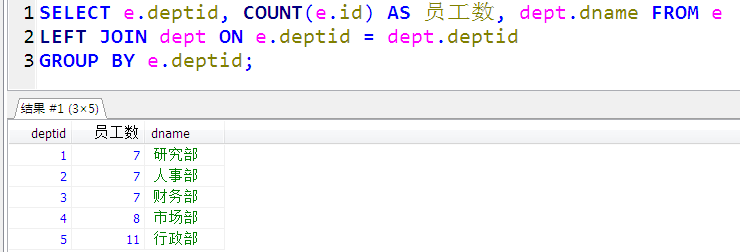
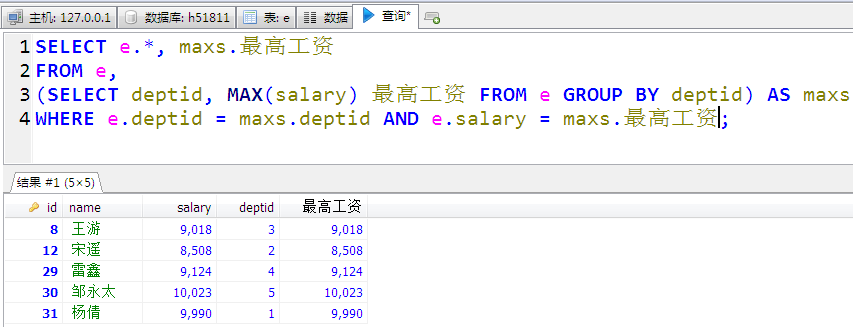
# 第三天上课笔记

