

# SQL入门

# 本章知识点目标



- 知识点1：数据库基础知识
- 知识点2：SQL简介
- 知识点3：SQL分类
- 知识点4：DDL语句
- 知识点5：DML语句
- 知识点6：DCL语句

# 知识点1 【数据库基础知识】 -1

## • 什么是数据(Data)?

- 数据的定义:描述事物的符号记录。在日常生活中，人们直接用语言来描述事物；在计算机中，为了存储和处理这些事物，就要将事物的特征抽象出来组成一个记录来描述。数据是数据库中存储的基本对象。
- 数据的种类：文字、图形、图象、声音
- 数据的特点：数据与其语义是不可分的

学生档案中的学生记录（李白，男，2000，厦门，计算机系，2018）

语义：学生姓名、性别、出生年、籍贯、所在系别、入学时间

解释：李白是个大学生，2000出生，厦门人，2018年考入计算机系

## • 什么是数据表？

- 关系型数据库最基本的数据存储单元就是数据表，可以把关系型数据库中的每个数据表想象为Excel中的表格，横向为一条数据、纵向为一列数据。
- 数据表每一行被称为一条数据，每一列被称为一个字段，为数据库建立数据表时，通常要指定该表包含多少列，每列的数据类型等信息，无需指定多少行，每次新增一条数据，就会多一行数据，所以说数据表中的行是动态改变的，每行用户保存一条数据。

# 知识点1 【数据库基础知识】 -2



- 什么是数据库(Database)?

- 人们收集并抽取出一个应用所需要的大量数据之后，并将其保存起来以供进一步加工处理，进一步抽取有用信息。
- 数据库的定义：数据库(Database,简称DB)是长期储存在计算机内、有组织的、可共享的大量数据集合

- 数据库管理系统(DBMS)

- 数据库管理系统 (Database Management System, 简称DBMS) 是位于用户与操作系统之间的一层数据管理软件。通常我们提到的数据库实际上是指数据库管理系统。
- DBMS的用途：科学地组织和存储数据、高效地获取和维护数据

# 知识点1 【数据库基础知识】 -3



- 数据库系统(DBS)

数据库系统 (Database System) , 是由数据库及其管理软件组成的系统。

数据库系统是为适应数据处理的需要而发展起来的一种较为理想的数据处理系统, 也是一个为实际可运行的存储、维护和应用系统提供数据的软件系统, 是存储介质、处理对象和管理系统的集合体

- 数据库、数据库管理系统、数据库系统的区别

**数据库系统** (DataBase System) 简称DBS, 包括**数据库** (DataBase) 简称DB、**数据库管理系统** (DataBase Management System) 简称DBMS、应用系统、数据库管理员 (DataBase Administrator) 简称DBA。所以DBS是个大的概念, DB是专门存数据的集合, DBMS是由DBA对DB的查询、更新、删除、修改、备份、恢复等操作的。DBMS用来操纵和管理DB的软件, 用于建立、使用和维护DB。它对DB进行统一的管理和控制, 以保证DB的安全性和完整性, 用户可以通过DBMS访问DB中的数据, DBA也可以通过DBMS进行DB的维护工作, 它可使多个应用程序和用户拥有不同的方法在同时或不同时刻去建立、修改和询问DB (也就是说DBMS可以将控制权发挥到极致 (也就是所说的安全性))。DB是长期存储在计算机内的有组织、可共享的大量的数据集合。它可以供各种用户共享, 具有最小冗余度和较高的数据独立性。



