数据类型

一、数据类型

1.1 常用数据库类型

数据类型	参 数	描述
char(n)	n=1 to 2000 字节	定长字符串，n字节长，如果不指定长度，缺省为1个字节长（一个汉字为2字节）
varchar2(n)	n=1 to 4000 字节	可变长的字符串，具体定义时指明最大长度n， 这种数据类型可以放数字、字母以及ASCII码字符集(或者EBCDIC等数据库系统接受的字符集标准)中的所有符号。 如果数据长度没有达到最大值n，Oracle 8i会根据数据大小自动调节字段长度， 如果你的数据前后有空格，Oracle 8i会自动将其删去。VARCHAR2是最常用的数据类型。 可做索引的最大长度3209。
number(m,n)	m=1 to 38 n=-84 to 127	可变长的数值列，允许0、正值及负值，m是所有有效数字的位数，n是小数点以后的位数。 如：number(5,2)，则这个字段的最大值是99,999，如果数值超出了位数限制就会被截取多余的位数。 如：number(5,2)，但在一行数据中的这个字段输入575.316，则真正保存到字段中的数值是575.32。 如：number(3,0)，输入575.316，真正保存的数据是575。
date	无	从公元前4712年1月1日到公元4712年12月31日的所有合法日期，Oracle 8i其实在内部是按7个字节来保存日期数据，在定义中还包括小时、分、秒。 缺省格式为DD-MON-YY，如07-11月-00 表示2000年11月7日。
long	无	可变长字符列，最大长度限制是2GB，用于不需要作字符串搜索的长串数据，如果要进行字符搜索就要用varchar2类型。 long是一种较老的数据类型，将来会逐渐被BLOB、CLOB、NCLOB等大的对象数据类型所取代。
raw(n)	n=1 to 2000	可变长二进制数据，在具体定义字段的时候必须指明最大长度n，Oracle 8i用这种格式来保存较小的图形文件或带格式的文本文件，如Miceosoft Word文档。 raw是一种较老的数据类型，将来会逐渐被BLOB、CLOB、NCLOB等大的对象数据类型所取代。
long raw	无	可变长二进制数据，最大长度是2GB。Oracle 8i用这种格式来保存较大的图形文件或带格式的文本文件，如Miceosoft Word文档，以及音频、视频等非文本文件。 在同一张表中不能同时有long类型和long raw类型，long raw也是一种较老的数据类型，将来会逐渐被BLOB、CLOB、NCLOB等大的对象数据类型所取代。
blob clob nclob	无	三种大型对象(LOB)，用来保存较大的图形文件或带格式的文本文件，如Miceosoft Word文档，以及音频、视频等非文本文件，最大长度是4GB。 LOB有几种类型，取决于你使用的字节的类型，Oracle 8i实实在在地将这些数据存储在数据库内部保存。 可以执行读取、存储、写入等特殊操作。
bfile	无	在数据库外部保存的大型二进制对象文件，最大长度是4GB。 这种外部的LOB类型，通过数据库记录变化情况，但是数据的具体保存是在数据库外部进行的。 Oracle 8i可以读取、查询BFILE，但是不能写入。 大小由操作系统决定。

