复习：

软件过程

软件定义期

可行性研究阶段

需求分析阶段

软件开发期

概要设计阶段 模块、数据

详细设计阶段

编码实现阶段

测试阶段

软件维护期

项目部署阶段

项目维护阶段

RDBMS分为两部分：

服务器端：负责管理维护数据

客户端：用于连接到服务器，并提交增删改查命令

客户端连接服务器的两种模式：

交互模式： mysql -uroot

脚本模式： mysql -uroot < d:/x.sql

MySQL中的列类型（3大类8小种）：

数值类型：

tinyint、smallint、int、bigint

float、double、decimal

bool

日期/时间类型

date、time、datetime

字符串类型

char、varchar、text

MySQL中的列约束：

主键约束：PRIMARY KEY

非空约束：NOT NULL

唯一约束：UNIQUE 影响插入/修改效率

默认值约束： DEFAULT

检查约束：CHECK 影响插入/修改效率

外键约束：FOREIGN KEY... REFERENCES... 影响插入/修改效率

今日目标：

(1)补充小知识点

(2)SELECT - 简单查询（必须掌握）

(3)SELECT - 复杂查询（难度较大）

1.项目中如何存储日期时间数据

大体有三种方式

**(1)VARCHAR存储**：不足：不便于比较大小，格式不灵活

**(2)DATE/TIME/DATETIME存储**：不足：不便于实现国际化，不同的编程语言支持程度不同

**(3)BIGINT存储**：表示距离计算机元年的毫秒值，任何编程语言都可以把大数字转换为日期时间

中国：2017-10-25

美国：10-25-2017

欧洲：25/10/2017

i18n：internationalization，国际化，实现了国际化的项目应该对中国人显示中国人的习惯格式，对美国人显示美国人的习惯格式.......

计算机中如何存储日期时间：一个很大的数字，表示目标日期距离“计算机元年(1970-1-1 0:0:0 GMT)”经过了多少毫秒:

|  |  |
| --- | --- |
| 数字 | 代表的时间 |
| 0 | 1970-1-1 0:0:0 |
| 1000 | 1970-1-1 0:0:1 |
| -1000 | 1969-12-31 23:59:59 |
| 1000\*60 | 1970-1-1 0:1:0 |
| 1000\*60\*60 | 1970-1-1 1:0:0 |
| 1000\*60\*60\*24 | 1970-1-2 0:0:0 |
| 1000\*60\*60\*24\*365 | 1971-1-1 0:0:0 |

2.MySQL中使用自增列

id INT PRIMARY KEY AUTO\_INCREMENT

**自增列：**只能用于整数列，且必须是主键列。自增列无需手工赋值，会自动采用1/2/3....数列，在当前最大值基础上+1。

注意：SQL标准中没有此关键字，它是MySQL所专有的！

练习：编写jd.sql文件，创建数据库jd，包含如下两个表：

dept( did(pk), dname )

10 研发部

20 市场部

30 运营部

40 测试部

emp( eid(pk), ename, salary, birthday, deptId(fk) )

插入10位员工信息，分属于10/20/30三个部门

1.简单查询 —— 只查询特定的列

示例：查询出所有员工的姓名、工资、和编号

SELECT ename, salary, eid FROM emp;

练习：查询出所有的员工姓名、性别、生日、姓名

SELECT ename, sex, birthday, ename FROM emp;

2.简单查询 —— 查询所有的列

示例：查询员工的所有信息

SELECT \* FROM emp;

3.简单查询 —— 给列取别名

示例：查询出员工姓名，所在部门编号，要求列名用中文呈现

SELECT ename AS 姓名, deptId `部门 编号` FROM emp;

注意：给列取别名用AS关键字，且可以省略；别名中若有空格，需要用` `括起来。

练习：查询出所有员工的编号、姓名、工资、生日，列名全部显示为中文

SELECT eid AS 编号, ename AS 姓名, salary 工资, birthday 生日 FROM emp;

练习：查询出所有员工的编号、姓名、工资、生日，列名全部显示为一个英文字符

SELECT eid AS i, ename AS n, salary s, birthday b

FROM emp;

4.简单查询 —— 只显示不同的记录

示例：显示出哪些部门编号下有员工

SELECT DISTINCT deptId

FROM emp;

说明： distinct：不同的

练习：查询出公司中有哪些性别的员工

SELECT DISTINCT sex FROM emp;

5.简单查询 —— 在查询时执行计算

示例：计算2/3的商

SELECT 2/3;

示例：查询出每个员工的姓名及其年薪

SELECT ename, salary, salary\*12

FROM emp;

练习：老板想查看一下若给每人每月加薪500，年终再发2000年终奖，则每人一年共可以拿到多少钱

SELECT ename, (salary+500)\*12+2000 AS 年薪

FROM emp;

练习：假设每人每月要交社保500，剩余的月薪再交外加30%的个人所得税，查询出每人每月到手工资是多少

SELECT ename, (salary-500)\*(1-0.3)

FROM emp;

6.简单查询 —— 查询结果集的排序

示例：查询出所有员工信息，按工资由小到大排列

SELECT \*

FROM emp

ORDER BY salary ; #ASC，ascendant 升序

示例：查询出所有员工信息，按工资由大到小排列

SELECT \*

FROM emp

ORDER BY salary DESC; #descendant 降序

练习：查询出所有员工信息，按年龄由小到大排序

SELECT \* FROM emp

ORDER BY birthday DESC;

练习：查询出所有员工信息，按姓名降序排序

SELECT \* FROM emp

ORDER BY ename DESC;

练习：查询出所有员工信息，按工资降序排列，若两人工资相同，再按姓名升序排列

SELECT \* FROM emp

ORDER BY salary DESC, ename ASC;

