

# 创建和管理表

# 目 标

通过本章学习，您将可以：

- 创建数据库
- 创建表
- 描述各种数据类型
- 修改表的定义
- 删除，重命名和清空表

# 创建数据库

- 创建一个保存员工信息的数据库

- `create database employees;`

- 相关其他命令

- `show databases;` 查看当前所有数据库

- `use employees;` “使用” 一个数据库，使其作为当前数据库

# 命名规则

- 数据库名不得超过30个字符，变量名限制为29个
- 必须只能包含 A-Z, a-z, 0-9, \_ 共63个字符
- 不能在对象名的字符间留空格
- 必须不能和用户定义的其他对象重名
- 必须保证你的字段没有和保留字、数据库系统或常用方法冲突
- 保持字段名和类型的一致性, 在命名字段并为其指定数据类型的时候一定要保证一致性。假如数据类型在一个表里是整数, 那在另一个表里可就别变成字符型了

# CREATE TABLE 语句

- 必须具备：
  - CREATE TABLE 权限
  - 存储空间

```
CREATE TABLE [schema.] table  
              (column datatype [DEFAULT expr][, ...]);
```

- 必须指定：
  - 表名
  - 列名，数据类型，尺寸

# 创建表

- 语法

```
CREATE TABLE dept
      (deptno INT (2) ,
       dname   VARCHAR(14) ,
       loc     VARCHAR(13) ) ;
```

Table created.

- 确认

```
DESCRIBE dept
```

Field	Type	Null	Key	Default	Extra
id	int(2) 6B	YES		(NULL) OK	
name	var... 11B	YES		(NULL) OK	

# 常用数据类型

INT	使用4个字节保存整数数据
CHAR( <i>size</i> )	定长字符数据。若未指定，默认为1个字符，最大长度255
VARCHAR( <i>size</i> )	可变长字符数据，根据字符串实际长度保存， <b>必须指定长度</b>
FLOAT(M,D)	单精度，M=整数位+小数位，D=小数位。 $D \leq M \leq 255, 0 \leq D \leq 30$ ，默认 $M+D \leq 6$
DOUBLE(M,D)	双精度。 $D \leq M \leq 255, 0 \leq D \leq 30$ ，默认 $M+D \leq 15$
DATE	日期型数据，格式'YYYY-MM-DD'
BLOB	二进制形式的长文本数据，最大可达4G
TEXT	长文本数据，最大可达4G

创建表

```
CREATE TABLE emp (  
  #int类型, 自增  
  emp_id INT AUTO_INCREMENT,  
  #最多保存20个中英文字符  
  emp_name CHAR (20),  
  #总位数不超过15位  
  salary DOUBLE,  
  #日期类型  
  birthday DATE,  
  #主键  
  PRIMARY KEY (emp_id)  
);
```

Field	Type	Null	Key	Default	Extra
emp_id	int(11)	7B NO	PRI	(NULL)	OK auto_increment
emp_name	char(20)	8B YES		(NULL)	OK
salary	double	6B YES		(NULL)	OK
birthday	date	4B YES		(NULL)	OK



# 使用子查询创建表

- 使用 `AS subquery` 选项，**将创建表和插入数据结合起来**

```
CREATE TABLE table  
    [(column, column...)]  
AS subquery;
```

- 指定的列和子查询中的列要一一对应
- 通过列名和默认值定义列

# 使用子查询创建表

复制现有的表：

```
create table emp1 as select * from employees;  
create table emp2 as select * from employees where 1=2;  
--创建的emp2是空表。
```

# 使用子查询创建表举例

```
CREATE TABLE dept80
AS
  SELECT  employee_id, last_name,
          salary*12 ANNSAL,
          hire_date
  FROM    employees
  WHERE   department_id = 80;
```

Table created.

```
DESCRIBE dept80
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

34 rows selected

# ALTER TABLE 语句

使用ALTER TABLE 语句可以实现：

- 向已有的表中添加列
- 修改现有表中的列
- 删除现有表中的列
- 重命名现有表中的列

# 追加一个新列

新列

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE
149	Zlotkey	126000	29-JAN-00
174	Abel	132000	11-MAY-96
176	Taylor	103200	24-MAR-98

JOB_ID

追加一个新列



DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

```
ALTER TABLE dept80  
ADD job_id varchar(15);
```

# 修改一个列

- 可以修改列的数据类型，尺寸和默认值

```
ALTER TABLE dept80  
MODIFY last_name VARCHAR(30);  
Table altered.
```

```
ALTER TABLE dept80  
MODIFY salary double(9,2) default 1000;  
Table altered.
```

- 对默认值的修改只影响今后对表的修改

# 修改一个列

MySql:

添加单列:

ALTER TABLE 表名 ADD 列名 数据类型

添加多列:

ALTER TABLE 表名 ADD 列名1 数据类型1, Add 列名2 数据类型2

修改单列数据类型:

ALTER TABLE 表名 CHANGE COLUMN 列名 数据类型

同时修改多列数据类型:

ALTER TABLE 表名 CHANGE COLUMN 列名 数据类型, CHANGE COLUMN 列名 数据类型

删除单列:

ALTER TABLE 表名 DROP COLUMN 列名

删除多列:

ALTER TABLE 表名 DROP COLUMN 列名1, DROP COLUMN 列名2

同时添加和修改多列:

ALTER TABLE 表名 ADD 列名1 数据类型1, CHANGE COLUMN 列名 数据类型, DROP COLUMN 列名1

( COLUMN 关键字可以省略)

# 修改一个列

Oracle(PL / SQL):

添加单列:

```
ALTER TABLE 表名 ADD 列名 数据类型
```

添加多列:

```
ALTER TABLE 表名 ADD ( 列名1 数据类型1,列名2 数据类型2)
```

修改单列数据类型:

```
ALTER TABLE 表名 MODIFY 列名 数据类型
```

同时修改多列数据类型:

```
ALTER TABLE 表名 MODIFY ( 列名1 数据类型1,列名2 数据类型2)
```

删除单列:

```
ALTER TABLE 表名 DROP COLUMN 列名
```

删除多列:

```
ALTER TABLE 表名 DROP (列名1,列名2)
```

同时添加和修改多列:

```
ALTER TABLE 表名 ADD ( 列名1 数据类型1,列名2 数据类型2) MODIFY ( 列名3 数据类型3,列名4 数据类型4)
```



# 删除一个列

使用 DROP COLUMN 子句删除不再需要的列.

```
ALTER TABLE dept80  
DROP COLUMN job_id;  
Table altered.
```

# 重命名一个列

使用 `CHANGE old_column new_column dataType`  
子句重命名列

```
ALTER TABLE dept80  
CHANGE department_name dept_name varchar(15);  
Table altered.
```

# 删除表

- 数据和结构都被删除
- 所有正在运行的相关事务被提交
- 所有相关索引被删除
- DROP TABLE 语句不能回滚

```
DROP TABLE dept80;
```

```
Table dropped.
```

# 清空表

- **TRUNCATE TABLE** 语句：

- 删除表中所有的数据
- 释放表的存储空间

```
TRUNCATE TABLE detail_dept;  
Table truncated.
```

- TRUNCATE语句**不能回滚**
- 可以使用 DELETE 语句删除数据, 可以回滚
- 对比:

```
delete from emp2;  
select * from emp2;  
rollback;  
select * from emp2;
```

# 改变对象的名称

- 执行RENAME语句改变表, 视图的名称

```
ALTER table dept  
RENAME TO detail_dept;  
Table renamed.
```

- 必须是对象的拥有者