* 数据库原理

**1.1. 数据库简介**

**1.1.1. 文件存储**

**1.1.2. DB和DBMS**

数据库（Database,简称DB）是按照数据结构来组织、存储和管理数据的仓库。

数据库管理系统(Database Management System,简称DBMS)：管理数据库的软件。

数据库的典型特征包括：数据的结构化，数据间的共享，减少数据的冗余度，以及数据的独立性。

数据库成为数据的知识仓库，并对这些数据的存储、访问、安全、数据一致性、并发操作及备份恢复负责。

**1.1.3. 关系数据库简介**

主流数据库产品：

* Oracle（Oracle）
* DB2（IBM）
* SQL Server（MS）
* MySQL（Oracle）

**1.1.4. 表的概念**

在关系数据库中，数据被存放于二维数据表(Table)中。

一个关系数据库由多个数据表组成，数据表是关系数据库的基本存储结构，由行和列组成，行（Row）也就是横排数据，也经常被称作记录(Record)，列（Column）就是纵列数据，也被称作字段(Field)。表和表之间是存在关联关系的。

**1.2. 主流关系型数据库**

**1.2.1. Oracle数据库概述**

Oracle是当今著名的Oracle(甲骨文)公司的数据库产品，它是世界上第一个商品化的关系型数据库管理系统，也是第一个推出和数据库结合的第四代语言开发工具的数据库产品。

Oracle采用标准的SQL结构化查询语言，支持多种数据类型，提供面向对象的数据支持，具有第四代语言开发工具，支持UNIX、WINDOWS、OS/2等多种平台。Oracle公司的软件产品丰富，包括Oracle服务器产品，Oracle开发工具和Oracle应用软件。其中最著名的就是Oracle数据库，目前最新的版本是Oracle 12c。

**1.2.5. MySQL数据库概述**

MySQL是一个开放源码的小型关系型数据库管理系统，开发者为瑞典MySQL AB公司。目前MySQL被广泛地应用在Internet上的中小型网站中。

2008年1月16日，Sun正式收购MySQL。2009年4月20日，SUN被Oracle公司收购。目前的最新版本是MySQL5.6.

**1.3. SQL概述**

**1.3.1. 结构化查询语言**

SQL(Structured Query Language) 是结构化查询语言的缩写。

SQL是在关系数据库上执行数据操作、检索及维护所使用的标准语言,可以用来查询数据，操纵数据，定义数据，控制数据，所有数据库都使用相同或者相似的语言。

SQL可分为:

1. 数据定义语言（DDL） : Data Definition Language
2. 数据操纵语言（DML） : Data Manipulation Language
3. 事务控制语言（TCL）：Transaction Control Language)
4. 数据查询语言（DQL）：Data Query Language
5. 数据控制语言（DCL） : Data Control Language

执行SQL语句时，用户只需要知道其逻辑含义，而不需要知道SQL语句的具体执行步骤。

**2.1. Oracle数据类型**

**2.1.1. NUMBER**

NUMBER表示数字类型，经常被定义成NUMBER（P，S）形式，其中：

•P表示数字的总位数

•S表示小数点后面的位数

例如在表Emp中的Sal列的定义如下：

Sal NUMBER(6,2)

表示Sal列中的数据，整数位最大为4位，小数位最大位数是2位，也就是最大取值：9999.99。

**2.1.2. CHAR**

CHAR表示固定长度的字符类型，经常被定义成CHAR（N）形式, N表示占用的字节数，N的**最大取值是2000**。

例如在表Emp中的Ename列的定义如下：Ename CHAR(20)；

表示Ename列中最多可存储20个字节的字符串，并且占用的空间是***固定***的20个字节。

**2.1.3. VARCHAR2**

VARCHAR2表示变长的字符类型，定义格式是VARCHAR2（N）, N表示最多可占用的字节数，**最大长度是4000字节**。

例如在表Emp中的JOB列的定义如下：JOB VARCHAR2(100)；

表示JOB列中最多可存储长度为100个字节的字符串。根据其中保存的数据长度，占用的空间是***变化***的，最大占用空间为100个字节。

**2.1.4. DATE**

DATE用于定义日期时间的数据，长度是7个字节，默认格式是：DD-MON-RR, 例如：“11-APR-71”。如果是中文环境，是“11-4月-71”这种形式。

例如在表Emp中的Hiredate列的定义如下：Hiredate DATE；

表示Hiredate列中存放的是日期数据。

**2.2. 创建表**

**2.2.1. CREATE语句**

创建表的语法是：

01.--创建表emp

02.CREATE TABLE emp(

03. id NUMBER(10),

04. name VARCHAR2(20),

05. gender CHAR(1),

06. birth DATE,

07. salary NUMBER(6,2),

08. job VARCHAR2(30),

09. deptid NUMBER(2)

