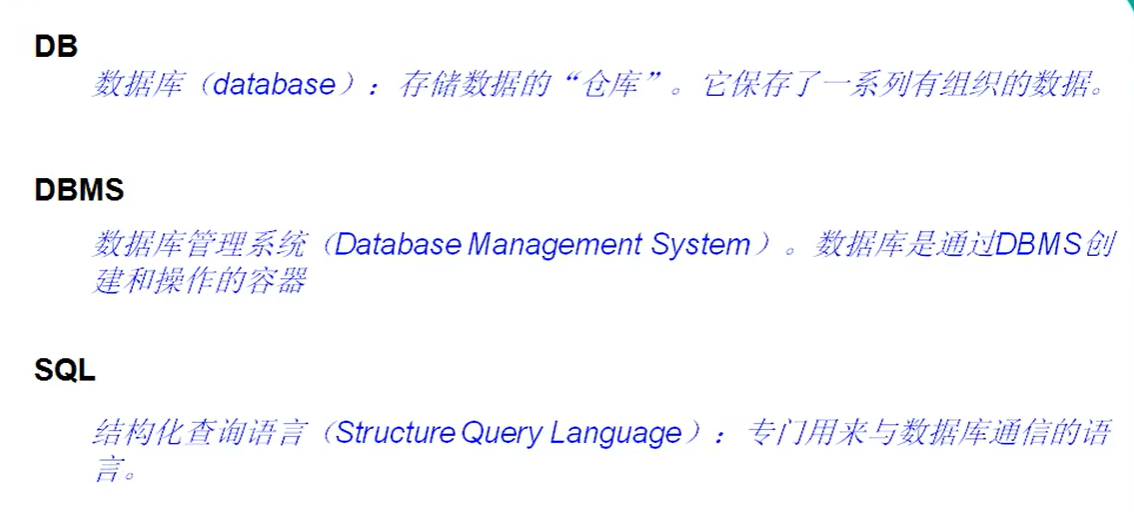
**MySQL**

**数据库的概念**

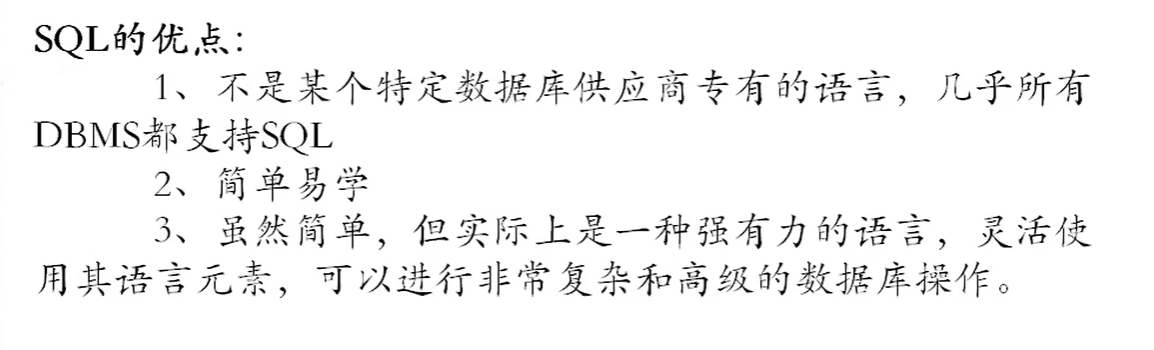


**DB：数据库，用于存放一系列有组织的数据的容器**

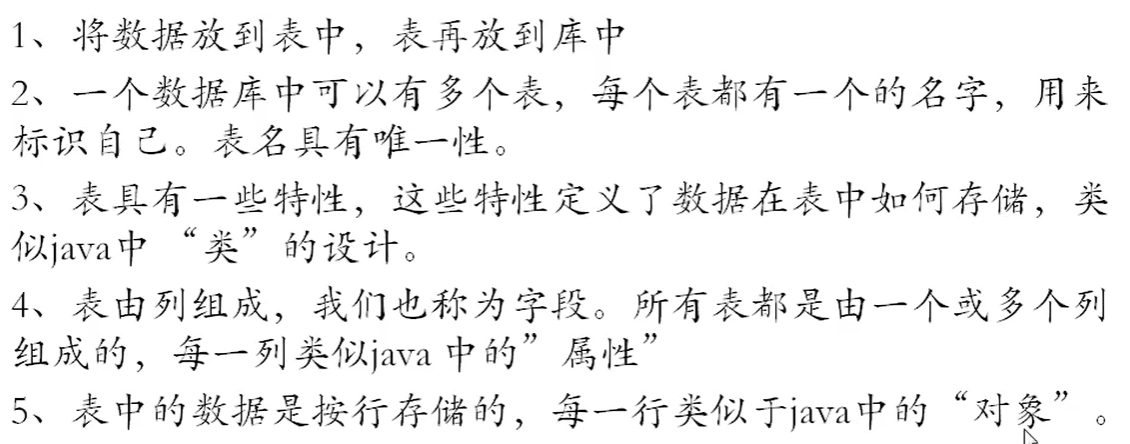
**DBMS: 数据库管理系统，又称为数据库软件(产品)，用于管理DB中的数据**

**SQL：结构化查询语言，用于和DBMS通信的语言（DBMS能够识别，并执行的语言）**

**SQL包括DQL(数据查询语言)，DML(数据操纵语言)，DDL(数据定义语言)，DCL(数据控制语言)，TCL(事物控制语言)**



数据库的特点



**DQL(数据库查询语言)的学习**

**基础查询**

语法:

SELECT (字段，常量值，表达式，函数) FROM 表名

查询列表

查询字段(列)

查询单个，多个，或全部字段；

查询常量

eg:

SELECT 100\*6 AS 结果；

执行结果:



拓展：

**起别名**

AS 可以给查询的字段起别名，还可以使用空格取别名

SELECT 100\*6 AS 结果；等价于 SELECT 100\*6 结果；

如果要查询的字段有重名的情况，可以使用别名区分；但这里需要注意，如果取的别名中包含特殊符号(空格，#，$......)，需要用双引号将别名引起来

**去重**

在要查询的字段前加distinct 关键字，可以使查询的结果去除重复值

