دانشگاه آزاد اسلامی واحد پرند

عنوان :

جزوه درس آزمایشگاه پایگاه داده ها

تهیه وتنظیم :

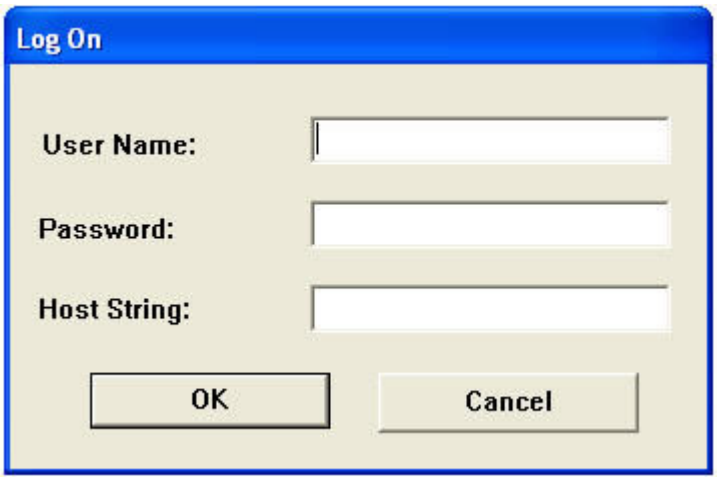
مهندس محمد فرجي مهماندار

فهرست

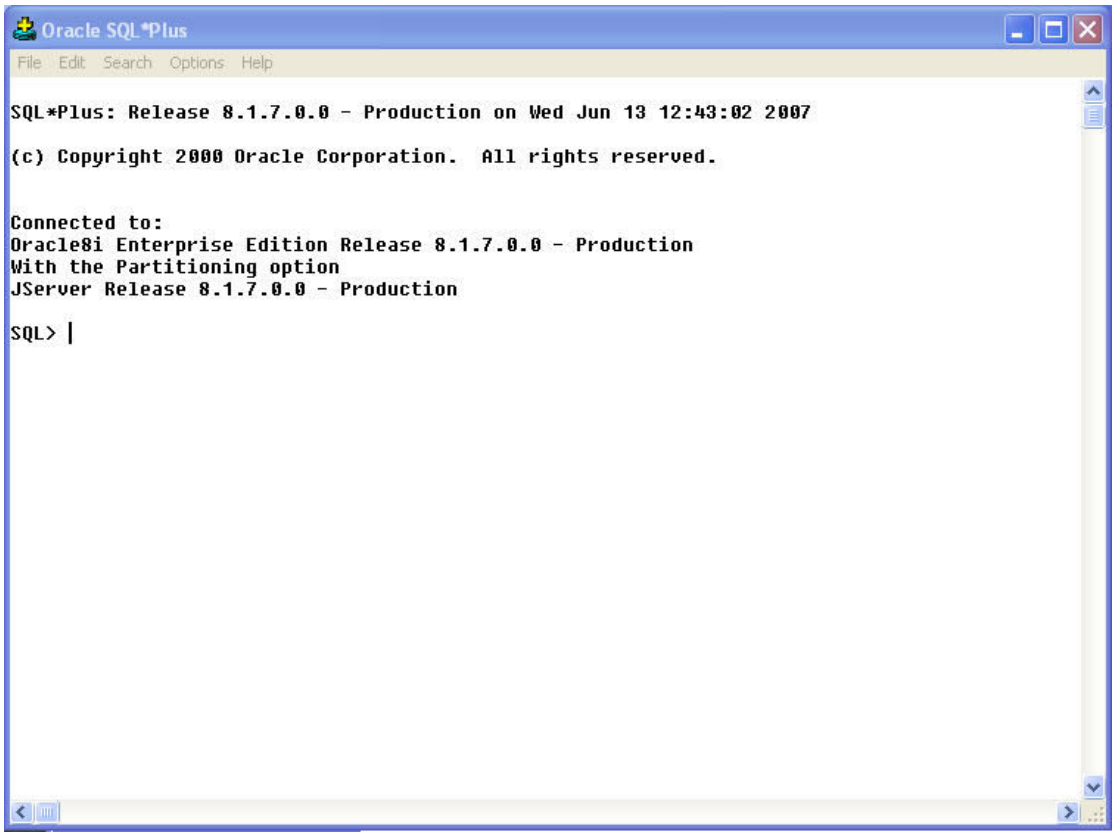
فصل اول : مقدمه ای بر زبان SQL

نحوه ورود به محیط SQl*plus :

SQl*plus یکی ازمحصولات شرکت اوراکل است این برنامه محیطی رافراهم می سازد کـه درآن مـی تـوان دسـتورات SQL رابطورمستقیم ویابرروی فایل ، نوشته وسپس اجرانمود. بـرای ورود بـه آن درمحـیط وینـدوز ، بایدابتـدا بـرروی نـشانه(Icon) SQL*Plus کلیک نمایید. دراینحالت نام کاربر ، کلمه رمز ونام بانک اطلاعاتی درخواست می گـردد کـه بایـدآنهارا بـه دقـت مشخص کنید.



پس ازدرج صحیح اطلاعات فوق اعلان (Prompt) < SQL ظاهرمی گردد درمقابل این اعلان ، میتوان تمامی فـرامین SQL اجرانمود.



نحوه اصلاح عبارتهای SQL بااستفاده ازفرامین SQL*Plus:

۱)اگرپیش ازاتمام یک فرمان کلید Return رافشاردهیدخط بعدی بایک شماره خط بعنوان اعلان ، ظاهرخواهدشد.