10.);

**2.2.2. DESC语句**

DESC是查看表的数据结构，语法是：

DESC table\_name;

**2.2.3. DEFAULT**

可以通过DEFAULT子句给列指定默认值，例如，在emp表中给gender列赋默认值‘M’,如果插入数据时没有指定性别的员工，则默认是男性。

创建表emp1：

01.CREATE TABLE emp1(

04. gender CHAR(1) DEFAULT ‘M’,

09.);

**2.2.4. NOT NULL**

默认情况下，任何列都允许有空值。非空(Not Null)是一种约束条件，用于确保字段值不为空，当某个字段被设置了非空约束条件，这个字段中必须存在有效值，也就是说，当执行插入数据的操作时，必须提供这个列的数据，当执行更新操作时，不能给这个列的值设置为NULL。

创建表emp2，设置name字段为非空，也即是表emp2中的每条记录，name列都必须有值。

01.CREATE TABLE emp2(

03. name VARCHAR2(20) NOT NULL,

09.);

**2.3. 修改表**

**2.3.1. 修改表名**

在建表后如果希望修改表名，可以使用RENAME语句实现，语法如下，将改变表名old\_name为new\_name：

RENAME old\_name TO new\_name;

**2.3.2. 增加列**

在建表之后，要给表增加列可以使用ALTER TABLE的ADD子句实现。语法如下：

ALTER TABLE table\_name ADD

(column datatype [DEFAULT expr] [, column datatype…])

注意一点，列只能增加在最后，不能插入到现有的列中。下例给表employee增加一列hiredate，并设置默认值为当前日期。

01.ALTER TABLE employee ADD (hiredate DATE DEFAULT sysdate);

**2.3.3. 删除列**

在建表之后，使用ALTER TABLE的DROP子句删除不需要的列。语法如下：

ALTER TABLE table\_name DROP (column);

下例删除表employee的列hiredate：

ALTER TABLE employee DROP (hiredate);

**2.3.4. 修改列**

建表之后，可以改变表中列的数据类型、长度和默认值，注意这种修改仅对以后插入的数据有效，另外如果表中已经有数据的情况下，把长度由大改小，有可能不成功，比如原来的类型是VARCHAR2(100),其中已经存放了100个字节长度的数据，如果要改为VARCHAR2（80），则不会修改成功。

下例修改表employee的列job，并增加默认值的设置:

01.ALTER TABLE employee

02.MODIFY(job VARCHAR2(40) DEFAULT 'CLERK' );

**2.4. DML语句**

在做后续练习之前，先熟悉本部分使用的样例表emp和dept的结构和数据：

**2.4.1. INSERT语句**

INSERT语句用来给数据表增加记录，每次增加一条记录。语法如下：

INSERT INTO table\_name[(column[, column…])]

VALUES(value[, value…]);

需要注意的一点，所有的DML操作，都需要再执行事务提交语句commit，才算真正确认了此操作。

向employee表插入一条记录的例子如下：

01.INSERT INTO employee(id, name, job, salary)

02.VALUES(1001, 'rose', 'PROGRAMMER', 5500);

Oracle中的日期数据比较特殊，如果插入的列有日期字段，需要考虑日期的格式。Oracle的默认的日期格式‘DD-MON-RR，你或者按照默认格式插入数据，或者自定义日期格式，用TO\_DATE函数转换为日期类型的数据。

01.--使用默认日期格式插入记录

02.INSERT INTO employee(id, name, job,birth)

03.VALUES(1002, 'martha', 'ANALYST', '01-9月-89');

05.--使用自定义日期格式插入记录

06.INSERT INTO employee(id, name, job, birth)

07.VALUES(1003, 'donna', 'MANAGER',

08.TO\_DATE('1978-09-01', 'YYYY-MM-DD'));

**2.4.2. UPDATE语句**

UPDATE语句用来更新表中的记录，语法如下：

其中WHERE子句是可选的，但是如果没有WHERE子句，则全表的数据都会被更新，务必小心。

下例更改职员ROSE的薪水为8500:

