

## 课程介绍

# 学习目标



## 数据添加

INSERT语法、INSERT方言语法、INSERT子查询、IGNORE关键字



## 数据修改

UPDATE语法、UPDATE表连接



## 数据删除

DELETE语法、DELETE表连接



# 数据操作语言：INSERT语句

# INSERT语句

- ◆ INSERT语句可以向数据表写入记录，可以是一条记录，也可以是多条记录

```
INSERT INTO 表名 ( 字段1, 字段2, ..... )  
VALUES ( 值1, 值2, ..... );
```

```
INSERT INTO 表名 ( 字段1, 字段2, ..... )  
VALUES ( 值1, 值2, ..... ), ( 值1, 值2, ..... );
```

# INSERT语句方言

- ◆ MYSQL的INSERT语句还有一种方言语法

```
INSERT INTO 表名 SET 字段1=值1, 字段2=值2, .....;
```



# IGNORE关键字

- ◆ IGNORE关键字会让INSERT只插入数据库不存在的记录

```
INSERT [ IGNORE ] INTO 表名 ..... ;
```

```
INSERT IGNORE INTO t_dept(deptno,dname,loc)  
VALUES (40,"技术部","北京");
```

```
[SQL]INSERT IGNORE INTO t_dept(deptno,dname,loc)  
VALUES(40,"技术部","北京");  
受影响行: 0  
时间: 0.001s
```

# INSERT中的子查询

- ◆ INSERT语句中可以包含子查询语句，代表把子查询的内容写入到数据表
- ◆ 把人数超过5人的部门记录拷贝到新的部门表

```
CREATE TABLE t_dept_new LIKE t_dept;  
  
INSERT INTO t_dept_new(deptno,dname,loc)  
(SELECT d.deptno,d.dname,d.loc FROM t_dept d JOIN  
(SELECT deptno FROM t_emp GROUP BY deptno  
HAVING COUNT(*)>=5) temp  
ON d.deptno = temp.deptno);
```

# 数据操作语言：UPDATE语句（一）



# UPDATE语句

- ◆ UPDATE语句用于修改表的记录

```
UPDATE [ IGNORE ] 表名  
SET 字段1=值1, 字段2=值2, .....  
[ WHERE 条件1, 条件2, ..... ]  
[ ORDER BY ..... ]  
[ LIMIT ..... ] ;
```

# 课堂练习1

- ◆ 把每个员工的编号和上司的编号+1，用ORDER BY子句完成
- ◆ 把月收入前三名的员工底薪减100元，用LIMIT子句完成

## 课堂练习2

- ◆ 把10部门中，工龄超过20年的员工，底薪增加200元
- ◆ 把ALLEN调往RESEARCH部门，职务调整为ANALYST

# UPDATE语句的表连接（一）

- ◆ 因为WHERE子句中的子查询效率非常低，所以我们可以利用表连接的方式来改造UPDATE语句

```
UPDATE 表1 JOIN 表2 ON 条件  
SET 字段1 = 值1 , 字段2 = 值2 , ..... ;
```

- ◆ 表连接的UPDATE语句可以修改多张表的记录

## UPDATE语句的表连接（二）

- ◆ UPDATE语句的表连接可以演变成下面的样子

```
UPDATE 表1, 表2
```

```
SET 字段1 = 值1, 字段2 = 值2, .....
```

```
WHERE 连接条件;
```

## 课后作业

- ◆ 把底薪低于公司平均底薪的员工，底薪增加150元

## 数据操作语言：UPDATE语句（二）

## 课后作业

- ◆ 把底薪低于公司平均底薪的员工，底薪增加150元



## UPDATE语句的表连接（三）

- ◆ UPDATE语句的表连接既可以是内连接，又可以是外连接

```
UPDATE 表1 [ LEFT | RIGHT ] JOIN 表2 ON 条件  
SET 字段1 = 值1 , 字段2 = 值2 , ..... ;
```

- ◆ 把没有部门的员工，或者SALES部门低于2000元底薪的员工，都调往20部门

# 数据操作语言：DELETE语句（一）

# DELETE语句

◆ DELETE语句用于删除记录，语法如下：

```
DELETE [ IGNORE ] FROM 表名  
[ WHERE 条件1, 条件2, ..... ]  
[ ORDER BY ..... ]  
[ LIMIT ..... ] ;
```

# 课堂练习1

- ◆ 删除10部门中，工龄超过20年的员工记录
- ◆ 删除20部门中工资最高的员工记录

## 课堂练习2

- ◆ 删除SALES部门和该部门的全部员工记录
- ◆ 删除每个低于部门平均底薪的员工记录

# DELETE语句的表连接（一）

- ◆ 因为相关子查询效率非常低，所以我们可以利用表连接的方式来改造DELETE语句

```
DELETE 表1, ..... FROM 表1 JOIN 表2 ON 条件  
[ WHERE 条件1, 条件2, ..... ]  
[ ORDER BY ..... ]  
[ LIMIT ..... ] ;
```

# 课后作业

- ◆ 删除员工KING和他的直接下属的员工记录，用表连接实现。



## 数据操作语言：DELETE语句（二）



## 课后作业

- ◆ 删除员工KING和他的直接下属的员工记录，用表连接实现。



## DELETE语句的表连接（二）

- ◆ DELETE语句的表连接既可以是内连接，又可以是外连接

```
DELETE 表1 , ..... FROM 表1 [ LEFT | RIGHT ] JOIN 表2  
ON 条件 ..... ;
```

- ◆ 删除SALES部门的员工，以及没有部门的员工。

# 快速删除数据表全部记录

- ◆ DELETE语句是在事务机制下删除记录，删除记录之前，先把将要删除的记录保存到日志文件里，然后再删除记录。
- ◆ TRUNCATE语句在事务机制之外删除记录，速度远超过DELETE语句

```
TRUNCATE TABLE 表名 ;
```

## 课程总结

## 技能清单

技能知识1

☐

技能知识2

☐

技能知识3

☐

技能知识4

☐

掌握INSERT语法



掌握UPDATE语法



掌握DELETE语法



掌握数据更新和删除中的表连接语法

# 知识体系