1.2 MySQL与Oracle

数据库常见数据类型对应关系

编号	ORACLE	MYSQL	注释
1	NUMBER	int / DECIMAL	DECIMAL就是NUMBER(10,2)这样的结构INT就是是NUMBER(10)，表示整型； MYSQL有很多类int型，tinyint mediumint bigint等，不同的int宽度不一样
2	Varchar2（n）	varchar(n)	
3	Date	DATATIME	日期字段的处理 MYSQL日期字段分DATE和TIME两种，ORACLE日期字段只有DATE，包含年月日时分秒信息，用当前数据库的系统时间为 SYSDATE, 精确到秒，或者用字符串转换成日期型函数TO_DATE('2001-08-01','YYYY-MM-DD')年-月-日 24小时:分钟:秒的格式YYYY-MM-DD HH24:MI:SS TO_DATE()还有很多种日期格式, 可以参看ORACLE DOC.日期型字段转换成字符串函数TO_CHAR('2001-08-01','YYYY-MM-DD HH24:MI:SS') 日期字段的数学运算公式有很大的不同。MYSQL找到离当前时间7天用 DATE_FIELD_NAME > SUBDATE（NOW（），INTERVAL 7 DAY) ORACLE找到离当前时间7天用 DATE_FIELD_NAME > SYSDATE - 7; MYSQL中插入当前时间的几个函数是：NOW()函数以 `'YYYY-MM-DD HH:MM:SS'返回当前的日期时间，可以直接存到DATETIME字段中。CURDATE()以'YYYY-MM-DD'的格式返回今天的日期，可以直接存到DATE字段中。CURTIME()以'HH:MM:SS'的格式返回当前的时间，可以直接存到TIME字段中。例：insert into tablename (fieldname) values (now()) 而oracle中当前时间是sysdate
4	INTEGER	int / INTEGER	Mysql中INTEGER等价于int
5	EXCEPTION	SQLEXCEPTION	详见<<2009001-eService-O2MG.doc>>中2.5 Mysql异常处理
6	CONSTANT VARCHAR2(1)	mysql中没有CONSTANT关键字	从ORACLE迁移到MYSQL,所有CONSTANT常量只能定义成变量
7	TYPE g_grp_cur IS REF CURSOR;	光标 : mysql中有替代方案	详见<<2009001-eService-O2MG.doc>>中2.2 光标处理
8	TYPE unpacklist_type IS TABLE OF VARCHAR2(2000) INDEX BY BINARY_INTEGER;	数组: mysql中借助临时表处理或者直接写逻辑到相应的代码中， 直接对集合中每个值进行相应的处理	详见<<2009001-eService-O2MG.doc>>中2.4 数组处理

编号	ORACLE	MYSQL	注释
9	自动增长的序列	自动增长的数据类型	MYSQL有自动增长的数据类型，插入记录时不用操作此字段，会自动获得数据值。ORACLE没有自动增长的数据类型，需要建立一个自动增长的序列号，插入记录时要把序列号的下一个值赋于此字段。
10	NULL	NULL	空字符的处理 MYSQL的非空字段也有空的内容，ORACLE里定义了非空字段就不容许有空的内容。按MYSQL的NOT NULL来定义ORACLE表结构, 导数据的时候会产生错误。因此导数据时要对空字符进行判断，如果为NULL或空字符，需要把它改成一个空格的字符串。

二、数据库基础操作

2.1 数据表

2.1.1 建表

CREATE TABLE 语句用于创建数据库中的表。

表由行和列组成，每个表都必须有个表名。

```
-- 创建表
/*
CREATE TABLE table_name
(
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
....
);

数据类型:
数值：int,integer,number
int,integer默认为:number(38)
number长度(1~38,-84~127)
*/

-- 查看表结构
-- DESC 表名; 命令窗口执行
```

2.1.2 事务

事务是什么

事务在数据库中是工作的逻辑单元，单个事务是由一个或多个完成一组的相关行为的SQL语句组成，通过事务机制，可以确保这一组SQL语句所作的操作要么都成功执行，完成整个工作单元操作，要么一个也不执行。

Oracle中默认事务提交方式为手动提交

```
-- 事务操作命令
COMMIT;
ROLLBACK;
```

2.1.3 number

```
-- number:xxxxx.xxxxx
CREATE TABLE a(
    num1 NUMBER, -- 没有精度度number(1~38,-84~127)
    num2 NUMBER(3),-- 整数位长度3:999.5>val>-999.5
    num3 NUMBER(3,0),-- 整数位长度3:999.5>val>-999.5
    num4 NUMBER(3,-1),-- 精确到个位,小数点前1位,对当前位做四舍五入,9995>val>-9995
    num5 NUMBER(3,1), -- 总长度3位,保留1位小数,对小数点后第2位进行四舍五入
99.95>val>-99.95
    num6 NUMBER(3,-3),
    num7 NUMBER(3,3), -- 0.9995>val>-0.9995
    num8 NUMBER(3,-4),
    num9 NUMBER(3,4) -- 0.09995>val>-0.09995
)
```

