复习

DDL 定义数据 CREATE / DROP/ALTER

DML 操作数据 INSERT/UPDATE/DELETE

DQL 查询数据 SELECT

DCL 控制权限 GRANT/REVOKE

编码 ASCII/Latin-1/GB2312/GBK/Unicode

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/BOOL

FLOAT/DOUBLE/DECIMAL(M,D)

日期时间型 DATE/TIME/DATETIME

字符串型 VARCHAR(M)/CHAR(M)/TEXT

列约束

主键约束

NULL

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，不允许使用NULL

(3)唯一约束——UNIQUE

声明了唯一约束的列上不能插入重复的值，允许插入NULL，而且允许插入多个NULL

(4)默认值约束——DEFAULT

可以使用DEFAULT关键字声明默认值，有两种使用方式

INSERT INTO laptop VALUES(1, DEFAULT,…);

INSERT INTO laptop(lid,price) VALUES(2,3000); 给特定的列插入值，其它的列全部使用默认值

练习：设置笔记本分类表中，笔记本的数量默认值为1，使用两种方式插入数据。

(5)检查约束——CHECK

检查约束可以对插入的数据进行自定义的验证

CREATE TABLE student(

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

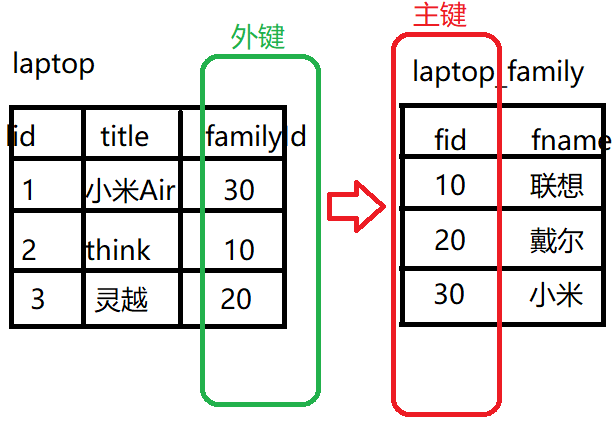
);

Mysql不支持检查约束，会降低数据的插入速度

(6)外键约束——FOREIGN KEY

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

FOREIGN KEY(列) REFERENCES 数据表(主键列)



2.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，赋值为NULL，会自动获取当前的最大值，然后加1插入。

注意事项：

只适用于整数型的主键列

允许手动赋值

练习：编写脚本文件01\_tedu.sql，创建数据库tedu，进入该数据库

创建保存部门数据的表dept，包含(did, dname)

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，sex，birthday，salary工资，deptId 所属部门编号)

插入15条数据

3.简单查询

(1)查询特定的列

示例: 查询所有员工的姓名、性别

SELECT ename,sex FROM emp;

练习: 查询所有员工的编号、姓名、生日、工资

SELECT eid,ename,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：查询所有员工的编号，姓名，性别，生日，使用汉字别名

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

练习：查询所有员工的编号和姓名，使用一个字母别名

SELECT eid a,ename b FROM emp;

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

(4)显示不同的记录/合并相同的项

示例：查询出公司都有哪些性别的员工

SELECT DISTINCT sex FROM emp;

练习：查询出员工都分布在哪些部门

SELECT DISTINCT deptId FROM emp;

(5)查询时执行计算

示例：2+3-5\*4.2/8.8+13

SELECT 2+3-5\*4.2/8.8+13;

练习：查询所有员工的姓名及其年薪

SELECT ename,salary\*12 FROM emp;

练习：假如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，给列使用汉字别名。

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升序排列

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

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

练习：查询所有的员工，结果集按照工资降序排列

SELECT \* FROM emp ORDER BY salary DESC;

练习：查询所有的员工，结果集按照工资升序排序

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序

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，按照姓名排序

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

练习：查询所有的员工，结果集按照性别升序排列，如果性别相同，按照生日降序排列

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  默认是升序排列(ASC) |

(7)条件查询

示例：查询出编号为5的员工

SELECT \* FROM emp WHERE eid=5;

练习：查询出姓名为King的员工的编号，姓名，生日，工资

SELECT eid,ename,birthday,salary FROM emp WHERE ename='king';

练习：查询出20号部门下的员工，结果集按照工资降序排列

SELECT \* FROM emp WHERE deptId=20 ORDER BY salary DESC;

练习：查询工资在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

SELECT \* FROM emp WHERE deptId IS NULL;

练习：查询出有明确部门的员工

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6000以上和8000以下的所有员工

SELECT \* FROM emp WHERE salary>=6000 AND salary<=8000;

SELECT \* FROM emp WHERE salary BETWEEN 6000 AND 8000;

练习：查询出工资在6000以下以及8000以上的员工 OR

SELECT \* FROM emp WHERE salary<6000 OR salary>8000;

SELECT \* FROM emp WHERE salary NOT BETWEEN 6000 AND 8000;

练习：查询出1990年之前以及1993年之后出生的员工

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

SELECT \* FROM emp WHERE birthday NOT BETWEEN '1990-1-1' AND '1993-12-31';

练习：查询出1993年出生的员工有哪些

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

SELECT \* FROM emp WHERE birthday BETWEEN '1993-1-1' AND '1993-12-31';

练习：查询出20号和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：查询出不在20号和30号部门的员工

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

|  |
| --- |
| WHERE  AND / OR  (NOT) BETWEEN AND  IS (NOT) NULL  (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

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

练习：查询出姓名中以e结尾的员工

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

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

|  |
| --- |
| % 任意多个字符 >=0  \_ 任意一个字符 =1  以上两个字符要结合LIKE关键字使用 |

(9)分页查询

假如查询的结果集有太多的数据，一次显示不完，可以分页显示。

需要有两个条件：当前的页码、每页的数据量

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

SELECT \* FROM emp LIMIT start,count;

start: 开始的值

count: 每页的数据量

假如每页显示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;

第4页：SELECT \* FROM emp LIMIT 15,5;

第5页：SELECT \* FROM emp LIMIT 20,5;

练习：假如每页显示6条记录，写出查询前3页的SQL语句

SELECT \* FROM emp LIMIT 0,6;

SELECT \* FROM emp LIMIT 6,6;

SELECT \* FROM emp LIMIT 12,6;

注意： LIMIT后的两个值必须是数值型。

课后任务

(1)复习今天内容，删除代码保留注释，重新编写代码。

(2)练习

查询出工资在8000以上的女员工的姓名、性别、生日、工资，结果集按照工资的降序排列，取前3条数据。