练习：查询出所有员工信息，按员工姓名升序排列，要求所有女员工排在所有男员工之前

SELECT \* FROM emp

ORDER BY sex, ename;

7.简单查询 —— 条件查询

示例：查询出编号为5的员工所有信息

SELECT \* FROM emp

WHERE eid=5 ;

练习：查询出姓名为KING的员工的编号、工资、生日

SELECT eid, birthday, salary

FROM emp

WHERE ename='KING';

练习：查询出20号部门下所有员工的信息

SELECT \*

FROM emp

WHERE deptId=20;

练习：查询出所有的男员工信息

SELECT \*

FROM emp

WHERE sex=1;

示例：查询出工资大于等于5000的员工所有信息

SELECT \* FROM emp

WHERE salary >= 5000 ;

|  |
| --- |
| = > < >= <= !=或<> |

练习：查询出1991-1-1之后出生的员工所有信息

SELECT \* FROM emp

WHERE birthday>'1991-1-1';

练习：查询出不在10号部门的所有员工信息

SELECT \* FROM emp

WHERE deptId != 10;

|  |
| --- |
| IS NULL、 IS NOT NULL、 AND、 OR、 NOT |

练习：查询出暂时没有部门的员工信息

SELECT \* FROM emp

WHERE deptId IS NULL;

#WHERE deptId = NULL ; #NULL不能用=或!=判定

练习：查询出有明确部门的员工信息

SELECT \* FROM emp

WHERE deptId IS NOT NULL;

#HERE deptId!=NULL; #NULL不能用=或!=判定

练习：查询出工资大于6000的女员工所有信息

SELECT \* FROM emp

WHERE (salary>6000) AND (sex=0);

练习：查询出工资在5000~5999之间的员工信息

SELECT \* FROM emp

WHERE salary>=5000 AND salary<=5999;

#WHERE 5000<=salary<=5999; 错误写法！

该语句等价于

SELECT \* FROM emp

WHERE salary BETWEEN 5000 AND 5999;

练习：查询出工资小于4000和大于8000的员工信息

SELECT \* FROM emp

WHERE salary<4000 OR salary>8000;

练习：查询出在1991年出生的员工信息

SELECT \* FROM emp

WHERE birthday>='1991-1-1' AND birthday<='1991-12-31';

等价于

SELECT \* FROM emp

WHERE birthday BETWEEN '1991-1-1' AND '1991-12-31';

练习：查询出在1990年之前和1993之后出生的员工信息

SELECT \* FROM emp WHERE birthday<'1990-1-1' OR birthday>'1993-12-31';

练习：查询出在10和30号部门的员工信息

SELECT \* FROM emp WHERE deptId=10 OR deptId=30;

练习：查询出在10、30、50、80号部门的员工信息

SELECT \* FROM emp WHERE deptId=10 OR deptId=30 OR deptId=50 OR deptId=80;

该语句等价于

SELECT \* FROM emp WHERE deptId IN (10, 30, 50, 80);

练习：查询出不在10和30号部门的员工信息

SELECT \* FROM emp WHERE deptId!=10 AND deptId!=30;

该语句等价于

SELECT \* FROM emp WHERE deptId NOT IN (10, 30);

8.简单查询 —— 模糊条件查询

示例：查询出姓名中包含字母E的员工所有信息

SELECT \* FROM emp

WHERE ename LIKE '%E%';

#WHERE ename='%E%'; #错误写法！

|  |
| --- |
| SQL通配符： 下面两个通配符必须与LIKE组合应用  % 匹配任意多个任意字符  \_ 匹配一个任意字符 |

练习：查询出姓名以E结尾的员工所有信息

SELECT \* FROM emp WHERE ename LIKE '%E';

练习：查询出姓名中倒数第二个字符是E的员工所有信息

SELECT \* FROM emp WHERE ename LIKE '%E\_';

9.简单查询 —— 分页查询

分页查询：若数据库中的满足条件的记录行数太多，一般会采取“一页一页”的方式逐步展示给用户。

不同的数据库实现分页查询语法各不相同，MySQL中的分页查询时最简单的！形如：

SELECT ...

FROM ...

WHERE ...

ORDER BY ...

**LIMIT start, count** ;

start：从哪一行开始读取数据，数据库中的第一行记录称为第0行

count：一次最多可以读取的行数

假设：每一页最多呈现6条记录(称为“页面大小”pageSize)

第1页： SELECT .... LIMIT 0, 6 ;

第2页： SELECT .... LIMIT 6, 6 ;

第3页： SELECT .... LIMIT 12, 6 ;

第4页： SELECT .... LIMIT 18, 6 ;

....

第n页： SELECT .... LIMIT (n-1)\*pageSize, pageSize ;

练习：查询出所有员工信息，只显示第1页，每页最多显示6条

SELECT \* FROM emp LIMIT 0, 6;

练习：查询出所有员工信息，只显示第2页，每页最多显示6条

SELECT \* FROM emp LIMIT 6, 6;

练习：查询出所有员工信息，只显示第3页，每页最多显示6条

SELECT \* FROM emp LIMIT 12, 6;

练习：查询出所有员工信息，只显示第4页，每页最多显示6条

SELECT \* FROM emp LIMIT 18, 6;

练习：查询出10号部门的所有员工信息，只显示第1页，每页最多显示5条记录

SELECT \* FROM emp

WHERE deptId=10

LIMIT 0, 5;

课后自学：

高阶面试题：MySQL中的存储引擎是什么？有哪些？

baike.baidu.com

课后练习：

(1)删除今天课堂笔记中的所有示例和练习中的代码，自己根据注释文字写出所有的SQL语句 —— 经典练习强化！

(2)根据“学子商城数据库结构”设计图及说明文档，编写.sql文件，写出创建库和表所需的所有语句。