交叉连接、自然连接----------“SQL99”

文本

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图片包含 文本

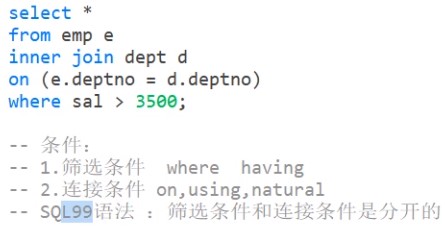
描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

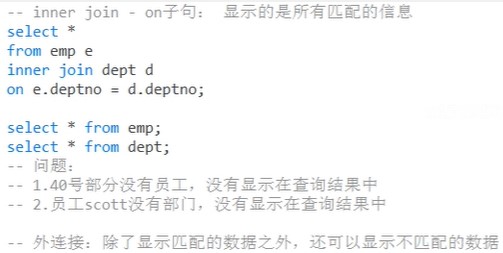
描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成



7.2、外连接（左外连接、右外连接、全连接）



图形用户界面, 文本, 应用程序

描述已自动生成

文本

描述已自动生成

文本

描述已自动生成

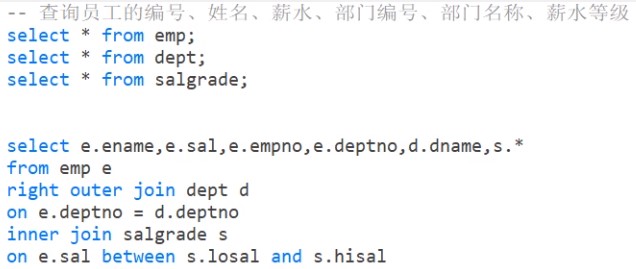


7.3、多表查询补充事项

（1）内连接不加任何条件（on语句筛选的）就是交叉连接（笛卡尔乘积）

（2）内连接写上该语句也是交叉连接（“on 1 = 1”）

（3）多表查询可以简化为两张表之间进行查询



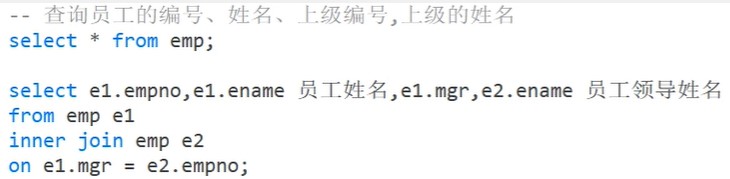
7.4、自连接--------自关联

（1）什么是自关联

图示, 表格

中度可信度描述已自动生成

（2）具体应用



7.5、SQL92语法与SQL99语法的区别

（1）92语法的多表查询“了解”

手机屏幕截图

描述已自动生成





（2）99语法与92语法在多表查询存在的区别

男子的脸部特写与配字黑白照

中度可信度描述已自动生成

一些文字和图片的手机截图

描述已自动生成

8、子查询（嵌套查询）

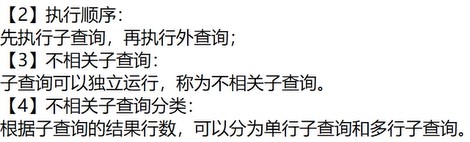
定义：是指将一个 SELECT 查询（子查询）的结果作为另一个 SQL 语句（主查询）的数据来源或者判断条件。

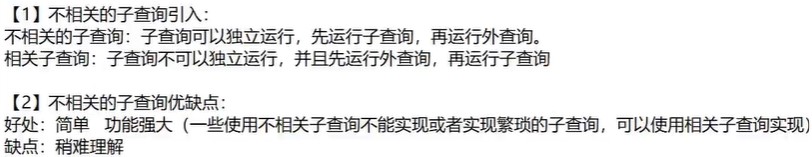
图表

描述已自动生成



子查询又分为相关子查询和不相关子查询





（1）单行子查询--------子查询结果只有一个（不相关子查询）



（2）多行子查询--------子查询结果有多个（不相关子查询）

\*单行查询遇到的问题：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

\*修改

图形用户界面

中度可信度描述已自动生成



\*例子

图形用户界面, 文本, 应用程序

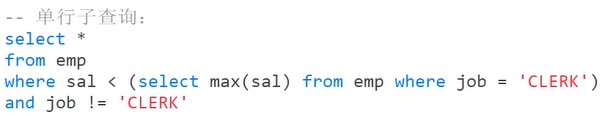
描述已自动生成

图片包含 文本

描述已自动生成

图形用户界面, 应用程序

描述已自动生成



（3）相关子查询

相关子查询的运行顺序：先运行外查询，在运行子查询

相关子查询与不相关子查询进行比较：

图形用户界面, 文本

中度可信度描述已自动生成

（在相关子查询中，里面子查询是不能独立运行的）



\*例题

图形用户界面, 文本, 应用程序

描述已自动生成

（有些岗位只有一个员工）

9、事物

9.1、事物及其特征

事物机制的应用：淘宝订单交易，微信转账等。

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

9.2、事物并发问题的解决方法------------“隔离”

（1）解决不可重复读--------锁住当前操作的那一行

（2）解决幻读-------锁住当前操作的表格

为了解决事物并发出现的问题（脏读、幻读、不可重复读），我们可以引入隔离机制

表格

描述已自动生成

表格, 日历

描述已自动生成

\*MySQL默认使用的是REPEATABLER READ（）

文本

描述已自动生成

read uncommitted读未提交

read committed读提交

repeatable read可重复读取

serializable可序化

10、视图

文本

描述已自动生成

视图--------筛子---------过滤-------筛选想要的信息

数据库只存放了视图对应的SQL语句。

视图是一个虚拟的表，本质是一个虚拟的SQL命令集合。

（1）创建单表视图

图形用户界面, 文本, 应用程序

描述已自动生成

（虽然视图里没有30的数据，但原表里插入这个30的数据了）

\*为了避免插入一个不是“deptno = 20”的数据应该怎么办？-----------加入判断条件，判断是不是属于20的，属于的话就插入，不是的话就丢弃

图形用户界面, 文本, 应用程序

描述已自动生成

replace--------替换以创建的视图

with check option---------判断条件

（2）创建多表视图

图形用户界面, 文本, 应用程序

描述已自动生成

（3）创建统计视图

图形用户界面, 文本, 应用程序

描述已自动生成

（4）创建基于视图的视图

图形用户界面, 文本, 应用程序

描述已自动生成

11、自定义函数创建

图形用户界面, 文本, 应用程序

描述已自动生成

（通配符--------Linux中的---------差不多）

文本

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本

中度可信度描述已自动生成