复习：

day01：

软件过程：

软件定义期：可行性研究、需求分析

软件开发期：概要设计、详细设计、编码实现、测试

软件维护期：项目部署、项目维护

day02：

MySQL中常用的SQL语句：

show databases;

use 库名;

show tables;

desc 表名;

show warnings; #显示上一条语句产生的警告

show create table 表名; #显示创建表时所用的语句

-------------------------------------

drop database if exists 库名;

create database 库名 charset=utf8;

use 库名;

create table 表名( 列名 列类型 列约束 );

insert into 表名 values('值');

delete from 表名 where 条件;

update 表名 set 列名='值' where 条件;

select \* from 表名;

day03:

列类型：

数值类型：tinyint/smallint/int/bigint/decimal/bool

日期时间类型：date/time/datetime

字符串类型：char/varchar/text

列约束：

唯一/非空/**主键**/默认值/检查/外键

day04：

简单查询：

查询特定的列 select ename,sex from emp

给列取别名 select ename as n,sex s from emp

显示不同的值 select distinct sex from emp

执行运算 select salary\*12 from emp

单条件查询 select \* from emp where salary>6000

多条件查询 select \* from emp where deptId=10 and sex=0

模糊条件查询 select \* from emp where ename like '%E\_\_'

复杂查询：

今日目标：

(1)MySQL中的查询

(2)WEB服务器

|  |
| --- |
| Order: 订单、排序 |
| Set：设置、集合 |
| Table：表、桌子 |

1.MySQL中的简单查询 —— 查询结果的排序

示例：查询出所有员工信息，要求按工资由小到大排序

SELECT \* FROM emp

ORDER BY salary ; #默认是由小到大

#ORDER BY salary ASC; #ascendant，升序

示例：查询出所有员工信息，要求按工资由大到小排序

SELECT \* FROM emp

ORDER BY salary DESC ; #descendant，降序

练习：查询出所有员工信息，按照姓名由小到大排序

SELECT \* FROM emp

ORDER BY ename ;

练习：查询出所有员工信息，按照姓名由大到小排序

SELECT \* FROM emp

ORDER BY ename DESC;

练习：查询出所有员工信息，按照年龄由大到小排序

SELECT \* FROM emp

ORDER BY birthday ASC ; #生日由小到大排列

练习：查询10号部门中员工的所有信息，按工资由大到小排列

SELECT \* FROM emp

WHERE deptId=10

ORDER BY salary DESC ;

#注意：where子句和order by子句的顺序问题

练习：查询出所有员工信息，按工资由大到小排列；工资相同的，再按姓名由小到大排列

SELECT \* FROM emp

ORDER BY salary DESC, ename ASC ;

练习：查询所有员工信息，按生日由大到小排列——前提是：所有女员工必须排在所有男员工之前

SELECT \* FROM emp

ORDER BY sex ASC , birthday DESC ;

2.MySQL中的简单查询 —— 分页查询 —— 重点&面试重点

分页查询：用于分批次的显示大量的数据。语法：

SELECT .... FROM ... WHERE ... ORDER BY ....

LIMIT start, count ;

(1)start：是一个数字，表示从哪一行开始获取数据

(2)count：是一个数字，表示一次最多获取的行数，即页面大小

假设每页显示5行记录，则

第1页： LIMIT 0, 5

第2页： LIMIT 5, 5

第3页： LIMIT 10, 5

第4页： LIMIT 15, 5

第5页： LIMIT 20, 5

....

第n页： LIMIT (n-1)\*5, 5

练习：分页查询出员工信息，页面大小为5，显示第1页

SELECT \* FROM emp LIMIT 0, 5 ;

练习：分页查询出员工信息，页面大小为5，显示第2页

SELECT \* FROM emp LIMIT 5, 5 ;

练习：分页查询出员工信息，页面大小为5，显示第3页

SELECT \* FROM emp LIMIT 10, 5 ;

练习：分页查询出员工信息，页面大小为5，显示第4页

SELECT \* FROM emp LIMIT 15, 5 ;

3.MySQL复杂查询 —— 子查询

示例：查询出“研发部”所有员工的信息

步骤1：查询研发部的部门编号 => 10

SELECT did FROM dept WHERE dname='研发部';

步骤2：查询部门编号为10的员工信息

SELECT \* FROM emp WHERE deptId = 10;

