

1. 创建普通表

```
CREATE TABLE T2(TNO NUMBER(3),TNAME VARCHAR(12)) TABLESPACE USERS;
```

2. 创建临时表

临时表是一种特殊的表。当需要对某一（几个）表中的一批数据进行反复的操作时，通过为这批数据创建一个临时表可能会简化操作并且有可能提高效率。

```
CREATE GLOBAL TEMPORARY TABLE T3 ON COMMIT PRESERVE ROWS AS SELECT * FROM  
SCOTT.EMP WHERE JOB NOT LIKE 'SAL%';
```

临时表中存储的是会话私有数据，这些数据只在事务进行或会话期间存在。可通过如下子句来空间数据存在的周期：

ON COMMIT DELETE ROWS:说明数据行只有在事务中可见，也是默认值。

ON COMMIT PRESERVE ROWS:说明数据行在整个会话中可见。

3. 表的维护和管理

4. 非分区表的重组

5. 表的截断和删除

如果表中的数据已经不再需要时，可以使用 TRUNCATE TABLE 语句将他们全部删除掉（截断）。该语句是 DDL 语句。

```
TRUNCATE TABLE T2;
```

该语句有如下的特征：

- 删除表中所有的数据行，但保留表的结构。

- 对应的索引也被截断。

- 因为该语句为 DDL 语句，所以不会产生还原数据，所删除的数据也无法恢复。

- 该语句释放表所占用的磁盘空间。

- 并不触发（运行）表的删除触发器。

- 如果一个表正在被一个外键所引用，则该表不能截断。

DELETE 语句是 DML 语句，它有如下特征：

- 删除数据后，数据被写入回滚段。不提交的话数据可以恢复。commit 后数据才被清空。

- 不会释放之前占用的存储空间。

```
DROP TABLE 用户名.表名 [CASCADE CONSTRAINTS];
```

DROP TABLE 语句的特征：

- 删除表中所有的数据行和表的结构。

- 删除表的所有索引。

- 如果没有备份，所删除的表无法恢复。

- 该语句释放表所使用的 EXTENTS；

- 提交所有的挂起事务。

- 所有基于该表的视图 (views) 和别名 (synonyms) 依然保留但已无效。