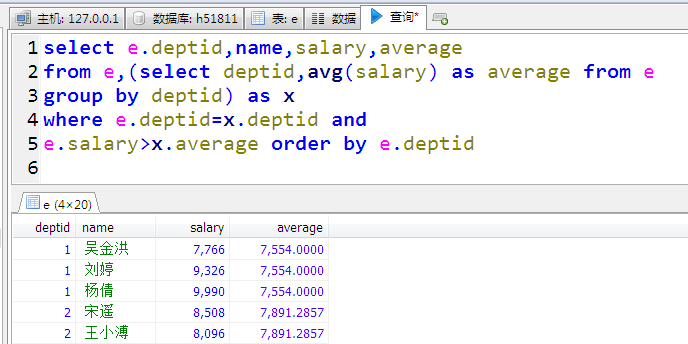
1. 查出每个部门有多少员工；



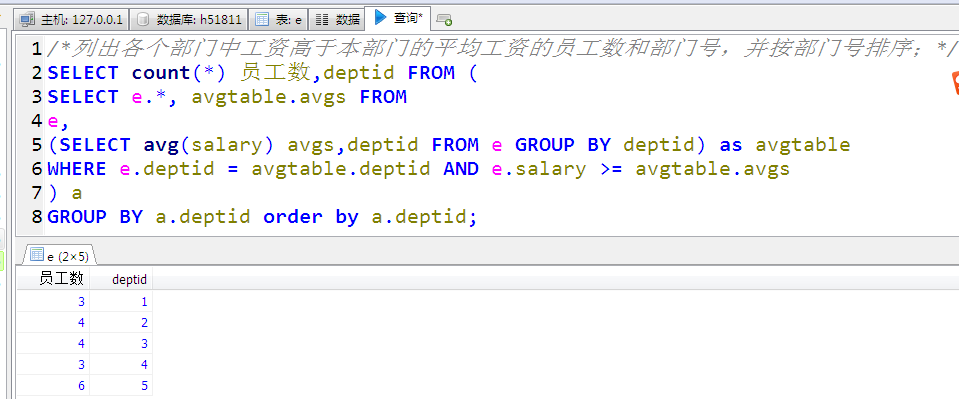
1. 查出每个部门工资最高的员工；



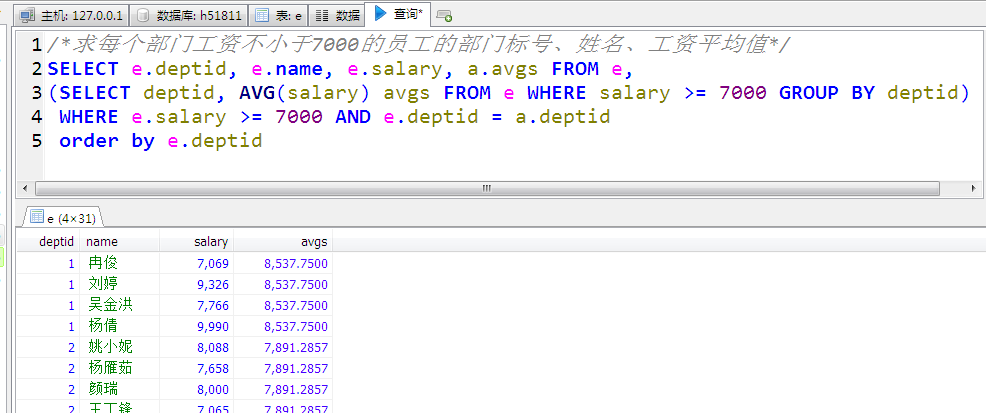
1. 查出每个部门高于部门平均工资的员工名单；



1. 列出各个部门中工资高于本部门的平均工资的员工数和部门号，并按部门号排序；



1. 求每个部门工资不小于7000的员工的部门标号、姓名、工资平均值；



# 如何创建表？

一件事情（功能点）需要一张表，一张表干一件事情；

学生信息：students

班级：class

成绩：score

考勤：check

这个表应该如何设计？

范式：通俗理解；

灵活的使用而不拘泥于范式：需求、性能、范式；

当表的**属性**不能再分解为更基本的数据单位时，保持了属性的**原子性和唯一性**，不产生冗余数据，则称R是满足第一范式，简称1NF；1NF是关系模式的最低要求。