01.UPDATE employee SET salary = 8500 WHERE name = 'ROSE';

**2.4.3. DELETE语句**

和UPDATE语句一样，WHERE子句是可选的，但是如果没有WHERE子句，则全表的数据都会被删除，务必确认后再予以删除。

下例将删除职位是空的员工记录:

DELETE FROM employee WHERE job is null;

* Oracle字符串操作

**1.1. 字符串类型**

**1.1.1. CHAR和VARCHAR2类型**

CHAR和VARCHAR2类型都是用来表示字符串数据类型，用来在表中存放字符串信息, 比如姓名、职业、地址等。

CHAR存放定长字符，如果数据存不满定长长度，则补齐空格；

VARCHAR2存放变长字符，实际数据有多少长度则占用多少。

CHAR类型浪费空间换取查询时间的缩短，VARCHAR2节省空间查询时间较CHAR类型要长。字符串按照自然顺序排序。

**1.1.2. CHAR和VARCHAR2的存储编码**

字符串在数据库中存储的默认单位是字节，也可显式指定为字符。如：

•CHAR(10)，等价于 CHAR(10 BYTE)

•如果指定单位为字符：CHAR(10 CHAR)，20个字节

•VARCHAR2(10)， 等价于VARCHAR2（10 BYTE）

•指定单位为字符：VARCHAR2(10 CHAR)，20个字节

每个英文字符占用一个字节，每个中文字符按编码不同，占用2-4个字节：

•ZHS16GBK: 2个字节

•UTF-8: 2-4个字节

**1.1.3. CHAR和VARCHAR2的最大长度**

CHAR类型的最大取值为2000字节，也就是定义为CHAR（2000）。其中最多保存2000个英文字符，1000个汉字（GBK）。

VARCHAR2最大取值为4000字节，也就是VARCHAR2（4000），最多保存4000个英文字符，2000个汉字（GBK）。

CHAR如果不指定长度，默认为1个字节， VARCHAR2必须指定长度。

**1.1.4. LONG和CLOB类型**

LONG类型可以认为是VARCHAR2的加长版，用来存储变长字符串，最多达2GB的字符串数据，但是LONG类型有诸多限制，所以不建议使用：

1. 每个表只能有一个LONG类型列；
2. 不能作为主键；
3. 不能建立索引；
4. 不能出现在查询条件中等

CLOB用来存储定长或变长字符串，最多达4GB的字符串数据，ORACLE建议开发中使用CLOB替代LONG类型，比如如下方式定义数据

01.CREATE TABLE student(

02.id NUMBER(4),

03.name CHAR(20),

04.detail CLOB);

**1.2. 字符串函数**

**1.2.1. CONCAT和“||”**

CONCAT是字符串连接函数，语法是：

01.CONCAT(char1, char2)

用于返回两个字符串连接后的结果，两个参数char1、char2是要连接的两个字符串。concat只能有两个参数，所以如果连接三个字符串时，需要两个concat函数。比如连接emp表中的name列和salary列，中间用“：”隔开：

01.SELECT CONCAT(CONCAT(name, ' : '), sal)FROM emp;

图-1 CONCAT连接字符串的结果

在连接两个以上操作符时并不是很方便。concat的等价操作是连接操作符”||”。当多个字串连接时，用||符号更直观。下述SQL语句实现相同的效果：

01. SELECTename || ' : ' || sal FROMemp;

在连接时，如果任何一个参数是NULL，相当于连接了一个空格。

**1.2.2. LENGTH**

LENGTH(char)用于返回参数字符串的长度。如果字符类型是VARCHAR2，返回字符的实际长度，如果字符类型是CHAR，长度还包括后补的空格。例如：

01.SELECTename, LENGTH(name) FROM emp;