**显示表结构(查看表的结构)**

DESC 表名；(sql server中没有此命令)

**mysql 中的 + 号运算符**

* SELECT 100 + 100；两个操作数都为数值型，则做加法运算
* SELECT ‘124’ + 100；若一方为字符型，则会尝试将字符型数据转化成数值型

若转换成功继续执行加法操作，执行结果为: 224

* SELECT ‘abc’ + 100；若转化失败，将字符型数据转化成0，继续运算，执行结果为100；在sql server 中转化失败，就执行报错，无法再运行
* SELECT NULL + 100；若一方为NULL，执行结果就为NULL使用concat可以将要查询的多个字段拼接成一个字段

格式: concat(字段1，字段2，...)

**IFNULL函数**

用法:

IFNULL(列名，返回值)；//如果查询的某个列中的值为NULL，可以指定一个返回值作为查询结果返回

**条件查询**

语法:

SELECT

查询列表

FROM

表名

WHERE

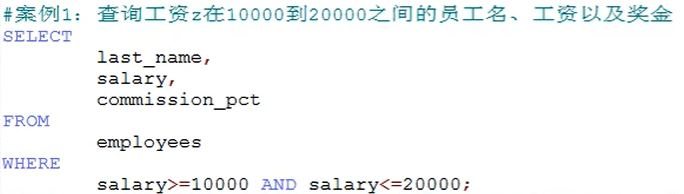
筛选条件;

分类:

* **按条件表达式筛选**

条件运算符: > < = != <> >= <=

例如:



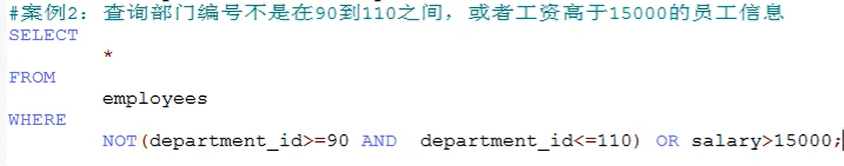
* **按逻辑表达式筛选**

逻辑运算符: && || !

and(与) or(或) not(非) 推荐使用这种方式

逻辑运算符一般用来连接条件表达式

例如:



