复习

ASCII/Latin-1/GB2312/GBK/BIG5

Unicode UTF-8/UTF-16/UTF-32

脚本文件/客户端连接服务器端/服务器端创建数据库

列类型

数值型

TINYINT/SMALLINT/INT/BIGINT/

FLOAT/DOUBLE/DECIMAL/BOOL

日期时间型

DATE/TIME/DATETIME

字符串型

VARCHAR/CHAR/TEXT

1.列约束

(1)主键约束 —— PRIMARY KEY

(2)非空约束 —— NOT NULL

声明了非空约束的列上禁止插入NULL值

(3)唯一约束 —— UNIQUE

声明了唯一约束的列上不允许插入重复的值，但允许插入NULL，甚至多个NULL

(4)检查约束 —— CHECK

可以对要插入的数据进行自定义的验证

CREATE TABLE student(

score TINYINT CHECK(score>=0 AND score<=100)

);

mysql不支持检查约束，会降低数据的插入速度，增加服务器的压力。

(5)默认值约束 —— DEFAULT

可以使用DEFAULT关键字设置默认值，有两种使用方式

INSERT INTO laptop VALUES(8,’小新’,**DEFAULT**…);

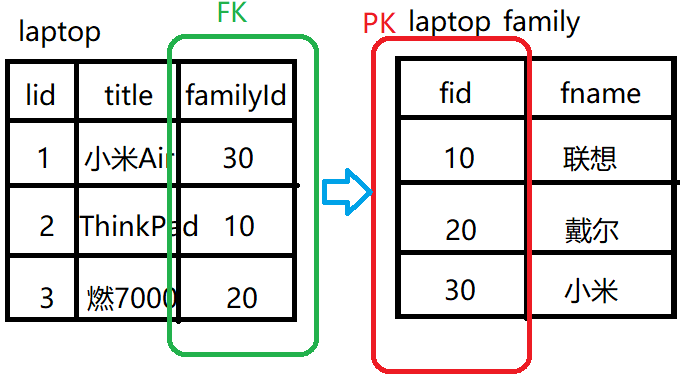
INSERT INTO laptop(lid,title) VALUES(8,’小新’);

其它的列全部使用默认值

练习: 设置分类表中，笔记本的数量默认值为1，分别使用两种方式运用默认值。

(6)外键约束 —— FOREIGN KEY

声明了外键约束的列，取值必须在另一个表的主键列上出现过，两者的列类型保持一致，允许使用NULL，甚至多个NULL。



2.mysql中的自增列

AUTO\_INCREMENT: 自动增长，假如一个列声明了自增列，无需手动赋值，直接赋值为NULL，会获取当前的最大值，然后加1

**注意：**

只适用于整数型的主键列上

自增列允许手动赋值

练习：编写脚本文件01\_tedu.sql，先丢弃在创建数据库tedu，设置编码utf8，进入该数据库；创建保存部门数据的表dept，包含(did, dname部门名称)，插入以下数据

10 研发部 20 市场部 30 运营部 40 测试部

创建保存员工数据的表emp，包含eid，ename姓名，sex性别，birthday生日，工资salary，deptId 所属部门编号，插入若干条数据

3.简单查询

(1)查询特定的列

示例:查询所有员工的编号、姓名

SELECT eid,ename FROM emp;

练习：查询所有员工的姓名、性别、生日、工资

SELECT ename,sex,birthday,salary FROM emp;

(2)查询所有的列

SELECT \* FROM emp;

SELECT eid,ename,sex,birthday,salary,deptId FROM emp;

(3)给列起别名

示例：查询所有员工的姓名和工资，使用汉字别名

SELECT ename AS 姓名,salary AS 工资 FROM emp;

练习：查询所有员工的姓名、性别、生日，编号，使用汉字别名

SELECT ename AS 姓名,sex AS 性别,birthday 生日,eid 编号 FROM emp;

练习：查询员工的编号和姓名，使用一个字母作为别名

SELECT eid a,ename b FROM emp;

|  |
| --- |
| 在起别名的时候，AS关键字可以省略，保留空格 |

(4)合并相同的项/显示不同的记录

示例: 查询出员工都分布在哪些部门

SELECT DISTINCT deptId FROM emp;

练习: 查询公司都有哪些性别的员工

SELECT DISTINCT sex FROM emp;

(5)查询时执行计算

示例: 计算18+23-56/19+38\*21

SELECT 18+23-56/19+38\*21;

练习: 查询出所有员工的姓名及其年薪

SELECT ename, salary\*12 FROM emp;

练习: 假设每个员工的工资增加500，年终奖5000，查询出所有员工的姓名及其年薪，给列起汉字别名

SELECT ename 姓名, (salary+500)\*12+5000 年薪 FROM emp;