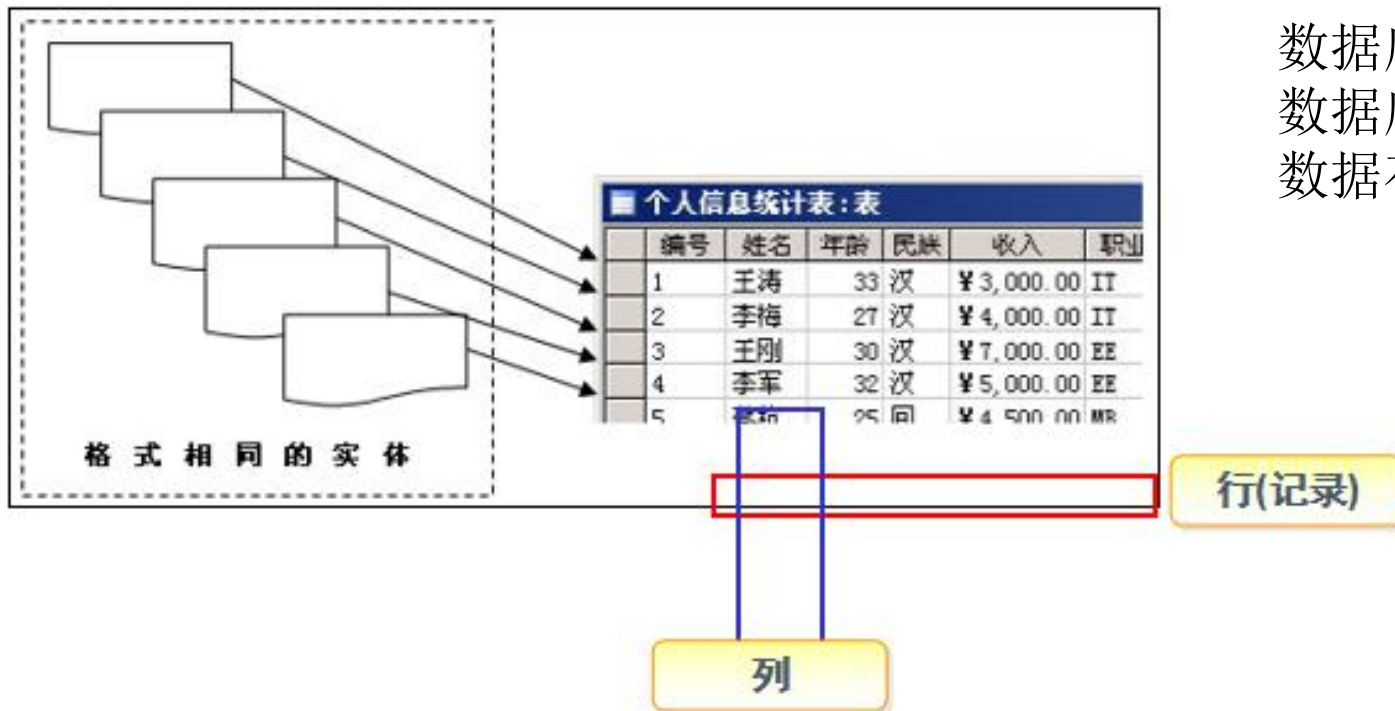
## • 常见的数据库系统

- 目前DBMS的生产商众多，产品也不尽相同，如：
- Oracle公司的Oracle系列；
- Microsoft公司的Access系列和SQL Server系列；
- IBM公司的DB2；
- Sybase公司的Sybase；
- 还有自由开源的MySQL等等。
- NoSQL数据库，Redis，Mongodb等