若R满足1NF，且它的每一非主属性完全依赖于主键，从而保证一行数据只做一个事情，则称R满足第二范式，简称2NF。

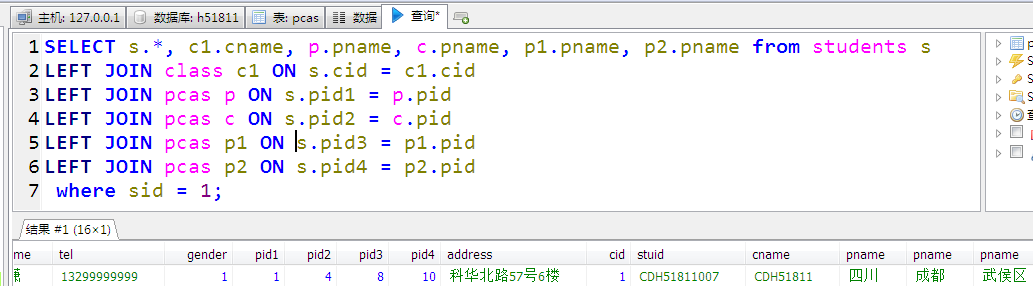
在设计表的时候，第一件事情就是创建一个主键：唯一的值



**删除、查询、修改**的时候，是根据主键来操作的；

若R满足2NF，且每一非主属性不传递依赖于主键，也就是让非主属性和主键之间有直接关系，则称R满足第三范式，简称3NF。

LEFT JOIN的多表以及一个表多次的情况：



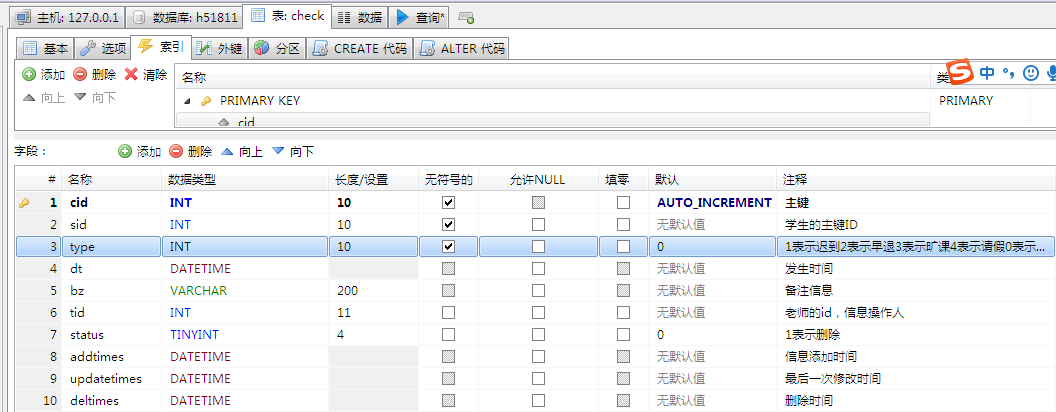
**外键：**



我们在设计表的时候，主观意识上知道外键关系，一般不设计外键；

考勤表设计：

考勤：迟到、早退、旷课、上课睡觉、打游戏、作业提交情况等；



# 复习：

数据库服务器：安装了数据库管理系统软件的服务器；

数据库管理系统：DBMS, 常见：MySQL、Oracle；

数据库：组织和管理数据的仓库；

表：数据库里面存放数据的地方；

字段、属性、列：；

记录：一条一条的数据；

数据：字段和记录交叉的地方；

三个范式：

字段如果能够分割尽可能的分割，保证字段的原子性；

字段的内容不能重复，保证字段的唯一性；

表里面要有一个主键，保证一条记录只干一件事情；

数据避免冗余：如果有大量的冗余数据，应该单独创建一个表；

字段的数据类型：要不要引号的问题

数字：

Int

文本：字符串

Char：

Varchar：

Text：存放长文本的，比如新闻详情；

日期时间：

Date、time、datetime、year

对数据的操作：增删改查

增加：INSERT INTO table( f1, f2, f3…) VALUES (v1, ‘v2’, v3….)

删除：DELETE FROM table where 判断条件

TRUNCATE table：表示清空表，主键会从1重新开始。

修改：UPDATE table SET f1 = v1, f2=’v2’, f3=v3 ….. WHERE 判断条件；

真正在做删除的时候应该进行修改操作：status = 0；

查询：

SELECT 要查询的字段 FROM table WHERE 判断条件

\*：全部的字段；

AND 、OR、NOT：连接判断条件；

LIKE：模糊查询；

ORDER BY：排序，ASC|DESC;

GROUP BY：分组查询；

COUNT、MAX、MIN、AVG、SUM、RAND：常用函数;

IN：指定可选值

BETWEEN…AND：指定范围

LIMIT start, num:返回指定的条数，可以实现分页；

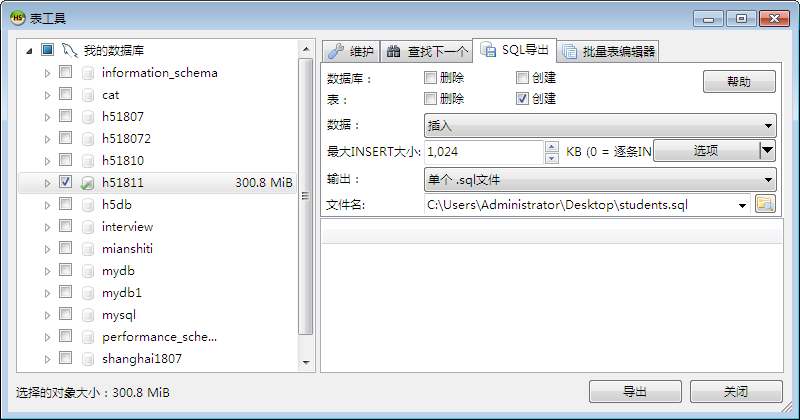
DISTINCT：去重查询；

HAVING：和聚合函数一起作为判断条件；

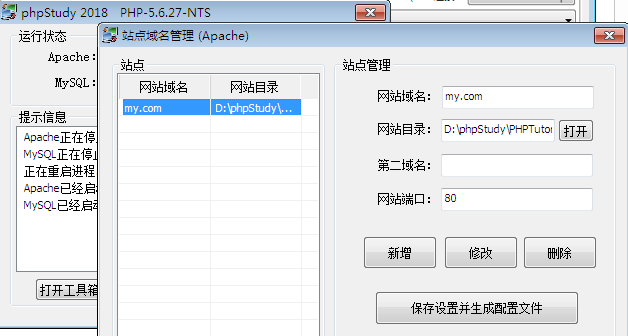
CONVERT：转换

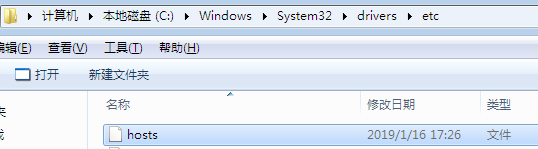
CONCAT：字段或字符串连接

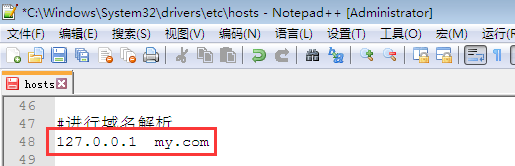
### 数据库或表的导出导入：

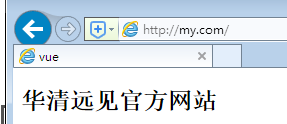


Host配置文件：进行本地域名解析

站点管理：  








Chrome浏览器的设置：



网址：<https://blog.csdn.net/thewindkee/article/details/80668974>