2.1.4 主键

```
CREATE TABLE dept(
    did INT NOT NULL PRIMARY KEY,-- 没有主键自增
    dname VARCHAR2(10)
)
```

2.1.5 check约束

```
CREATE TABLE emp(
    eid INT NOT NULL PRIMARY KEY,
    ename VARCHAR2(10),
    sex CHAR(1) CHECK(sex=1 OR sex=0), -- 性别：0男,1女
    did INT
)
```

2.1.6 CONSTRAINT

设置表约束

```
/*
约束：CONSTRAINT 约束明 约束类型
*/
CREATE TABLE emp2(
    eid INT NOT NULL PRIMARY KEY,
    ename VARCHAR2(10),
    sex CHAR(1), -- 性别：0男,1女
    did INT,
    CONSTRAINT ck_sex CHECK(sex=1 OR sex=0),
    CONSTRAINT fk_did FOREIGN KEY(did) REFERENCES dept(did)
)
```

2.1.7 删除表

```
-- 删除表
DROP TABLE emp2;
```

2.1.8 修改表

```
-- 修改表
-- 添加列
ALTER TABLE emp ADD phone NUMBER(11);

-- 修改列
ALTER TABLE emp MODIFY phone NUMBER(12);

-- 删除列
ALTER TABLE emp DROP COLUMN phone;

-- 添加约束
ALTER TABLE emp ADD CONSTRAINT fk_emp_dept FOREIGN KEY(did) REFERENCES
dept(did);

-- 删除约束
ALTER TABLE emp DROP CONSTRAINT fk_emp_dept;

-- 添加语句
INSERT INTO dept VALUES(1,'开发部');

-- 批量添加:查询列数与添加的列数一致,类型一致
INSERT INTO dept(did,dname) SELECT classid,classname FROM classes;
```

2.1.9 删除表

```
-- 删除:DELETE需要提交事务,可以添加where语句,TRUNCATE直接清空全表,直接生效
DELETE dept WHERE did=1001;

TRUNCATE TABLE dept;
```

2.2 序列

序列(SEQUENCE)是序列号生成器，可以为表中的行自动生成序列号，产生一组等间隔的数值(类型为数字)。不占用磁盘空间，占用内存。

其主要用途是生成表的主键值，可以在插入语句中引用，也可以通过查询检查当前值，或使序列增至下一个值。

2.2.1 创建序列

```
-- 序列
/*
CREATE SEQUENCE 序列名
[START WITH 数字 -- 序列起始值
INCREMENT BY 数字 -- 自增/自减
MAXVALUE 数字/ NOMAXVALUE  -- 最大值
```

```
MINVALUE 数字/ NOMINVALUE  -- 最小值
CACHE 数字 -- 缓存
CYCLE / NOCYCLE -- 达到最大值/最小值后是否重新开始循环
ORDER] -- 取值顺序
*/

CREATE SEQUENCE seq_dept
START WITH 1 -- 不能比最小值小,小于START WITH+INCREMENT BY
INCREMENT BY 1 -- 正数：自增;负数：自减
MAXVALUE 100
MINVALUE 1
CACHE 2
```

2.2.2 删除序列

```
-- 删除序列
DROP SEQUENCE seq_dept;
```

2.2.3 使用

```
-- 使用
SELECT seq_dept.currval FROM dual; -- 当前值：序列创建完成之后至少使用过1次
SELECT seq_dept.nextval FROM dual; -- 下一个值

ALTER TABLE dept ADD uptime DATE;
ALTER TABLE dept ADD times Timestamp;
```

2.3 日期类型

```
-- insert
INSERT INTO dept VALUES(seq_dept.nextval,'xxx',SYSDATE,SYSDATE);
INSERT INTO dept VALUES(seq_dept.nextval,'xxx','19-1月 2018',SYSDATE);
INSERT INTO dept VALUES(seq_dept.nextval,'xxx','19-1月2018',SYSDATE);
INSERT INTO dept VALUES(seq_dept.nextval,'xxx','19-1月 -18',SYSDATE);
INSERT INTO dept VALUES(seq_dept.nextval,'xxx',DATE'2018-02-20',DATE'2018-02-
19');
INSERT INTO dept VALUES(seq_dept.nextval,'xxx',to_date('2017-01-01','yyyy-mm-
dd'),SYSDATE);
```