# 知识点1 【数据库基础知识】 -5

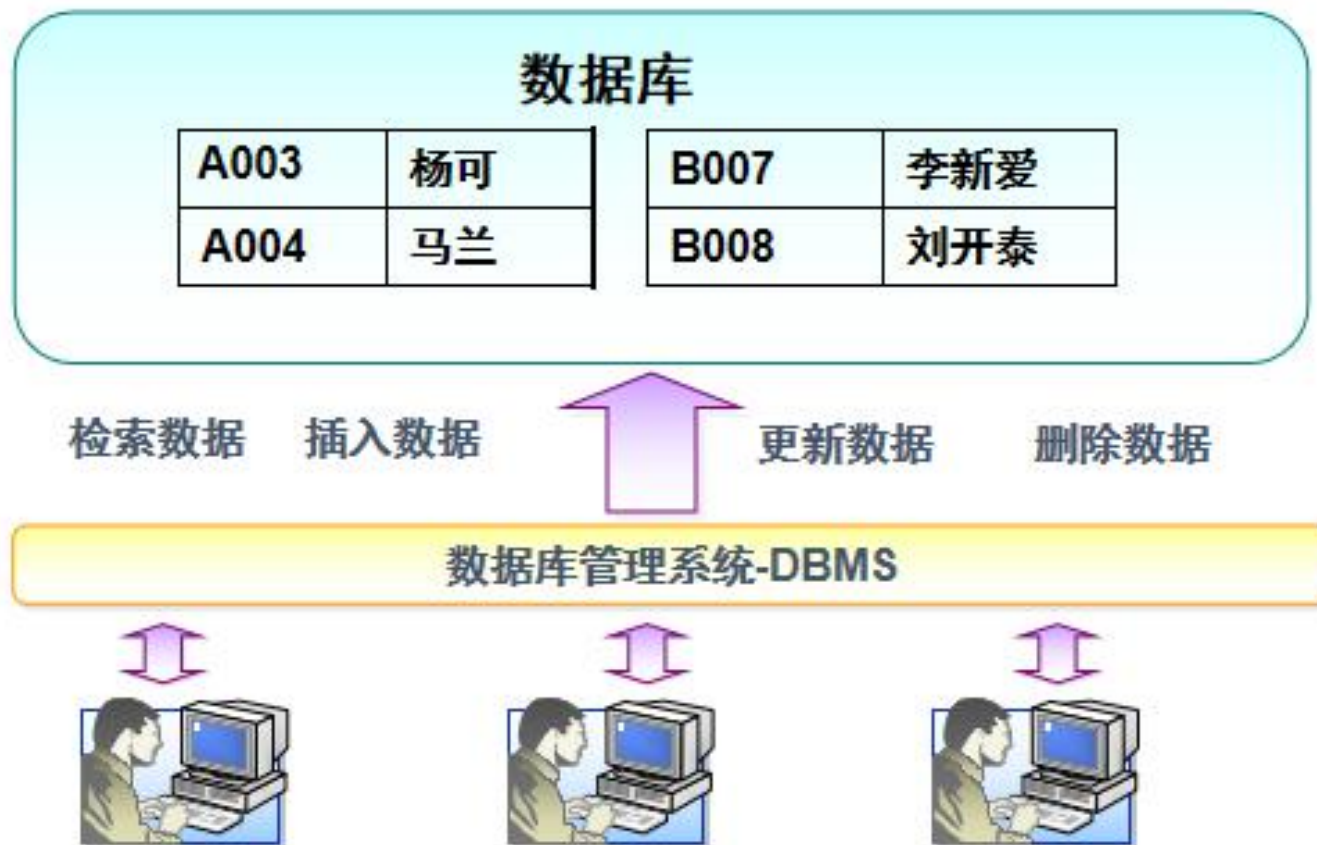


## • 数据库的基本概念



数据库就是“数据”的“仓库”  
数据库由表、关系以及操作对象组成  
数据存放在表中

# 知识点1 【数据库基础知识】 -6





## 知识点2 【SQL简介】 -1



- 什么是SQL?
  - SQL指结构化查询语言 (Structured Query Language)
  - SQL使我们有能力访问数据库
  - SQL是一种 ANSI的标准计算机语言

## 知识点2 【SQL简介】 -2



- SQL能做什么？
  - SQL面向数据库执行查询
  - SQL可在数据库中插入新的记录
  - SQL可更新数据库中的数据
  - SQL可从数据库删除记录
  - SQL可创建新数据库
  - SQL可在数据库中创建新表
  - SQL可在数据库中创建存储过程
  - SQL可在数据库中创建视图
  - SQL可以设置表、存储过程和视图的权限

## 知识点3 【SQL分类】 -1



- SQL语句主要可以划分为以下3个类别：
  - DDL (Data Definition Language) 语句：数据定义语言，这些语句定义了不同的数据段、数据库、表、列、索引等数据库对象。常用的语句关键字主要包括create、drop、alter等。
  - DML (Data Manipulation Language) 语句：数据操纵语句，用于添加、删除、更新和查询数据库记录，并检查数据完整性。常用的语句关键字主要包括insert、delete、update和select等。
  - DCL (Data Control Language) 语句：数据控制语句，用于控制不同数据段直接的许可和访问级别的语句。这些语句定义了数据库、表、字段、用户的访问权限和安全级别。常用的语句关键字主要包括grant、revoke等。

## 知识点4 【DDL语句】 -1

- 数据库模式定义语言DDL(Data Definition Language)，是用于描述数据库中要存储的现实世界实体的语言。主要由create（添加）、alter（修改）、drop（删除）和 truncate（删除）四个关键字完成。它和DML语句的最大区别是：DML只是操作表内部的数据，而不涉及表的定义、结构的修改，更不会涉及到其他对象。
- DDL语句更多地是由数据库管理员（DBA）使用，开发人员一般很少使用。

## 知识点4 【DDL语句】 -2



### • 创建数据库

- 语法: CREATE DATABASE dbname 【default charset utf8 collate utf8\_general\_ci】 ;

- 例: 创建数据库test1, 命令如下:

- create database test1;

```
mysql> create database test1;  
Query OK, 1 row affected (0.24 sec)
```

- 执行结果解析:Query OK, 1 row affected (0.24 sec) 可以分为3个部分

- Query OK: 表示上面的命令执行成功
- 1 row affected: 表示操作只影响了数据库中一行的记录
- 0.24 sec: 记录了操作执行的时间

- 如果已经存在这个数据库, 系统会提示:

```
mysql> create database test1;  
ERROR 1007 (HY000): Can't create database 'test1'; database exists
```

## 知识点4 【DDL语句】 -3



- 查看系统中有哪些数据库

- 语法:show databases

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| sakila     |
| sys        |
| test1      |
| world      |
+-----+
7 rows in set (0.01 sec)
```

- information\_schema: 主要存储系统中的一些数据库对象信息，比如用户表信息、列信息、权限信息、字符集信息、分区信息等。每个用户都可以查看这个数据库，但根据权限的不同看到的内容不同。
- performance\_schema: MySQL5.5引入的系统库，用于存储系统性能相关的动态参数表。
- sys: MySQL5.7引入的系统库，本身不记录系统数据，基于information\_schema和performance\_schema之上，封装了一层更加易于调优和诊断的系统视图。
- mysql: 存储系统的用户权限信息。
- sakila和world: 两个为样例数据库。

## 知识点4 【DDL语句】 -4

- 选择数据库

- 语法: use dbname;