将列出name和name的字符串长度：

图-2Lenth函数的运行结果

**1.2.3. UPPER、LOWER和INITCAP**

这三个函数全部是英文的大小写转换函数，用来转换字符的大小写：

•UPPER(char)用于将字符转换为大写形式

•LOWER(char)用于将字符转换为小写形式

•INITCAP(char)用于将字符串中每个单词的首字符大写，其它字符小写，单词之间用空格和非字母字符分隔

如果这三个函数的输入参数是NULL值，仍然返回NULL值。例如：

01. SELECT UPPER('hello world'), LOWER('HELLO WORLD'), INITCAP('hello world')

02. FROM DUAL;

将列出参数“hello world”的大写、小写和首字符大写的形式。一般用来查询数据表中不确定大小写的情况。

**1.2.4. TRIM、LTRIM、RTRIM**

这三个TRIM函数的作用都是截去子字符串。语法形式及解释：

•TRIM(c2 FROM c1) 表示从c1的前后截去c2

•LTRIM(c1[, c2]) 表示从c1的左边（Left）截去c2

•RTRIM(c1[, c2]) 表示从c1的右边（Right）截去c2

在后两个函数中，如果没有参数c2，就去除空格。例如：

其中最常用的是TRIM，经常用来去掉字符串前后的空格。

**1.2.5. LPAD、RPAD**

PAD意即补丁，LPAD和RPAD两个函数都叫做补位函数，LPAD表示LEFT PAD，在左边打补丁，RPAD表示RIGHT PAD，在右边打补丁。语法如下：

•LPAD(char1, n, char2) 左补位函数

•RPAD(char1, n, char2) 右补位函数

参数的含义：在字符串参数char1的左端或右端用char2补足到n位，其中参数char2可重复多次。例如在EMP表中使用左补位，将sal用$补齐6位，运行结果如图-5所示。

**1.2.6. SUBSTR**

SUBSTR表示在一个字符串中截取子串，语法是：

01.SUBSTR(char, [m[, n]])

用于返回char中从m位开始取n个字符的子串，字符串的首位计数从1开始。参数含义如下：

•如果m = 0，则从首字符开始，如果m取负数，则从尾部开始

•如果没有设置n，或者n的长度超过了char的长度，则取到字符串末尾为止

**1.2.7. INSTR**

用来返回在一个字符串中子串的位置。语法是：

01.INSTR(char1, char2[, n [, m]])

参数的含义：

•返回子串char2在源字符串char1中的位置

•从n的位置开始搜索，没有指定n，从第1个字符开始搜索

•m用于指定子串的第m次出现次数，如果不指定取值1

•如果在char1中没有找到子串char2 ，返回0

* Oracle数值操作

**2.1. 数值类型**

**2.1.1. NUMBER（p）表示整数**

数据表中的数值类型用NUMBER表示，完整语法是：

01.NUMBER(precision ，scale)

**2.1.2. NUMBER（P,S）表示浮点数**

如果NUMBER(precision ，scale)的两个参数全部显式定义，则表示浮点数：

•precision：NUMBER可以存储的最大数字长度（不包括左右两边的0）

•scale：在小数点右边的最大数字长度（包括左侧0）

**2.2. 数值函数**

**2.2.1. ROUND**

数值函数指参数是数值类型的函数。常用的有ROUND、TRUNC、MOD、CEIL和FLOOR。其中ROUND用来四舍五入，语法如下：

ROUND(n[, m])用于将参数n按照m的数字要求四舍五入。其中：

01.SELECT ROUND(45.678, 2) FROM DUAL; --45.68

02.SELECT ROUND(45.678, 0) FROM DUAL;--46

03.SELECT ROUND(45.678, -1) FROM DUAL;--50

**2.2.2. trunc**

TRUNC(n[, m])的功能是截取，其中n和m的定义和ROUND(n[, m])相同，不同的是功能上按照截取的方式处理数字n。例如：

01.SELECT TRUNC(45.678, 2) FROM DUAL; --45.67

02.SELECT TRUNC(45.678, 0) FROM DUAL;--45

03.SELECT TRUNC(45.678, -1) FROM DUAL;--40

**2.2.3. MOD**

MOD(m, n)是取模函数，返回m除以n后的余数，如果n为0则直接返回m。例如：

01.--薪水值按1000取余数

02.SELECTename, sal, MOD(sal, 1000) FROM emp;

**2.2.4. CEIL和FLOOR**

CEIL(n)、FLOOR(n)这两个函数顾名思义，一个是天花板，就是取大于或等于n的最小整数值，一个是地板，就是取小于或等于n的最大整数值。比如数字n = 4.5，那么它的CEIL是5.0，它的FLOOR是4.0。在SQL语句中的例子如下：

