# 一、数据库对象创建与管理

## 1. 表空间创建：

创建了一个名为ts\_names的表空间，初始大小为100MB，自动扩展每次增加10MB。

**> CREATE TABLESPACE ts\_names -- 表空间名字**

**> DATAFILE 'c:\ts\_names.dbf' -- 数据文件路径**

**> SIZE 100M -- 初始大小**

**> AUTOEXTEND ON -- 自动扩展**

**> NEXT 10M; -- 拓展大小**

## 用户创建与权限分配：

创建了一个用户fisher，密码为password，默认表空间为fisheries，并赋予了DBA权限。

**> CREATE USER fisher**

**> IDENTIFIED BY password**

**> DEFAULT TABLESPACE fisheries;**

**> GRANT DBA TO fisher;**

# 二、数据类型

## 1. 字符型：

1. **CHAR：固定长度，最多2000字节。**
2. **VARCHAR2：可变长度，最多4000字节。**
3. **LONG：大文本，最大2GB。**

## 2. 数值型：

1. **NUMBER：数值类型，如NUMBER(5)表示最大值为99999，NUMBER(5,2)表示最大值为999.99。**

## 3. 日期型：

1. **DATE：精确到秒。**
2. **TIMESTAMP：精确到秒的小数点后9位。**

## 4. 二进制型：

1. **CLOB：存储字符，最大4GB。**
2. **BLOB：存储二进制数据，最大4GB。**

# 三、表结构修改

## 1. 追加字段- ADD：

**> ALTER TABLE OWNERS ADD (REMARK VARCHAR2(20), OUTDATE DATE);**

## 2. 修改字段- MODIFY：

**> ALTER TABLE OWNERS MODIFY (REMARK VARCHAR2(50));**

## 3. 修改字段- RENAME：

**> ALTER TABLE OWNERS RENAME COLUMN REMARK TO COMMENT;**

## 4. 删除字段- DROP：

**> ALTER TABLE OWNERS DROP COLUMN COMMENT;**

## 5. 添加注释- COMMENT：

**>COMMENT ON COLUMN OWNERS.PASS\_TYPE IS 'COMMENT\_CONTENT';**

# 四、数据操作

## 1. 插入数据- INSERT：

**> INSERT INTO OWNERS (ID, NAME) VALUES (1, '张三丰');**

## 2. 更新数据- UPDATE：

**> UPDATE OWNERS SET ADDDATE = ADDDATE - 3 WHERE ID = 1;**

## 3. 删除数据- DELETE：

**> DELETE FROM OWNERS WHERE ID = 1;**

## 4. 清空表格- TRUNCATE：

**> TRUNCATE TABLE OWNERS;**

# 五、常用函数

## 1. 字符串函数

1. **CONCAT**

* **功能：拼接两个字符串。**
* **语法：CONCAT(string1, string2)**
* **实例：**

**> SELECT CONCAT('Hello', ' World') AS result FROM dual; -- 输出：Hello World**

1. **SUBSTR**

* **功能：从字符串中提取子字符串。**
* **语法：SUBSTR(string, start, length)**
  + **string：目标字符串。**
  + **start：子字符串的起始位置。**
  + **length：子字符串的长度（可选）。**
* **实例：**

**> SELECT SUBSTR('Hello World', 7, 5) AS result FROM dual; -- 输出：World**

1. **UPPER 和 LOWER**

* **功能：将字符串转换为大写/小写。**
* **语法：UPPER(string) / LOWER(string)**
* **实例：**

**> SELECT UPPER('hello') AS upper\_result, LOWER('WORLD') AS lower\_result FROM dual; -- 输出：HELLO, world**

1. **REPLACE**

* **功能：在字符串中替换指定的子字符串。**
* **语法：REPLACE(string, search\_string, replace\_string)**
  + **string：目标字符串。**
  + **search\_string：被替换的子字符串。**
  + **replace\_string：替换的子字符串。**
* **实例：**

**> SELECT REPLACE('Hello World', 'World', 'Universe') AS result FROM dual; -- 输出：Hello Universe**

## 2. 数值函数

1. **ABS**

* **功能：返回数值的绝对值。**
* **语法：ABS(value)**
* **实例：**

**> SELECT ABS(-10) AS result FROM dual;-- 输出：10**

1. **ROUND**

* **功能：按指定精度四舍五入。**
* **语法：ROUND(value, precision)**
  + **value：源数据。**
  + **precision：小数点后的保留位数（可选，默认为0）。**
* **实例：**

**> SELECT ROUND(123.456, 2) AS result FROM dual;-- 输出：123.46**

1. **MOD**

* **功能：返回两个数相除的余数。**
* **语法：MOD(value, divisor)**
  + **value：被除数。**
  + **precision：除数。**
* **实例：**