```
mysql> use test1  
Database changed
```

- 查看数据库下所有的数据表

- 语法: show tables;

```
mysql> show tables;  
Empty set (0.00 sec)
```

- 由于test1是刚创建的数据库，还没有表，所以显示为空。命令行下面的“Empty set”表示操作的结果为空。

- 删除数据库

```
mysql> drop database test1;  
Query OK, 0 rows affected (0.15 sec)
```

- 语法: drop database dbname;



## 知识点4 【DDL语句】 -5

### • 创建表

- 基本语法:

create table 表名(

列名 列的数据类型(长度) 列的约束,

列名 列的数据类型(长度) 列的约束,

列名 列的数据类型(长度) 列的约束,

列名 列的数据类型(长度) 列的约束

)

- 例1: 创建一个名称为emp的表。表中包括ename (姓名)、hiredate (雇用日期)、sal (薪水) 和deptno (部门编号) 4个字段。
- 例2: 创建一个名称为user的用户表。表中有用户id、用户名、密码、手机号、邮箱、角色状态、余额、年龄、注册时间、头像等字段。





## 知识点4 【DDL语句】 -6



### • 查看表结构

- 基本语法: `desc tablename;`

```
mysql> desc emp;
```

Field	Type	Null	Key	Default	Extra
ename	varchar(10)	NO	PRI	NULL	
hiredate	date	YES		NULL	
sal	decimal(10,2)	YES		NULL	
deptno	int(2)	YES		NULL	

```
4 rows in set (0.00 sec)
```

### • 使用create复制表

- 基本语法: `create table 复制出的表 LIKE 原表` //只复制表结构及约束, 但不复制数据
- 基本语法: `create table 复制出的表 as select * from 原表` //复制表结构及数据, 但不复制约束

## 知识点4 【DDL语句】 -7



- drop 删除数据库

- 基本语法: drop database 数据库名;(之前学习)

- drop 删除表

- 基本语法: drop table 表名;

- truncate 删除表内全部数据, 删除大量数据时建议使用

- 基本语法: drop table 表名;

- delete 删除表内数据

- 基本语法: delete from 表名 where.....; (后面学习)

## 知识点4 【DDL语句】-约束



- 约束的分类

- 约束是一种限制，它通过对表的行或列的数据做出限制，来确保表的数据的完整性、唯一性。
- MYSQL中，常用的几种约束：

约束类型：	主键	外键	唯一	非空	自增	默认值
关键字：	primary key	foreign key	unique	not null	auto_increment	default

## 知识点4 【DDL语句】-约束

### • 主键约束primary key

- 主键约束相当于 唯一约束 + 非空约束 的组合，主键约束列不允许重复，也不允许出现空值。
- 每个表最多只允许一个主键，建立主键约束可以在列级别创建，也可以在表级别创建。
- 当创建主键的约束时，系统默认会在所在的列和列组合上建立对应的唯一索引。

-- 基本模式

```
create table temp(  
id int primary key,  
name varchar(20)  
);
```

-- 组合模式

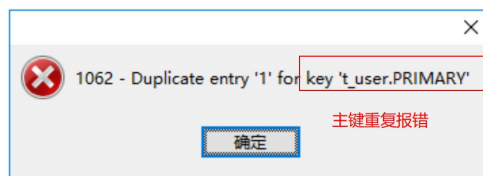
```
create table temp(  
id int ,  
name varchar(20),  
pwd varchar(20),  
primary key(id, name)  
);
```

-- 删除主键约束

```
alter table temp drop primary key;
```

-- 添加主键约束

```
alter table temp add primary key(id,name);
```



## 知识点4 【DDL语句】-约束



### • 唯一约束unique

- 唯一约束是指定table的列或列组合不能重复，保证数据的唯一性。
- 唯一约束不允许出现重复的值，但是可以为多个null。
- 同一个表可以有多个唯一约束，多个列组合的约束。
- 在创建唯一约束时，如果不给唯一约束名称，就默认和列名相同。
- 唯一约束不仅可以在一个表内创建，而且可以同时多表创建组合唯一约束。

```
-- 创建表时设置，表示用  
户名、密码不能重复  
create table temp(  
  id int not null ,  
  name varchar(20),  
  password varchar(10),  
  unique(name,password)  
);
```

```
-- 添加唯一约束  
alter table temp add unique (name, password);  
-- 修改唯一约束  
alter table temp modify name varchar(25) unique;  
-- 删除约束  
alter table temp drop index name;
```

## 知识点4 【DDL语句】-约束



- 非空约束

- NULL: 字段值可以为空，默认的
- NOT NULL: 字段值禁止为空

- 默认值约束

- default: 保证字段总会有值，即使没有插入值，都会有默认值。