01.SELECT CEIL(45.678) FROM DUAL; --46

02.SELECT FLOOR(45.678) FROM DUAL;--45

* Oracle日期操作

**3.1. 日期类型**

**3.1.1. DATE**

DATE和TIMESTAMP是ORACLE中最常用的日期类型。DATE用来保存日期和时间。表示范围从是公元前4712年1月1日至公元9999年12月31日。

DATE类型在数据库中的实际存储固定为7个字节，格式分别为：

• 第1字节：世纪+100

• 第2字节：年

• 第3字节：月

• 第4字节：天

• 第5字节：小时+1

• 第6字节：分+1

• 第7字节：秒+1

**3.1.2. TIMESTAMP**

TIMESTAMP表示时间戳，与DATE的区别是不仅可以保存日期和时间，还能保存小数秒，可指定为0-9位，默认6位，最高精度可以到ns(纳秒)级别。

数据库内部用7或者11个字节存储，精度为0时，用7字节存储，与DATE功能相同，精度大于0则用11字节存储。格式为：

• 第1字节-第7字节：和DATE相同

• 第8-11字节：纳秒，采用4个字节存储，内部运算类型为整型

用日期类型建表的例子：

01. CREATE TABLE test(

02.c1 DATE,

03.c2 TIMESTAMP(9));

**3.2. 日期关键字**

**3.2.1. SYSDATE**

SYSDATE本质是一个Oracle的内部函数，用来返回当前的系统时间，精确到秒，默认显示格式是DD-MON-RR，只有年月日并不显示时间。例如：

SELECT SYSDATE FROM DUAL;

如果数据库是英文环境，将显示“03-MAY-14”的形式，如果是中文环境，将显示“03-5月-14”的形式。如果想显示时分秒，需要将格式转换一下：

01.SELECT TO\_CHAR(SYSDATE,'yyyy-mm-dd day hh24:mi:ss') FROM DUAL;

在建表时，可以将系统时间SYSDATE作为某一列的默认值，当插入新的记录，将会取当时的系统时间，作为数据表的一列数据保存起来。

**3.2.2. SYSTIMESTAMP**

SYSTIMESTAMP也是Oracle的内部日期函数，返回当前系统日期和时间，精确到毫秒。例如：

01.SELECT SYSTIMESTAMP FROM DUAL;

02.SELECT TO\_CHAR(SYSTIMESTAMP,'SSSS.FF') FROM DUAL;

**3.3. 日期转换函数**

**3.3.1. TO\_DATE**

日期数据有时需要和字符串数据相互转换，需要用到日期转换函数，包括TO\_CHAR和TO\_DATE。

TO\_DATE的功能是将字符串按照定制格式转换为日期类型，语法格式是：

01.TO\_DATE(char[, fmt[, nlsparams]])

其中：char是要转换的字符串，fmt是转换格式，nlsparams是指定日期语言。其中比较重要的是格式，常用的日期格式如下：

01. SELECTename, hiredate

02. FROM emp

03. WHERE hiredate>

04. TO\_DATE(‘2002-01-01',

05. 'YYYY-MM-DD');

**3.3.2. TO\_CHAR**

TO\_CHAR的作用是将其它类型（日期，数值）的数据转换为字符类型，主要应用在日期类型上。语法格式：

01.TO\_CHAR(date[, fmt[, nlsparams]])

其中fmt是格式，将日期类型数据date按照fmt格式输出字符串，nlsparams用于指定日期语言。例如：

01. SELECTename,TO\_CHAR(hiredate, 'YYYY"年"MM"月"DD"日"')

02. FROM emp;

**3.4. 日期常用函数**

**3.4.1. LAST\_DAY**

LAST\_DAY(date)：返回日期date所在月的最后一天，一般是在按照自然月计算某些业务逻辑，或者安排月末周期性活动时很有用处。例子：

01.SELECT LAST\_DAY(SYSDATE) FROM DUAL;--查询当月的最后一天

02.SELECT LAST\_DAY('20-2月-09') FROM DUAL;--查询09年2月的最后一天

**3.4.2. ADD\_MONTHS**

ADD\_MONTHS(date, i)：返回日期date加上i个月后的日期值.