2.4 DUAL

```
SELECT * FROM dept;
DELETE dept;
-- dual;虚表：增删改，一行一列
SELECT SYSDATE FROM dual;
SELECT to_date('2017-01-01','yyyy-mm-dd') FROM dual;
SELECT to_date('2017/01/01','yyyy/mm/dd') FROM dual;
```

三、系统函数

函数	描述	SQL
AVG(column)	返回某列的平均值	SELECT AVG(score) FROM score;
COUNT(column)	返回某列的行数（不包括 NULL 值）	SELECT COUNT(stuno) FROM score;
COUNT(*)	返回被选行数	SELECT COUNT(*) FROM score;
MAX(column)	返回某列的最高值	SELECT MAX(score) FROM score;
MIN(column)	返回某列的最低值	SELECT MIN(score) FROM score;
SUM(column)	返回某列的总和	SELECT SUM(score) FROM score;
UPPER(c)	将某个域转换为大写	SELECT UPPER('abc') FROM dual;
LOWER(c)	将某个域转换为小写	SELECT LOWER('ABC') FROM dual;

2.1 RTRIM

```
-- 系统函数
-- LTRIM(str1):去除str1左边空格
-- LTRIM(str1,str2):去除str1左边str2包含的每一个字符
/*
LTRIM(str1,str2):从str1左侧开始判断字符是否在str2中存在,
如果存在删除,删除到第一位不在str2中存在的字符停止
*/
SELECT LTRIM('    X X X    ') FROM dual;
SELECT LTRIM('*****XXX*****','*') FROM dual;
SELECT LTRIM('*!*!*!*!*!XXX*!*!*!*!*!','*!') FROM dual;
SELECT LTRIM('****!!!!*!**!!!XXX***!!!*!!***','*!') FROM dual;

-- 去除右边空格
SELECT RTRIM('    X X X    ') FROM dual;
SELECT RTRIM('*****XXX*****','*') FROM dual;
SELECT RTRIM('*!*!*!*!*!XXX*!*!*!*!*!','*!') FROM dual;
SELECT RTRIM('****!!!!*!**!!!XXX***!!!*!!***','*!') FROM dual;

SELECT RTRIM(LTRIM('**!!XXX*!*!','*!'),'*!') FROM dual;

-- 去除两端空格
SELECT TRIM('    X X X    ') FROM dual;
-- TRIM(str1 FROM str2):在str2中去除两端的str1字符,str1只能包含一个字符
SELECT TRIM('*' FROM '*****XXX*****') FROM dual;
-- TRIM(LEADING/TRAILING str1 FROM str2)
SELECT TRIM(LEADING '*' FROM '*****XXX*****') FROM dual;
SELECT TRIM(TRAILING '*' FROM '*****XXX*****') FROM dual;
```

3.2 字符串

```
-- CONCAT('A','B') 拼接
SELECT CONCAT('A','B') FROM dual;
SELECT 'A'||'B' FROM dual;

-- SUBSTR(str,index,length):截取str字符串,从index开始,截取length长度
-- index从1开始,包含index位
SELECT SUBSTR('abcdefghigk',2),SUBSTR('abcdefghigk',3) FROM dual;
SELECT SUBSTR('abcdefghigk',2,5),SUBSTR('abcdefghigk',3,5) FROM dual;
```

3.3 转换

```
-- 转换函数:TO_CHAR()转换成字符串
SELECT 1, TO_CHAR(1), SYSDATE, TO_CHAR(SYSDATE) FROM DUAL;

-- 转日期
SELECT TO_DATE('2017-01-01', 'yyyy-mm-dd') FROM DUAL;

-- hh:12小时制 hh24:24小时  mi:分钟
SELECT TO_DATE('2017-01-01 13:13:13', 'yyyy-mm-dd hh24:mi:ss') FROM DUAL;
```