```
create table student(  
  -- 姓名 不能为空  
  sname varchar(40) not null,  
  -- 默认值为3 1-男 2-女 3--没选择  
  sex char(1) default '3',  
  -- 手机号 唯一，不能重复 并且不能为空  
  phone varchar(11) unique not null,  
);
```

## 知识点4 【DDL语句】-约束



### • 自增约束

- 通过给字段添加 AUTO\_INCREMENT 属性来实现主键自增长。
- 语法格式：字段名 数据类型 AUTO\_INCREMENT

```
create table student(  
    -- 学生编号 主键 自增  
    sno int primary key auto_increment,  
);
```

默认情况下，AUTO\_INCREMENT 的初始值是 1，每新增一条记录，字段值自动加 1。

一个表中只能有一个字段使用 AUTO\_INCREMENT 约束，且该字段必须有唯一索引，以避免序号重复（通常为主键）。

AUTO\_INCREMENT 约束的字段必须具备 NOT NULL 属性。

AUTO\_INCREMENT 约束的字段只能是整数类型（TINYINT、SMALLINT、INT、BIGINT 等）。

AUTO\_INCREMENT 约束字段的最大值受该字段的数据类型约束，如果达到上限，AUTO\_INCREMENT 就会失效。

## 知识点4 【DDL语句】-约束



- 外键约束foreign key

- 语法: Foreign key (本表字段名) references 主键表名 (主键名) 【on delete CASCADE】
- 外键约束是保证一个或两个表之间的参照完整性, 外键是构建于一个表的两个字段或是两个表的两个字段之间的参照关系
- 一张表中可以没有外键, 如果有就必须是另一张表的主键, 目的是关联两张表。

- 注意

- CASCADE: 父表中的参照列数据删除或更新后, 子表也随着删除或更新
- SET NULL: 当父表参照列数据删除或更新后, 子表中的外键列修改为null
- RESTRICT: 拒绝父表的更新或删除操作, 默认。
- NO ACTION: 同RESTRICT
- 案例: 创建学生表, 关联班级表。



## 知识点4 【DDL语句】-修改表结构



### • 修改表

- 添加单列: ALTER TABLE 表名 ADD[COLUMN] 列名 列定义 [FIRST|AFTER]
- 添加多列: ALTER TABLE 表名 ADD[COLUMN] (列名 列定义,.....)
- 删除列: ALTER TABLE 表名 DROP[COLUMN] 列名
- 删除多列: ALTER TABLE 表名 DROP[COLUMN] 列名,DROP[COLUMN] 列名
- 添加主键约束: ALTER TABLE 表名 ADD[CONSTRAINT [约束名称]] PRIMARY KEY(主键)
- 删除主键约束: ALTER TABLE 表名 DROP PRIMARY KEY;
- 修改列定义: ALTER TABLE 表名 MODIFY[COLUMN] 列名 列定义 [FIRST|AFTER 列名]
- 修改列名称: ALTER TABLE 表名 CHANGE[COLUMN] 旧列名 新列名 新列定义 [FIRST|AFTER 列名]
- 修改表名: ALTER TABLE 旧表名 RENAME 新表名

## 知识点5 【DML语句】-插入

- DML 操作是指对数据库中表记录的操作，主要包括表记录的插入（insert）、更新（update）、删除（delete）和查询（select），是开发人员日常使用最频繁的操作。
- 数据插入记录

- 全表插入语法：INSERT INTO 表名 VALUES( value1,value2,...,valuen);
- 选择列插入语法：INSERT INTO tablename(field1,field2,...fieldn) VALUES(value1,value2,...,valuen);

注意：1、可以不用指定列名，但是values后面的顺序应该和表字段的排列顺序一致；

2、含可空的字段、非空但含有默认值的字段以及自增字段，可以不在insert后的字段列表里面出现，values后面对应字段名称的值；

案例1：向学生表中添加的韩梅梅的学生信息

案例2：向学生表中添加小岳岳的学生信息，只添加姓名和手机号

## 知识点5 【DML语句】-更新



### • 更新记录

- 基本语法：UPDATE 表名 SET 列名=新值1,列名=新值2,...[WHERE CONDITION]

注意：where 子句是判断语句，用来设定条件，限制只更新匹配的行，where可以省略

### • 案例

- 1、更新小岳岳的年龄为18岁。
- 2、更新韩梅梅的年龄是20岁，手机号为13240099009。
- 3、年龄小于10岁的所有学生，更新邮件为空。

## 知识点5 【DML语句】-删除



- 删除记录

- 基本语法：DELETE FROM 表名 【WHERE where\_condition】；

说明：此语句删除表中的行，如果不带where子句，则删除整个表中的记录,但是表不被删除，删除有关联的记录时，先删除从表，再删除主表。where可以省略

- 案例：删除李四的用户信息

## 知识点5 【DML语句】-查询-1

- SELECT是SQL数据操纵语言(DML)中用于查询表格内字段数据的指令，可搭配条件限制的子句(如where)或排列顺序的子句(如order)等来获取查询结果。
- 查询语句基本语法：

**select** 查询的字段名1,查询的字段名2, ...

**from** 表名

**where** 子句

**group by**子句

**having** 子句

**order by** 列 asc/desc

**limit** a,b

SELECT语句用于从一个或多个表中查询出需要的信息。

## 知识点5 【DML语句】-查询-2



- 查询表中所有数据，一个最基本的SELECT语句如下：
- 查询语句基本语法：