•参数i可以是任何数字，大部分时候取正值整数

•如果i是小数，将会被截取整数后再参与运算

•如果i是负数，则获得的是减去i个月后的日期值

01. SELECTename, ADD\_MONTHS(hiredate, 20 \* 12) "20周年"

02. FROM emp;

**3.4.3. MONTHS\_BETWEEN**

MONTHS\_BETWEEN(date1, date2)：计算date1和date2两个日期值之间间隔了多少个月，实际运算是date1-date2，如果date2时间比date1晚，会得到负值。

除非两个日期间隔是整数月，否则会得到带小数位的结果，比如计算2009年9月1日到2009年10月10日之间间隔多少个月，会得到1.29个月。例如计算职员入职多少个月：

01. SELECTename, MONTHS\_BETWEEN(SYSDATE, hiredate) hiredate FROM emp；

**3.4.4. NEXT\_DAY**

NEXT\_DAY(date, char)：返回date日期数据的下一个周几，周几是由参数char来决定的。在中文环境下，直接使用”星期三”这种形式，英文环境下，需要使用”WEDNESDAY”这种英文的周几。为避免麻烦，可以直接用数字1-7表示周日-周六。

需要注意的是NEXT\_DAY不要按字面意思理解为明天。查询下个周三是几号：

01.SELECT NEXT\_DAY(SYSDATE, 4) next\_wedn FROM DUAL;

**3.4.5. LEAST、GREATEST**

比较函数LEAST和GREATEST语法如下：

01.GREATEST(expr1[, expr2[, expr3]]…)

02.LEAST(expr1[, expr2[, expr3]]…)

两个函数都可以有多个参数值，但参数类型必须一致，返回结果是参数列表中最大或最小的值。

在比较之前，在参数列表中第二个以后的参数会被隐含的转换为第一个参数的数据类型，所以如果可以转换，则继续比较，如果不能转换将会报错。

01.SELECT LEAST(SYSDATE, '10-10月 -08') FROM DUAL;

**3.4.6. EXTRACT**

EXTRACT直译是抽取或提取：

EXTRACT(date FROM datetime)：从参数datetime中提取参数date指定的数据，比如提取年、月、日。例如取出当前日期的年：

01.SELECT EXTRACT(YEAR FROM SYSDATE) current\_year FROM DUAL;

取出指定时间的小时：

01.SELECT EXTRACT(HOUR FROM TIMESTAMP '2008-10-10 10:10:10')

02.FROM DUAL;

* 4. 空值操作

**4.1. NULL的含义**

NULL是数据库里的重要概念，即空值。当表中的某些字段值，数据未知或暂时不存在，取值NULL。

Java中的简单数据类型是不能取值NULL的，在数据库中，任何数据类型均可取值NULL。

**4.2. NULL操作**

**4.2.1. 插入NULL值**

在数据表中插入记录时，如果要插入NULL值，可以用显式指定NULL值的方式，或者不插入某个字段值，即隐式表示NULL值。例如表student中：

01.CREATE TABLE student(id NUMBER(4), name CHAR(20), gender CHAR(1));

03.INSERT INTO student VALUES(1000, '李莫愁', 'F');

05.INSERT INTO student VALUES(1001, '林平之', NULL);--显式插入NULL值

07.INSERT INTO student(id, name) VALUES(1002, '张无忌');--隐式插入NULL值

**4.2.2. 更新成NULL值**

把数据表的某个字段更新为NULL值，和更新为其他数据的语法是相同的。比如更新

01.UPDATE student SET gender = NULL; --全表都被更新

注意这种更新只有在此列没有非空约束的情况下才可操作。如果gender列有非空约束，则无法更新为NULL值，上述语句会报错。

**4.2.3. NULL条件查询**

在条件查询中，因为NULL不等于任何值，所以不能用“列名=NULL”这种形式查询。必须用“列名 IS NULL”来判断，或者用“列名 IS NOT NULL”来查询非空数据。

01.SELECT \* FROM student WHERE gender IS NULL;

**4.2.4. 非空约束**

非空(NOT NULL)约束是约束条件的一种，用于确保数据表中某个字段值不为空。

因为在默认情况下，任何数据类型的列都允许有空值，但系统的业务逻辑可能会要求某些列不能取空值。这时需要在建表时指定该列不允许为空。

一旦某个字段被设置了非空约束条件，这个字段中必须存在有效值。即：当执行插入数据的操作时，必须提供这个列的数据，当执行更新操作时，不能给这个列的值设置为NULL。

01.--创建数据表student，其中gender列不允许为空

02.CREATE TABLE student

03. (id NUMBER(4),

04.name CHAR(20),

05.gender CHAR(1) NOT NULL);

