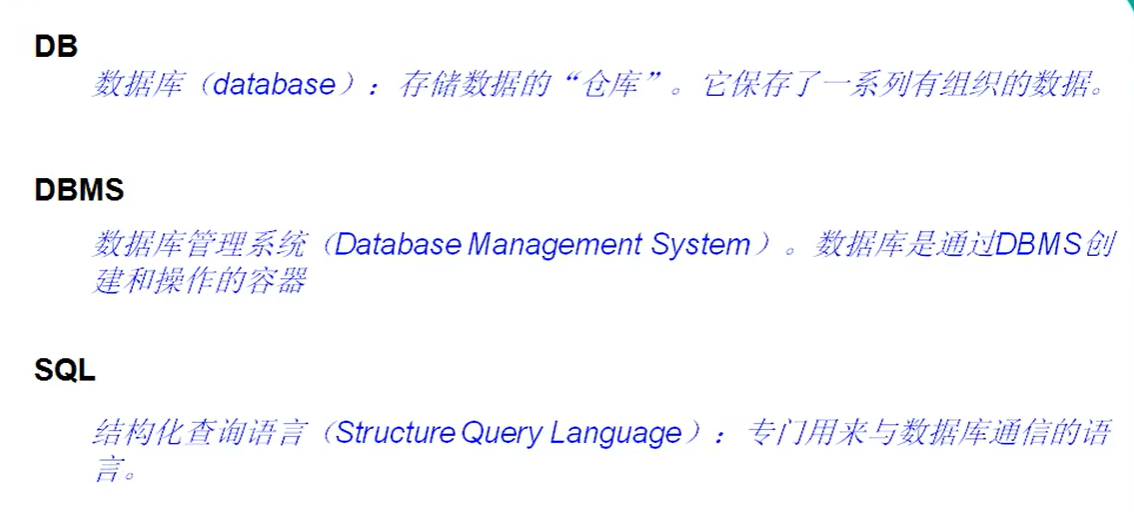
**MySQL**

**数据库的概念**

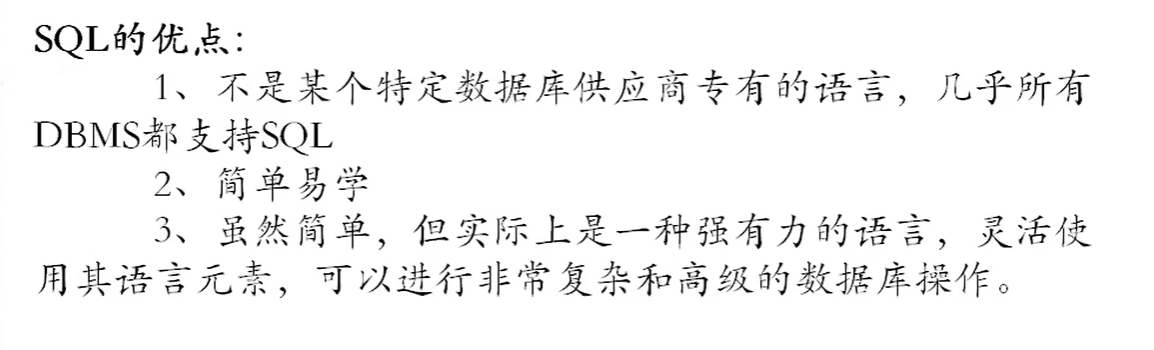


**DB：数据库，用于存放一系列有组织的数据的容器**

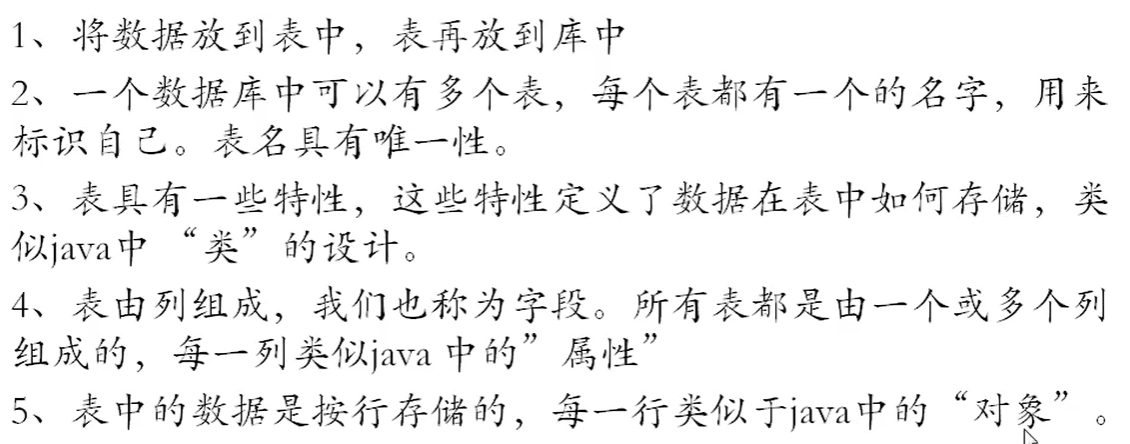
**DBMS: 数据库管理系统，又称为数据库软件(产品)，用于管理DB中的数据**