```
SELECT * FROM 表名
```

其中“\*”是一个特殊的列名，  
它表示一个表格中所有的列。

- 案例：查看所有员工全部信息
- 查询指定字段的数据：
  - 基本语法：select 字段1, 字段2,... from 表名
- 案例1:显示所有员工的姓名
- 案例2.显示所有员工的姓名和手机号

## 知识点5 【DML语句】-查询-3



- AS关键字

- 表中的列名由于经常采用简写或英文字符，有时不便于查看，这时可以使用为列重命名的方法将列名显示为易于理解的别名。
- 为列重命名可以采用如下方法：

```
SELECT 列名 AS 新列名,列名 AS 新列名,.. FROM 表名
```

- 对表取别名方法：

```
SELECT 新表名.列名, 新表名.列名,... FROM 表名 as 新表名
```

as关键字可以省略

案例：查询每个员工的姓名和工资，工资的列名显示为工资



- where条件语句

- 作用：检索数据记录中符合条件的行
- 注意：搜索的条件由一个或者多个表达式组成
- where子句：配合update、delete、select使用，where是对表中的行进行匹配。





### • where查询条件分类

查询条件	谓词	作用
比较	=	相等
	<=>	高级等于
	>	大于
	<	小于
	>=	大于或等于
	<=	小于或等于
	<>或!=	不等于



### • where查询条件分类

确定范围	BETWEEN 值1 AND 值2	在值1和值2之间，包含值1和值2
确定集合	IN	在指定的集合中
字符匹配	LIKE	与指定的模式串匹配
空值	IS NULL	为空
	IS NOT NULL	不为空
多重条件	NOT	与其它条件组合，表式反意
	AND	使多个条件同时满足（交集）
	OR	多个条件之一满足（并集）



## • where查询条件分类案例

①比较条件 = > >= < <= 高级等于<=> 不等于!=或<>

- 1、查询年龄为18岁的员工
- 2、查询年龄大于等于18岁的员工
- 3、查询年龄不等于18岁的员工

②判断空值 is null/is not null

- 1、查询手机号为空的员工
- 2、查询手机号不为空的员工
- 3、查询没有分配部分的员工
- 4、查询已经分配部分的员工

③and和or

- 1、查询员工表中开发部门的男性员工的姓名、性别和手机号码
- 2、查询lucy或者是jack的员工信息
- 3、查询员工年龄小于20岁的男性工资大于5000的员工信息
- 4、查询开发部或财务部男性的信息
- 5、用户登录

④确认集合(or关系 关键字or或者in)

- 1、查询年龄15 17 19的学生
- 2、查询lucy、rose和jack的员工信息

⑤确认范围(闭区间)between 值1 and 值2

- 1、薪资在5000-1000之间的员工信息

⑥精确查询

- 1、查看lucy员工详情

⑦模糊查询

- 1、查询名字里有o的员工的员工信息
- 2、查询名字中第二个字母是u的员工信息

## 知识点5 【DML语句】-查询-8



- 使用算术运算符

案例1：显示每个员工的年薪是多少

案例2：财务部门的员工全部涨薪1000元

- 字段排序

- 升序ASC 降序DESC

案例：查询所有员工信息，按照工资升序排序

- 去重distinct

案例1：查询员工表中的部门

案例2：查询员工表中部门的个数



- case when

- 语法：以case开头，中间when .... then....(可以写很多个，表示多个条件)，以end结束

案例：显示员工的信息以及工资的级别。如果大于10000是高工资，如果5000-10000之间是中等工资，如果小于5000 是基本工资。

- limit分页

语法：limit 起始条数 ,查询条数

案例：工资前三名的员工信息

## 知识点5 【DML语句】-聚合函数-1



- 聚合函数 (max,min,avg,sum,count)

- SQL提供的统计函数称为聚合函数.
- 主要的聚合函数:

COUNT(col) 返回指定列中非NULL值的个数

AVG(col) 返回指定数值列的平均值

MIN(col) 返回指定数值列的最小值

MAX(col) 返回指定数值列的最大值

SUM(col) 返回指定数值列的所有值之和

以上函数均可以同DISTINCT连用

案例1: 查询员工表中 最大的工资数  
最小工资数, 平均工资, 工资总和和  
公司人数

案例2: 查询员工表中工资最高的员  
工信息



### • 分组group by

- 语法：group by 字段1,字段2....
- 作用：用于对查询结果的分组统计。 eg：每个部门工资最高的
- 注意：
  - 1.通常和聚合函数一起使用才有意义。eg:sum count
  - 2.出现在select 后面的字段要么是在聚合函数中，要么就在group by后
  - 3.要筛选结果可以先使用where 再使用group by 最后使用having。

案例1：查询每个部门最高工资是多少

案例2：查询每个部门最高工资的员工信息



### • 分组group by 案例

案例3：创建表结构并且插入数据

各种产品年销售量统计表 sale

年	产品	销量
2005	a	700
2005	b	550
2005	c	600
2006	a	340
2006	b	500
2007	a	220
2007	b	350

(1) 获取每年销量最多的产品的相关信息。





### • 分组筛选 having

- 语法：having 对分组后的结果进行筛选，和group by 一起使用
- 案例：查询最高工资大于5000的部门。

### • 执行顺序 select ....from ... where ....group by .... having.....order by

(1)from---确认数据源

(2)where--- 将满足where条件的数据放在查询结果中

(3)group by --- 对where条件查询出来的结果进行分组

(4)having --- 对组进行筛选，留下有用的组

(5)select -- 根据select里的显示要求，对有用的数据进行统计计算和筛选

(6)order by -- 对统计好的数据进行排序



### • 合并数据

- ①union: 合并多条查询语句的结果, 合并过程中去掉重复的记录, 只保留一条。