**4.3. 空值函数**

**4.3.1. NVL**

NVL(expr1, expr2)：将NULL转变为非NULL值。如果expr1为NULL，则取值expr2, expr2是非空值。

其中expr1和expr2可以是任何数据类型，但两个参数的数据类型必须是一致的。

计算员工月收入，如果comm列为空值的话，最终计算结果将是空，不符合逻辑，所以先将取NULL值的comm列转换为0，再相加。

01.SELECTename, sal, comm,sal + nvl(comm, 0) salary

02.FROM emp;

**4.3.2. NVL2**

NVL2(expr1, expr2, expr3)：和NVL函数功能类似，都是将NULL转变为非空值。NVL2用来判断expr1是否为NULL，如果不是NULL，返回expr2，如果是NULL，返回expr3。

01. SELECTename, sal, comm,

02.nvl2(comm, sal + comm, sal) salary

03. FROM emp;

* 课后练习整理

1:显示格式：ename:job

SELECT ename||':'||job

FROM emp\_wwww

2:查看每个员工职位的字符个数

SELECT ename,LENGTH(ename)

FROM emp\_wwww

3:小写形式，与首字母大写形式

SELECT INITCAP(ename)||' '||LOWER(ename)

FROM emp\_wwww

4:去除

SELECT TRIM('a'FROM'aaaaabaaaaa')

FROM DUAL

SELECT LTRIM('aaaaabaaaaaa','a')

FROM DUAL

SELECT RTRIM('aaaaabaaaaaa','a')

FROM DUAL

5:补位函数，要求显示10位

SELECT LPAD(ename,10,' ')||RPAD(ename,10,' ')

FROM emp\_wwww

6:查看'DOCTOR WHO'中'WHO'的位置

SELECT INSTR('DOCTOR WHO','WHO',1,1)

FROM DUAL

7:截取字符串'DOCTOR WHO'中的'WHO'

SELECT SUBSTR('DOCTOR WHO',8,3)

FROM DUAL

8:查看55.789四舍五入

SELECT ROUND(55.789,2)||ROUND(55.789,0)||ROUND(55.789,-1)

FROM DUAL

9:查看55.789截取后保留小数点后2位

SELECT TRUNC(55.789,2)||TRUNC(55.789,0)||TRUNC(55.789,-1)

FROM DUAL

10:查看每名员工工资百位以下的数字?取余

SELECT ename,sal,MOD(sal,100)

FROM emp\_wwww

11:查看每名员工从入职到今天一共多少天

SELECT ename,ROUND((SYSDATE-hiredate),2)

FROM emp\_wwww

12:查看从2008-08-08号到今天为止一共经历了多少天?

SELECT ROUND((SYSDATE-TO\_DATE('2008-08-08','YYYY-MM-DD')),2)

FROM DUAL

13:将每名员工入职时间转换为指定格式字符串

SELECT ename,TO\_CHAR(hiredate,'YYYY"年"MM"月"DD"日"')

FROM emp\_wwww

14:查看每个员工入职所在月的月底是哪天?

SELECT ename,LAST\_DAY(hiredate)

FROM emp\_wwww

15:查看每名员工转正日期(入职后3个月)

SELECT ename,ADD\_MONTHS(hiredate,3)

FROM emp\_wwww

16:查看每名员工入职至今共多少个月?

SELECT ename,MONTHS\_BETWEEN(SYSDATE,hiredate)

FROM emp\_wwww

17:查看从明天开始一周内的周日是哪天?

SELECT NEXT\_DAY(SYSDATE,1)

FROM DUAL

18:查看82年以后入职的员工的入职日期，82年以前的按照1982年01月01号显示。格式都是DD-MON-RR(默认格式)

SELECT ename,hiredate

FROM emp\_wwww

WHERE EXTRACT(YEAR FROM hiredate)>1982

SELECT ename,TO\_CHAR(hiredate,'YYYY"年"MM"月"DD"日"')

FROM emp\_wwww

WHERE EXTRACT(YEAR FROM hiredate)<1982

19:查看每名员工的入职年份?

SELECT ename, EXTRACT(YEAR FROM hiredate)

FROM emp\_wwww

20:显示每个员工的总收入(工资加奖金)，奖金为NULL的只看工资

SELECT ename,sal,comm,sal+NVL(comm,0)

FROM emp\_wwww

21:使用NVL2实现20题的需求

SELECT ename,sal,comm,NVL2(comm,(sal+comm),sal)

FROM emp\_wwww

22:查看到今天为止自己活了多少天?

SELECT ROUND((SYSDATE-TO\_DATE('1991-04-22','YYYY-MM-DD')),0)

FROM DUAL