**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

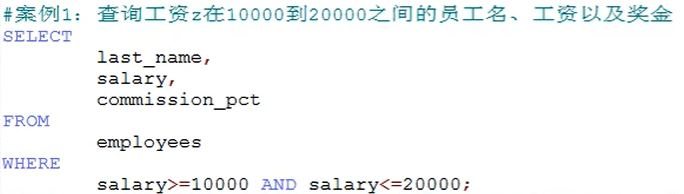
筛选条件;

分类:

* **按条件表达式筛选**

条件运算符: > < = != <> >= <=

例如:



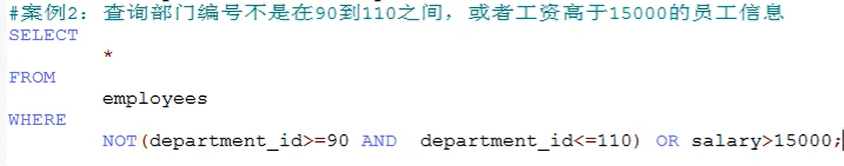
* **按逻辑表达式筛选**

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

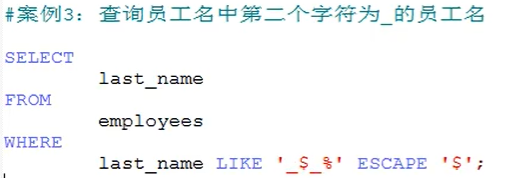
like一般和通配符搭配使用

常见通配符:

% 代表任意多个字符

\_ 代表单个字符

例如：

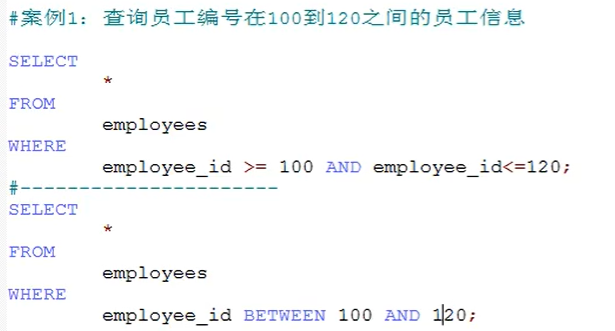


注: $ 起转义字符的作用，将其后面的下划线转换成普通的字符 ‘ \_ ’ ，这里也可以不用$代表转义字符，用其它合法的标识符都行。ESCAPE关键字声明字符$是一个转义字符

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

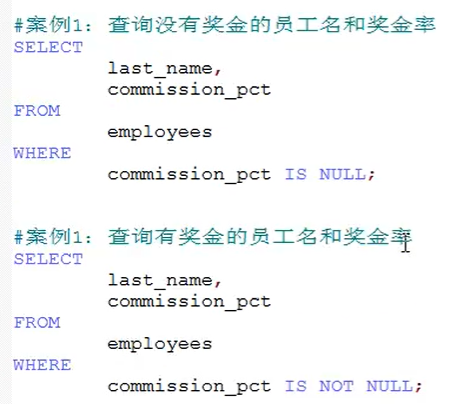
作用: 判断要查询的字段值是否属于处于某个集合中

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:



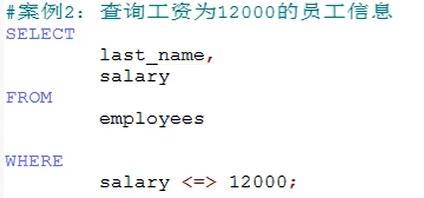
注意:

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)