```
SELECT column_name FROM table1  
UNION  
SELECT column_name FROM table2
```

案例: 返回所有部门名称, 要在列表里加一个请选择部门。

- ②union all: 合并多条查询语句的结果, 合并过程中保留重复记录

```
SELECT column_name FROM table1  
UNION ALL  
SELECT column_name FROM table2
```



### • EXISTS

- exists表示()内子查询语句返回结果不为空说明where条件成立就会执行主sql语句，如果为空就表示where条件不成立，sql语句就不会执行。not exists和exists相反，子查询语句结果为空，则表示where条件成立，执行sql语句。否则不执行。
- 例：

```
select * from employee where exists(select * from dep where 1=1);
```

```
select * from employee where exists(select * from dep where 1=0);
```

案例：两张表用户表users(userid,address,phone)和消费表 consume(userid,time,amount)，需要查消费超过5000的用户记录。

## 知识点5 【DML语句】-EXISTS和WITH-2



### • EXISTS

用in:

```
select * from users where userid in  
(select userid from consume where consume.amount>5000);
```

用exists:

```
select * from users where  
exists (select 1 from consume where users.userid=consume.userid and consume.amount>5000);
```

通常情况下采用exists要比in效率高。

(1)exists()后面的子查询被称做相关子查询，他是不返回列表的值的。只是返回一个true或false的结果(这也是为什么子查询里是"select 1"的原因，换成"select 6"完全一样，当然也可以select字段，但是明显效率低些)。

其运行方式是先运行主查询一次，再去子查询里查询与其对应的结果，如果是true则输出，反之则不输出，再根据主查询中的每一行去子查询里去查询。

(2)in()后面的子查询是返回结果集的，换句话说执行次序和exists()不一样，子查询先产生结果集，然后主查询再去结果集里去找符合要求的字段列表去，符合要求的输出，反之则不输出。

## 知识点5 【DML语句】-EXISTS和WITH-3



### • WITH

- 在数据库开发中有时候要对临时数据的一些处理，但是这些数据又不想放到table里面。这个时候可以考虑with as。with as是在内存中建立一个临时的table。可以对这个temp table里面的数据进行整理。

```
with table1 as(  
    select 1 as a from employee  
    union  
    select 2 as a from employee  
    union  
    select 3 as a from employee  
)
```

```
select * from table1;
```

注：with子句的好处在于：如果遇到多个同一子查询时，采用此方式效率比一般子查询高，而且代码结构更清晰易懂

案例：显示员工信息并计算工资合计、平均工资。（优化，尽量少查实体表）。

# 知识点5 【DML语句】-多表查询-1



## • 多表查询

- 语法：select 字段 from 表1,表2 where 连接条件;
- 注：当使用多表查询时，只会显示满足连接条件的数据，不满足连接条件的数据会忽略。

案例1：查询lucy的员工信息以及部门名称

案例2：查询所有员工的姓名、部门名称、工资

案例3：查询工资在7000 -- 12000 之间的员工姓名、工资、部门名称

练习1：查询工资在7000 -- 12000 之间女性的员工姓名、工资、部门名称

练习2：查看所有员工的姓名、工资、部门名称以及职位称



- 子查询（嵌入到其他sql语句中的select语句，也叫嵌套查询）

- ①单行子查询-子查询的结果只有一条数据

案例：查询表中与rose同部门的人员名字

- ②多行子查询-子查询查出来的结果是多条数据

案例：查询工资高于5000的员工所属部门信息



### • 自连接查询

- 自连接就是一个表和它自身进行连接，是多表连接的特殊情况。在自连接查询中，要先在FROM字句中为表分别定义两个不同的别名，然后使用这两个别名写出一个连接条件。

- `select 字段 from 表1 别名1,表1 别名2 where 连接条件;`

- 注：当使用自连接时，只会显示满足连接条件的数据，不满足连接条件的数据会忽略。

案例：（前提）为员工表中添加管理者id这一列