**> SELECT MOD(10, 3) AS result FROM dual;-- 输出：1**

## 3. 日期函数

1. **SYSDATE**

* **功能：返回当前日期和时间。**
* **语法：SYSDATE**
* **实例：**

**> SELECT SYSDATE AS current\_date FROM dual;-- 输出：当前日期和时间**

1. **ADD\_MONTHS**

* **功能：在日期上增加指定的月数。**
* **语法：ADD\_MONTHS(date, months)**
* **实例：**

**> SELECT ADD\_MONTHS(SYSDATE, 3) AS future\_date FROM dual;-- 输出：当前日期加3个月后的日期**

1. **LAST\_DAY**

* **功能：返回指定日期所在月的最后一天。**
* **语法：LAST\_DAY(date)**
* **实例：**

**> SELECT LAST\_DAY(SYSDATE) AS last\_day\_of\_month FROM dual; -- 输出：当前月的最后一天**

## 4. 转换函数

1. **TO\_CHAR**

* **功能：将日期或数字转换为字符串。**
* **语法：TO\_CHAR(value, format)**
  + **value：日期或数字。**
  + **format：格式模型（可选）。**
* **实例：**

**> SELECT TO\_CHAR(SYSDATE, 'YYYY-MM-DD HH24:MI:SS') AS formatted\_date FROM dual;-- 输出：2025-04-17 14:30:00**

1. **TO\_DATE**

* **功能：将字符串转换为日期。**
* **语法：TO\_DATE(string, format)**
  + **string：日期或数字。**
  + **format：格式模型。**
* **实例：**

**> SELECT TO\_DATE('2025-04-17 14:30:00', 'YYYY-MM-DD HH24:MI:SS') AS date\_result FROM dual; -- 输出：2025-04-17 14:30:00**

1. **TO\_NUMBER**

* **功能：将字符串转换为数字。**
* **语法：TO\_NUMBER(string)**
* **实例：**

**> SELECT TO\_NUMBER('123.45') AS number\_result FROM dual;-- 输出：123.45**

## 5. 空值处理函数

1. **NVL**

* **功能：如果第一个参数为NULL，则返回第二个参数。**
* **语法：NVL(value1, value2)**
* **实例：**

**> SELECT NVL(NULL, 'Default') AS result FROM dual;-- 输出：Default**

1. **NVL2**

* **功能：如果第一个参数不为NULL，则返回第二个参数；否则返回第三个参数。**
* **语法：NVL2(value1, value2, value3)**
* **实例：**

**> SELECT NVL2(NULL, 'Not Null', 'Null') AS result FROM dual;-- 输出：Null**

## 6. 条件函数

1. **DECODE**

* **功能：类似CASE语句，根据条件返回值。**
* **语法：DECODE(value, search1, result1, search2, result2, ..., default)**
* **实例：**

**> SELECT DECODE(grade, 'A', 'Excellent', 'B', 'Good', 'C', 'Fair', 'Poor') AS result FROM (SELECT 'B' AS grade FROM dual);-- 输出：Good**

## 7. 分析函数

1. **RANK**

* **功能：对结果集进行排名，值相同则序号跳跃。**
* **语法：RANK() OVER (ORDER BY column)**
* **实例：**

**> SELECT name, score, RANK() OVER (ORDER BY score DESC) AS rank FROM students;-- 输出：按分数降序排名，分数相同的序号跳跃**

1. **DENSE\_RANK**

* **功能：对结果集进行排名，值相同序号连续。**
* **语法：DENSE\_RANK() OVER (ORDER BY column)**
* **实例：**

**> SELECT name, score, DENSE\_RANK() OVER (ORDER BY score DESC) AS rank FROM students;-- 输出：按分数降序排名，分数相同的序号连续**

1. **ROW\_NUMBER**

* **功能：为结果集生成连续序号。**
* **语法：ROW\_NUMBER() OVER (ORDER BY column)**
* **实例：**

**> SELECT name, score, ROW\_NUMBER() OVER (ORDER BY score DESC) AS row\_num FROM students; -- 输出：按分数降序生成连续序号**

## 7. 其他函数

**LISTAGG**

**PIVOT**

**LIKE**

**INSTR**

**SUBSTR**

**MOD**

**TRIM([LEADING|TRAILING|BOTH] ['trim\_character' FROM] input\_string)**

# 六、关键字

## 1. 伪列

1. **ROWNUM**

* **功能：数据被检索后分配的编号。**

1. **ROWID**

* **功能：反映记录的物理地址。**

## 2. 集合运算

1. **UNION**

* **功能：返回两个查询的结果集，并去除重复记录。**
* **语法：**
  + **SELECT column FROM table1**
  + **UNION**
  + **SELECT column FROM table2;**

1. **UNION ALL**

* **功能：返回两个查询的结果集，包括重复记录。**
* **语法：**
  + **SELECT column FROM table1**
  + **UNION ALL**
  + **SELECT column FROM table2;**

1. **INTERSECT**

* **功能：返回两个查询结果集的交集。**
* **语法：**
  + **SELECT column FROM table1**
  + **INTERSECT**
  + **SELECT column FROM table2;**

1. **MINUS**

* **功能：返回第一个查询结果集减去第二个查询结果集的差集。**
* **语法：**
  + **SELECT column FROM table1**
  + **MINUS**
  + **SELECT column FROM table2;**