综合上述两步为一条语句：

SELECT \* FROM emp WHERE deptId =( #父查询

SELECT did FROM dept WHERE dname='研发部' #子查询

) ;

练习：查询出工资比TOM高的所有员工信息

步骤1：查询TOM的工资 => 6000

SELECT salary FROM emp WHERE ename='TOM';

步骤2：查询工资大于6000的员工信息

SELECT \* FROM emp WHERE salary>6000;

综合上述两步：

SELECT \* FROM emp WHERE salary>(

SELECT salary FROM emp WHERE ename='TOM'

);

练习：查询出比TOM年长的所有员工信息

步骤1：查询TOM的生日 => 1990-1-1

SELECT birthday FROM emp WHERE ename='TOM';

步骤2：查询出生日小于1990-1-1的员工所有信息

SELECT \* FROM emp WHERE birthday<'1990-1-1';

综合上述两步：

SELECT \* FROM emp WHERE birthday<(

SELECT birthday FROM emp WHERE ename='TOM'

);

练习：查询出与TOM同年出生的所有员工信息

步骤1：查询TOM的生日中的年份 => 1990，提示：用year( )运算

SELECT YEAR(birthday) FROM emp WHERE ename='TOM';

步骤2：查询出生日中年份等于1990的员工所有信息

SELECT \* FROM emp WHERE YEAR(birthday)='1990';

综合上述两步：

SELECT \* FROM emp WHERE YEAR(birthday)=(

SELECT YEAR(birthday) FROM emp WHERE ename='TOM'

);

4.MySQL复杂查询 —— 聚合/分组查询

函数：Function，就是一个功能体，可以接收若干数据，加以处理，得到最终希望的结果——类似饺子机。

MySQL提供了几个聚合函数：

COUNT()：计算若干数据的个数

SUM()：计算若干数据的总和

AVG()：计算若干数据的平均值

MAX()：计算若干数据的最大值

MIN()：计算若干数据的最小值

示例：查询出员工的总数

SELECT COUNT(eid) FROM emp; #14

SELECT COUNT(deptId) FROM emp; #13，NULL不算数

SELECT COUNT(sex, deptId) FROM emp;

SELECT COUNT(ename, sex, deptId) FROM emp;

SELECT COUNT(\*) FROM emp; #推荐写法

示例：查询出所有员工工资的总和

SELECT SUM(salary) FROM emp;

练习：查询出所有员工工资的平均值（用两种方法）

SELECT SUM(salary) / COUNT(salary) FROM emp;

SELECT AVG(salary) FROM emp;

练习：查询出所有员工工资的最大值和最小值

SELECT MAX(salary), MIN(salary) FROM emp;

练习：查询出工资高于所有人平均工资的员工的数量 —— 难度系数：5星

步骤1：查询出所有员工的平均工资 => 6842

SELECT AVG(salary) FROM emp;

步骤2：查询出工资大于6842的员工的数量

SELECT COUNT(\*) FROM emp WHERE salary>6842;

综合上述两步：

SELECT COUNT(\*) FROM emp WHERE salary>(

SELECT AVG(salary) FROM emp

);

示例：查询出每个部门的编号及该部门员工的数量——先分组再聚合计算

SELECT deptId, COUNT(\*) FROM emp

GROUP BY deptId ; #按照部门编号分组

|  |
| --- |
| 注意：分组查询要么查询分组条件列，要么是其它列的聚合函数——不能直接查询其它列。 |

练习：按照性别分组，查询每组中工资的最大值、最小值、平均值

SELECT sex, MAX(salary), MIN(salary), AVG(salary)

FROM emp

GROUP BY sex;

5.MySQL复杂查询 —— 跨表/多表查询 —— 了解

查询结果集中的数据来自于多个不同的表。

示例：查询每个员工的姓名及其所在部门的名称

SELECT ename, dname

FROM emp, dept ; #笛卡尔积！错误！

-----------------------------------------------------------

SELECT ename, deptId, did, dname

FROM emp, dept

WHERE deptId = did ; #跨表查询必须有连接条件

注意：上述写法是SQL-92版本中的写法。存在一定的缺陷：

(1)无法显示没有部门的员工 —— KING

(2)无法显示没有员工的部门 —— 测试部

SQL-99标准对此进行了改变，但是SQL变得很复杂，有四种写法：

**(1)内连接查询：inner join ... on ...**

SELECT ename, dname

FROM emp INNER JOIN dept

ON deptId=did ;