* **模糊查询**

like

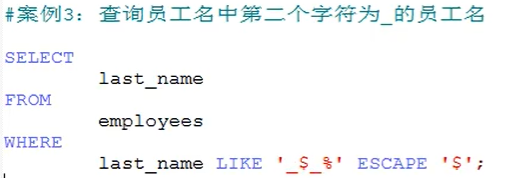
like一般和通配符搭配使用

常见通配符:

% 代表任意多个字符

\_ 代表单个字符

例如：

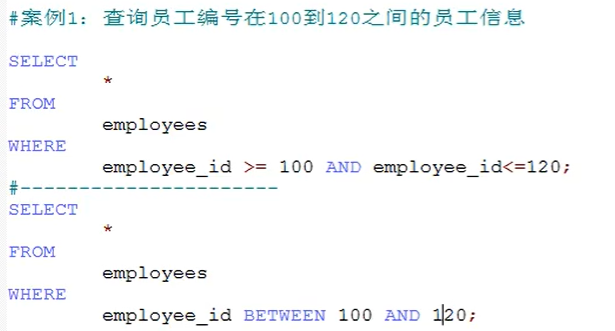


注: $ 起转义字符的作用，将其后面的下划线转换成普通的字符 ‘ \_ ’ ，这里也可以不用$代表转义字符，用其它合法的标识符都行。ESCAPE关键字声明字符$是一个转义字符

between and (在...之间) / not between and (不在...之间)

作用: 查询在某个范围内的数据时使用

例如:



对比而言，使用between and 会使语句显得更简洁

in

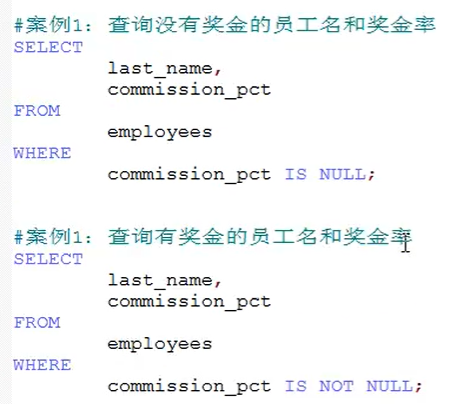
作用: 判断要查询的字段值是否属于处于某个集合中

is null / is not null

= 或 <> 不能用来判断null值

is null 或 is not null 可以用来判断null值

例如:



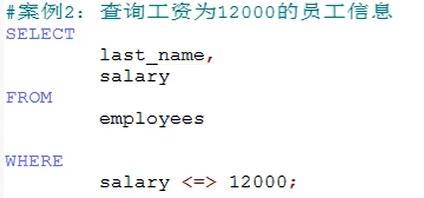
注意:

不能将commission\_pct IS NULL；改成 commission\_pct = NULL；

commission\_pct = NULL；从字面意思看起来没有错，但SQL中没有这种语法。

拓展:

<=> 安全等于 (sql server中没有这种用法)



is null 和 <=>

is null: 只可以判断NULL值 (建议使用)

<=>: 既能判断NULL值，又能判断普通的数值

**排序查询**



排序列表可以是单个字段，多个字段，别名，表达式，函数

ORDER BY 子句一般是放在查询语句的最后面

例如:

(按表达式排序)

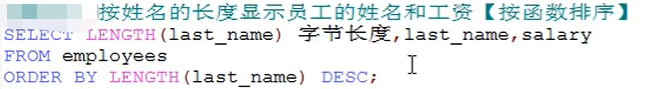


将



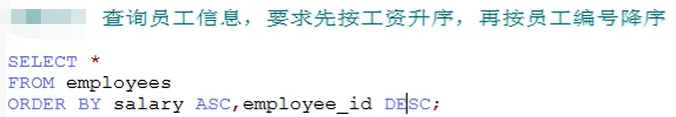
改为: ORDER BY 年薪; 也行

(按函数排序)



注:LENGTH(实参) 函数返回实参所占的字节数，sql server中为LEN()函数。

(按多个字段排序)