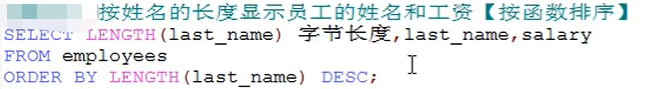


将



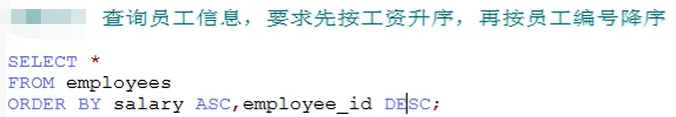
改为: ORDER BY 年薪; 也行

(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

**mysql中的一些常用命令**

- mysql -h localhost -u root -p //使用root用户，登录mysql

- set password for 用户名@localhost = password('新密码'); //mysql5.7以前修改密码的命令

- ALTER USER 'root'@'localhost' IDENTIFIED BY '123456'; //mysql5.7之后修改密码的命令

**存储引擎**

存储引擎是一种表存储及组织数据的方式，不同的存储引擎表存储数据的方式不

同。MySQL内置了9个存储引擎，使用InnoDB作为默认的存储引擎，较其它

存储引擎而言，它的最大特点对事务的支持。常见的存储引擎还有myisam，

memory。

show create table 表名 //查看表使用的是哪个存储引擎

show engines //查看所有的存储引擎

注：不同的版本的mysql支持的存储引擎可能不同。

**什么是事务? 事务的四种特性（acid原则）**

事务是一个完整的业务逻辑，是一个最小的，不可再分的工作单元，。简单讲的话，事务就是一系列操作的集合（这里的操作指的是能够对数据库中的数据进行修改的操作），这些操作作为一个整体一起提交，要么都执行、要么都不执行。事务的四大特性：

1 、原子性

事务是一个最小的，不可再分的工作单元，

2 、一致性

在同一个事务中的操作，要么多执行，要么都不执行，以保证数据的一致性。

3 、隔离性

一个事务的执行不能被其它并发的事务干扰。即一个事务内部的操作及使用的数据对其它并发事务是隔离的，并发执行的各个事务之间不能互相干扰。

4 、持久性

一个事务一旦提交，它对数据库中的数据的改变就应该是永久性的。接下来的其

它操作或故障不应该对其执行结果有任何影响。

注：在mysql中，事务是自动提交的。

start transation //手动开启事务

rollback //回滚事务

事务是怎么做到多个DML操作，要么一起执行，要么都不执行的呢?

InnoDB存储引擎，提供了一组用来记录事务性活动的日志文件

在事务的执行过程中，每一个操作都会被记录到日志文件中。在事务的执行过程

中可以提交事务，也可以回滚事务。

提交事务：清空用于记录事务活动的日志文件，，然后将数据全部持久化到数据

库表中。

回滚事务：将之前执行的所有操作都撤销，并且清空用于记录事务活动的日志文

件。

**索引**

索引是为了提高查询效率，而建立的一种机制。可以为表中的某一个字段建立索

引，也可以多个字段组合起来建立索引。总之，使用索引可以缩小检索的范围。

索引底层的数据结构是B+树（自平衡二叉树）。

什么情况下为字段添加索引

情况一：数据庞大

情况二：该字段经常出现where关键字后面，作为查找的条件，也就是这个字段经常被查找。

情况三：很少对这个字段进行增删改操作。因为这个字段经常发生改变，索引就要重新排序

注：在所有数据库中，只要是主键都会自动建立索引。在mysql中，如果某个字段添加了unique约束（唯一性约束），也会自动建立索引。

explain 查询语句 //查看某个查询语句的解释信息，从信息中可以查看到该语句是否使用了索引查找。

create index 索引名 on 表(字段1，字段2...) //为表中的某个字段或者多个字段建立索引

drop index 索引名 on 表 //删除表中的某个索引

索引的分类

单一索引：在一个字段上建立的索引。主键索引和唯一性索引是两个比较特殊的单一索引

复合索引：多个字段组合在一起，建立的索引

索引失效的情况

情况一：模糊查询时，如果%出现在关键字的左边，会导致索引失效

情况二：sql语句带or 关键字。因为or两边的的字段，如果其中的一个字没建立索引，会导致另一个字段建立的索引失效

情况三：如果已建立索引的字段参加了数学运算，会导致索引失效

情况四：在已建立索引的字段使用了函数，也会导致索引失效