```
alter table employee add mgr int;
```

(1)查询员工的姓名以及他的管理者的姓名



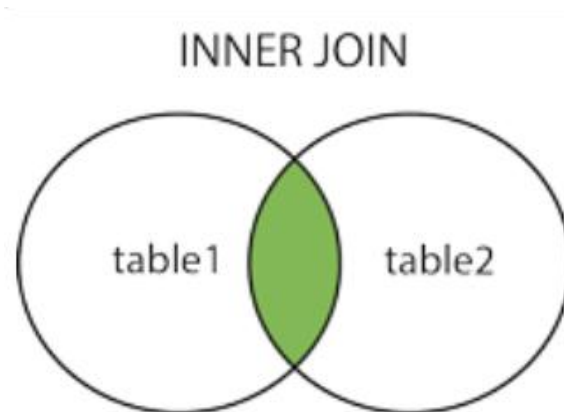


### • 内连接INNER JOIN

- 概述：MySQL INNER JOIN获取两个表中字段匹配关系的记录。查询结果只会显示满足条件的记录，和前面所讲的多表查询结果一样。
- 语法如下：

```
SELECT column_list  
FROM t1  
INNER JOIN t2 ON join_condition1  
INNER JOIN t3 ON join_condition2...  
WHERE where_conditions;
```

案例:查询所有的员工信息以及职位信息



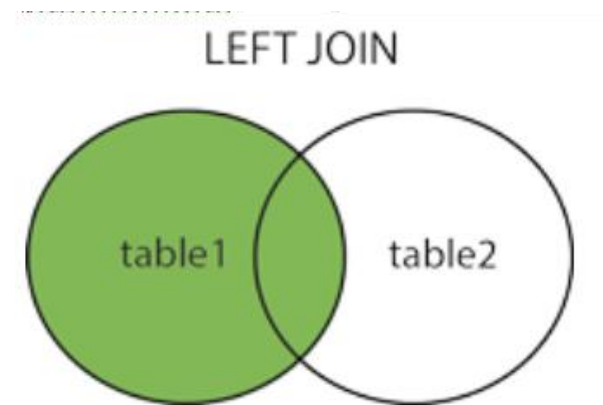


### • 左连接LEFT JOIN

- 概述：查询结果以左表为主，左表的数据全部显示，右表为辅，符合条件的显示，不符合条件的就补null空值。
- 语法如下：

```
SELECT column_list  
FROM t1  
LEFT JOIN t2 ON join_condition1  
LEFT JOIN t3 ON join_condition2...  
WHERE where_conditions;
```

案例：查看所有员工的姓名、工资以及职位

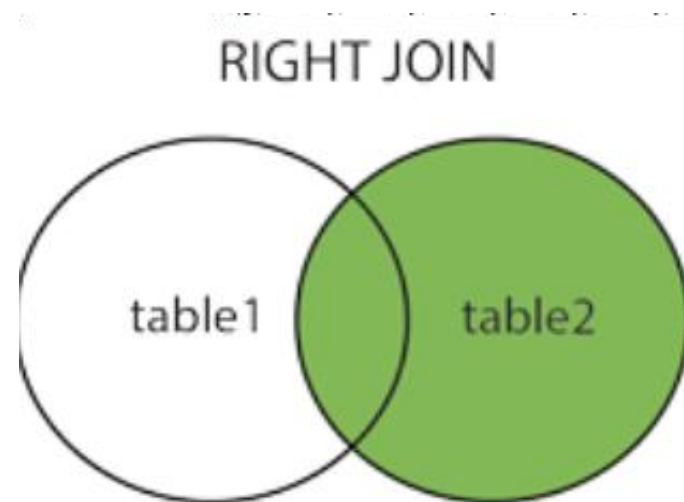


### • 右连接RIGHT JOIN

- 概述：查询结果以右表为主，右表的数据全部显示，左表为辅，符合条件的显示，不符合条件的就补空值。
- 语法如下：

```
SELECT column_list  
FROM t1  
RIGHT JOIN t2 ON join_condition1  
RIGHT JOIN t3 ON join_condition2...  
WHERE where_conditions;
```

案例：查询所有部门的员工



## 知识点6 【DCL语句】 -1

- DCL(Data Control Language), 即数据控制语言, 用来定义数据库的访问权限和安全级别, 及创建用户。
- 主要包括创建用户、给用户授权、对用户撤销授权、查询用户授权和删除用户等。

Thanks

感谢