注意: 是先按工资升序排序，工资相同的再按员工编号降序排序

a mod b = a-a/b\*b

**关系化规范化**

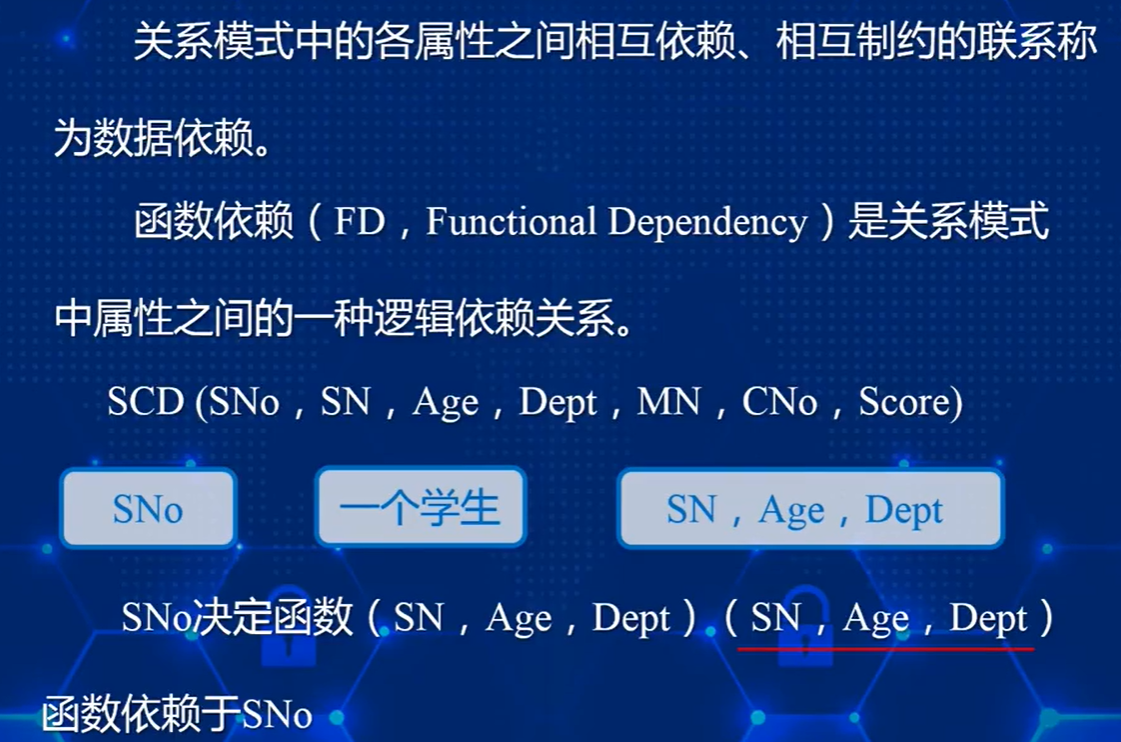
一个好的关系模式具备以下四个条件

* 尽可能少的数据冗余
* 没有插入异常
* 没有删除异常
* 没有更新异常

如果一个关系模式存在上诉问题，需要进行关系分解

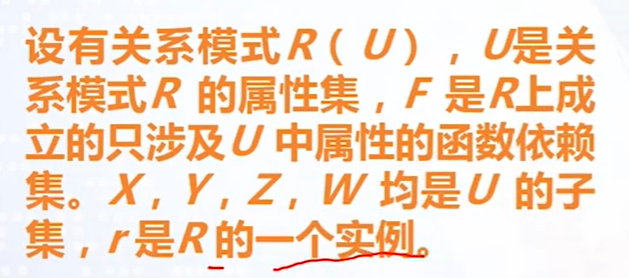
**函数依赖**

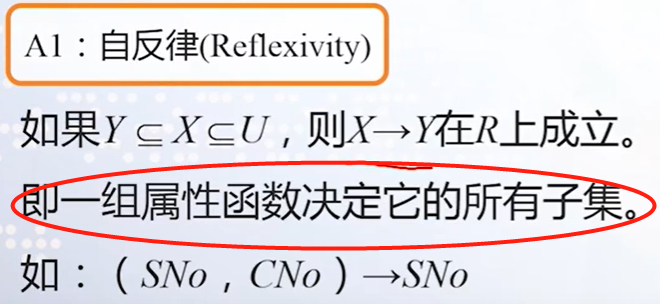
数据依赖包括函数依赖和多值依赖



**函数依赖的推理规则及正确性**

Armstrong公理及正确性





同样(Sno，Sco) -> Cno