3.4 日期

```
-- 日期
-- 系统时间
SELECT SYSDATE FROM dual;
-- 把日期转换成字符串,转换部分
SELECT TO_CHAR(SYSDATE, 'yyyy') FROM DUAL;-- 年
SELECT TO_CHAR(SYSDATE, 'mm') FROM DUAL;-- 月
SELECT TO_CHAR(SYSDATE, 'mi') FROM DUAL;-- 分

-- 周日-周六(1-7)
-- NEXT_DAY(date,周几):从date日期开始的下一个周几
SELECT DATE '2018-01-04',
       NEXT_DAY(DATE '2018-01-04', 1), -- 周日
       NEXT_DAY(DATE '2018-01-04', 2), -- 周一
       NEXT_DAY(DATE '2018-01-04', 3)  -- 周二
  FROM DUAL;

-- months_between(date1,date2):相差的月份date1-date2
SELECT MONTHS_BETWEEN(DATE '2017-01-04', DATE '2018-01-04'),
       MONTHS_BETWEEN(DATE '2018-01-04', DATE '2017-01-04')
  FROM DUAL;

-- add_months(date,num):添加指定的月份后的时间
SELECT add_months(DATE '2017-01-04',1) FROM dual;

-- last_day(date):日期所在月份的最后一天
SELECT last_day(DATE '2000-02-04') FROM dual;

-- TRUNC(str,index):截取字符串str,精确到index位
SELECT TRUNC(9),TRUNC(9.4),TRUNC(9.45),TRUNC(9.5),TRUNC(9.55) FROM dual;
SELECT TRUNC(9),TRUNC(9.4,1),TRUNC(9.45,1),TRUNC(9.5,1),TRUNC(9.55,1) FROM dual;
SELECT TRUNC(9),TRUNC(9.4,-1),TRUNC(9.45,-1),TRUNC(9.5,-1),TRUNC(9.55,-1) FROM
dual;
```

```
-- 截取日期
SELECT SYSDATE,TRUNC(SYSDATE,'yyyy'),TRUNC(SYSDATE,'mm'),TRUNC(SYSDATE,'mi')
FROM dual;
```

3.5 四舍五入

```
-- FLOOR():向下取整
SELECT FLOOR(9),FLOOR(9.4),FLOOR(9.45),FLOOR(9.5),FLOOR(9.55),FLOOR(-9.5) FROM
dual;

-- CEIL():向上取整
SELECT CEIL(9),CEIL(9.4),CEIL(9.45),CEIL(9.5),CEIL(9.55),CEIL(-9.5) FROM dual;

-- ROUND():四舍五入
SELECT ROUND(9),ROUND(9.4),ROUND(9.45),ROUND(9.5),ROUND(9.55),ROUND(-9.5) FROM
dual;
SELECT
ROUND(9,1),ROUND(9.4,1),ROUND(9.45,1),ROUND(9.5,1),ROUND(9.55,1),ROUND(-9.5,1)
FROM dual;
SELECT
ROUND(9,1),ROUND(9.4,-1),ROUND(9.45,-1),ROUND(9.5,-1),ROUND(9.55,-1),ROUND(-9.5,
-1) FROM dual;

-- 截取日期
SELECT SYSDATE,ROUND(SYSDATE,'dd'),ROUND(SYSDATE,'mi') FROM dual;
```

3.6 wm_concat

```
-- 逗号分隔,合并多行结果集为一行
SELECT wm_concat(stuname) FROM student WHERE classid=1005;
```

3.7 REPLACE

```
-- 替换字符串
SELECT REPLACE('AAAxxAx','A','S') FROM dual;

SELECT REPLACE(wm_concat(stuname),',','-') FROM student WHERE classid=1005;
```

3.8 NVL

```
-- NVL(str1,str2):str1是否为空(null或者空字符串),是返回str2,否返回str1
SELECT NVL('a',1),NVL('b',1),NVL(NULL,1),NVL('',1),NVL(' ',1) FROM dual;

-- NVL2(str1,str2,str3):str1是否为空(null或者空字符串),是返回str3,否返回str2
SELECT NVL2('a',1,2),NVL2('b',1,2),NVL2(NULL,1,2),NVL2('',1,2),NVL2(' ',1,2)
FROM dual;
```