(6)查询结果集排序

示例:查询所有的部门，结果集按照部门编号降序排列

SELECT \* FROM dept ORDER BY did DESC; #descendant

示例:查询所有的部门，结果集按照部门编号升序排列

SELECT \* FROM dept ORDER BY did ASC;#ascendant

练习:查询所有的员工，结果集按照工资升序排列

SELECT \* FROM emp ORDER BY salary ASC;

练习:查询所有的员工，结果集按照年龄从小到大排序

SELECT \* FROM emp ORDER BY birthday DESC;

练习:查询所有的员工，结果集按照姓名升序排列

SELECT \* FROM emp ORDER BY ename;

练习:查询所有的员工，结果集按照工资降序排列，如果工资相同，按照姓名升序排列

SELECT \* FROM emp ORDER BY salary DESC,ename;

练习:查询出所有员工，结果及按照生日降序排列，女员工显示在前边。

SELECT \* FROM emp ORDER BY sex,birthday DESC;

|  |
| --- |
| ORDER BY可以按照数值、日期时间、字符串排序  默认按照升序排列ASC |

(7)条件查询

示例: 查询出姓名叫king的员工所有列

SELECT \* FROM emp WHERE ename='king';

练习: 查询出编号为5的员工所有列

SELECT \* FROM emp WHERE eid=5;

练习: 查询出20号部门下的员工所有列

SELECT \* FROM emp WHERE deptId=20;

练习: 查询出所有的女员工

SELECT \* FROM emp WHERE sex=0;

练习: 查询出工资在5000以上的员工有哪些

SELECT \* FROM emp WHERE salary>=5000;

|  |
| --- |
| 比较运算符: > >= < <= = !=(不等于) |

练习: 查询出1991-1-1后出生的员工有哪些

SELECT \* FROM emp WHERE birthday>'1991-1-1';

练习: 查询出不在10号部门的员工有哪些

SELECT \* FROM emp WHERE deptId!=10;

练习: 查询出没有明确部门的员工有哪些

SELECT \* FROM emp WHERE deptId IS NULL;

练习: 查询出有明确部门的员工有哪些

SELECT \* FROM emp WHERE deptId IS NOT NULL;

练习: 查询出工资在6000以上的男员工有哪些

SELECT \* FROM emp WHERE salary>6000 AND sex=1;

练习: 查询出工资在5000~7000之间的员工有哪些

SELECT \* FROM emp WHERE salary>=5000 AND salary<=7000;

SELECT \* FROM emp WHERE salary BETWEEN 5000 AND 7000;

练习: 查询出工资在5000以下和7000以上的员工有哪些

SELECT \* FROM emp WHERE salary<5000 OR salary>7000;

SELECT \* FROM emp WHERE salary NOT BETWEEN 5000 AND 7000;

练习:查询出1990年之前和1993年之后出生的员工有哪些

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

练习：查询出1993年出生员工有哪些

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

练习：查询出20号部门和30号部门的员工有哪些

SELECT \* FROM emp WHERE deptId=20 OR deptId=30;

SELECT \* FROM emp WHERE deptId IN(20,30);

练习: 查询不在20号部门和30号部门的员工有哪些

SELECT \* FROM emp WHERE deptId NOT IN(20,30);

|  |
| --- |
| IS NULL / IS NOT NULL  AND / OR  BETWEEN .. AND.. / NOT BETWEEN.. AND..  IN( ) / NOT IN( ) |

(8)模糊条件查询

示例: 查询出姓名中含有字母e的员工有哪些

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

练习: 查询出姓名中以e结尾的员工有哪些

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

练习: 查询出姓名中倒数第2个字符为e的员工有哪些

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

|  |
| --- |
| % 可以匹配任意多个字符 >=0  \_ 可以匹配任意1个字符 =1  以上两个匹配字符必须结合LIKE使用 |

(9)分页查询

假如查询的结果集有太多的数据，一次性显示不完，可以使用分页查询。

需要有两个条件: 当前的页码、每页的数据量

SELECT \* FROM emp LIMIT start,count;

start: 是一个数字，表示从哪一条数据开始查询。

count: 是每页的数据量

|  |
| --- |
| start = (当前的页码-1)\*每页的数据量 |

假设每页显示5条记录

第1页: SELECT \* FROM emp LIMIT 0,5;

第2页: SELECT \* FROM emp LIMIT 5,5;

第3页: SELECT \* FROM emp LIMIT 10,5;

练习：假设每页显示6条记录，查询前3页

第1页: SELECT \* FROM emp LIMIT 0,6;

第2页: SELECT \* FROM emp LIMIT 6,6;

第3页: SELECT \* FROM emp LIMIT 12,6;

|  |
| --- |
| 注意事项：LIMIT后的两个值不能添加引号，只能是数值。 |

课后任务

(1)复习今天内容，删除代码保留注释，重新编码SQL语句

(2)练习

查询出工资在8000以上的女员工的姓名、性别、工资，结果集按照工资降序排列，获取前两个。

(3)查看学子商城数据库设计文档。