#内连接的效果与SQL-92标准完全一样

**(2)左外连接查询： left outer join ... on ...**

SELECT ename, dname

FROM emp LEFT OUTER JOIN dept

ON deptId=did ; #13+1

#可以显示出“左侧表”中所有的记录！即使右侧表中没有对应的记录

**(3)右外连接查询：right outer join ... on ...**

SELECT ename, dname

FROM emp RIGHT OUTER JOIN dept

ON deptId=did ; #13+1

#可以显示出“右侧表”中所有的记录！即使左侧表中没有对应的记录

**(4)全外连接查询：full join ... on ...**

SELECT ename, dname

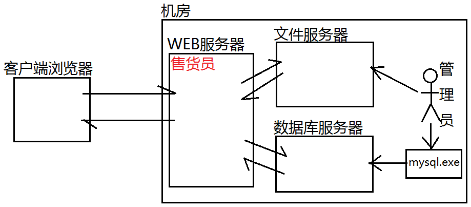
FROM emp FULL JOIN dept

ON deptId=did ; #13+1+1

#可以显示出“右侧表”和“右侧表”中所有的记录！—— MySQL不支持

|  |
| --- |
| 课外小知识：如何解决MySQL不支持全连接的问题 —— 使用结果集的合并  select ename, salary from emp\_us ;  select ename, salary from emp\_cn ; |
| 结果集的合并：UNION，把两条查询语句的结果合并为一个大结果  (select ename, salary from emp\_us)  UNION  (select ename, salary from emp\_cn) ;  两个结果集中的相同数据只显示一次 |
| (select ename, salary from emp\_us)  UNION ALL  (select ename, salary from emp\_cn) ;  两个结果集中的相同数据各自显示 |
| 练习：查询出员工姓名及其所在部门的名字，要求必须显示出所有的员工和所有的部门：左外连接合并上右外连接  (  SELECT ename, dname  FROM emp LEFT OUTER JOIN dept  ON deptId=did  )  UNION  (  SELECT ename, dname  FROM emp RIGHT OUTER JOIN dept  ON deptId=did  ) ; |

6.WEB服务器



WEB服务器：负责接收客户端请求，查找需要的数据/文件，给客户端以响应。Web服务器的分类：

**(1)静态WEB服务器**：提供的内容任何时间任何人访问都是一样的

常见的静态WEB内容：HTML/CSS/JS/图片/视频/Flash...

**(2)动态WEB服务器**：提供的内容不同时间不同人访问可能不同，一般都需要访问数据库、复杂运算、访问其它服务器。

常见的动态WEB技术：

JSP = HTML + JAVA

PHP = HTML + php

ASP.NET = HTML + C#

Node.js

Python

.....

7.PHP简介

1994年诞生，Personal Home Page语言，用于记录自己的网页的访问统计。后来开源给社区，逐渐流行；1999年，改名为 PHP is Hypertext Preprocessor。特点：开源、简单、跨平台、占用资源少。黄金组合：

LAMP = Linux + Apache + MySQL + PHP

8.搭建PHP开发和运行环境

(1)服务器端：下载并安装PHP服务器——XAMPP

c:/xampp/apache/bin/httpd.exe 静态WEB服务器

c:/xampp/php/php.exe PHP解释器

(2)服务器端：启动WEB服务器，保证端口被打开(默认是80)

(3)服务器端：编写网页，保存在WEB服务器的特定目录下

c:/xampp/htdocs/

记得删除此目录下的index.php文件！

-----------------------------------------------

(4)客户端：打开浏览器，访问远程的WEB服务器即可

127.0.0.1:80

课后练习：

(1)删除上面笔记中所有的SQL代码，根据注释重新写出示例代码——强化练习！

(2)创建3.php，在此网页中向客户端输出如下内容：

姓名：丁当

电话：13501234567

邮箱：dingdang@tedu.cn

————————————————————

版权所有® DINGDANG.COM

(3)创建4.php，在此网页中向客户端输出如下内容：

1\*1=1

2\*1=2 2\*2=4

3\*1=3 3\*2=6 3\*3=9

4\*1=4 4\*2=8 4\*3=12 4\*4=16

......