۲)علامت (Srmicolon) نشانه پایان یک دستوراست.

۳)یک فرمان SQL بهنگام درج دربافر SQL قرارمی گیرد وتا درج فرمان بعدی درآن باقی می ماند.

۴)بااستفاده ازدستورات ویرایشی درمحیط SQL*Plus می توان تغییرات لازم رابرروی عبارت موجوددربافر SQL اعمال نمود

این دستورات ویرایشی عبارتنداز :

نام فرمان	علامت اختصار	هدف فرمان
APPEND text	A text	متنی را به انتهای خط جاری ، اضافه میکند
CHANGE	c/old/new	متن موجود را بامتن جدید جایگزین می کند
CHANGE	c/text	متن موجود را ازخط جاری ، حذف می کند
GLEAR BUFFER	CL BUFF	همه خطوط موجوددربافر SQL راحذف می کند
DEL		خط جاری راحذف میکند
INPUT	I	تعدادنامشخصی ازخطوط رادرج می کند
INPUT	I text	خطی حاوی متن موردنظررادرج می کند
LIST	L	لیست همه خطوط موجود دربافر SQL رانشان می دهد
LIST	Ln	خط n ام ازبافر SQL را نشان می دهد
LIST	L M,n	محدوده ای ازخطوط موجود دربافر SQL (m تا n)رانشان میدهد
RUN	/ یا R	فرمان موجودرا بافر SQL رانشان داده واجرا میکند
n	N	خط شماره n ازدستورموجود دربافر SQL رافعال می سازد.

فرامین SQL*Plus :

این فرامین باید درمقابل اعلان SQL درج شوندوعبارتنداز :

فرمان	شرح
SAVE filename	محتویات بافر SQL را دریک فایل ذخیره میکند
GET filename	محتویات یک فایل رادربافر SQL قرارمی دهد
START filename	دستورات موجود دریک فایل رااجرا می کند
@filename	دستورات موجود دریک فایل رااجرا می کند
ED filename	ازویرایشگر پیش فرض (مثل notepad) برای ویرایش متن یک فایل استفاده میکند
SPOOL filename	باعث نوشته شدن تمامی دستورات ونتایج حاصله ، دریک فایل میگردد. پسونداین فایل ، بطورپیش فرض ، ist می باشد (spool بمعنی ثبت مراحل اجرای دستورات دریک فایل است
SPO[OL]OFF	باعث غیرفعال شدن حالت Spool میگردد
DESC[RIBE] tablename	ساختارجدول بانک اطلاعاتی رانشان می دهد
HELP	راهنمای سیستم رانشان میدهد
HOST command	یک فرمان سیستم عاملی رادرمحیط SQL*Plus اجرا می کند
CONN[ECT]useid/pass	باعث برقراری اتصال دیگری ، از طریق اتصال فعلی میگردد
EXIT	باعث خروج ازمحیط SQL*Plus میگردد
PRCPMPT text	این دستور ، معمولازمانی مورداستفاده قرارمی گیرد که بخواهیم به هنگام اجرای یک فایل حاوی فرامین SQL متنی رابرروی صفحه نمایش نشان دهیم.

دستورات عمومی زبان SQL عبارتنداز:

SELECT متداولترین دستوراست که برای بازیابی داده ها ازیک بانک اطلاعاتی مورداستفاده قرارمی گیرد ,DELETE UPDATE,INSERT ازاین دستورات ، بترتیب برای درج ، اصلاح وحذف سطرهای یک جدول از بانک اطلاعاتی استفاده می شود این دستورات رافرامین DML (Data Manipulation Language) می نامند.

DROP,ALTER,CREATE ازاین دستورات بترتیب برای ایجاد اصلاح وحذف ساختارهای داده ای نظیرجداول (Tables)، نمایه ها (Views)، شاخص ها (Indexes) استفاده می شود این دستورات رافرامین DDl(Data Definition Language) می نامند.

REVOKE, GRANT ازاین دستورات برای دادن اجازه های دستیابی به ساختارهای داده ای دریک بانک اطلاعاتی ویابازپس گیری آنها ، استفاده میشود این دستورات رافرامین ، DDL (Data Definition Language) می نامند.

قواعد نوشتن دستورات SQL :

- این فرامین می توانندد‌ریک یاچندخط نوشته شوند.
- بهتراست هرعبارت (Clause) دریک خط جداگانه نوشته شود.
- می توان عبارتها را بافاصله (tab) ازهم نوشت .
- بین حروف کلمه های فرمان ، نباید فاصله ای وجودداشته باشد.
- درنوشتن فرامین SQL محدودیتی بلحاظ حروف بزرگ وکوچک وجودندارد
- یک دستور SQL درمقابل اعلان SQL درج شده وخطوط بعدی ، شماره گذاری می شوند آخرین مجموعه خطوط درج شده تحت یک عبارت SQL دربافرقرارمی گیرند.
- عبارت موجود دربافر به روشهای زیرقابل اجراست.
  - باقراردادن علامت : (semicolon) درانتهای عبارت
  - باقراردادن علامت : (semicolon) یا / (slash) درآخرین خط بافر
  - بادرج کلمه R[un] درمقابل اعلان SQL

درزیرمثالهایی از عبارتهای SQL راملاحظه می نمایید.

```
SQL> SELECT  *  FROM  EMP;
SQL> SELECT  *
2    FROM
3    EMP
4    ;
SQL> SELECT   *
2    FROM  EMP;
```

بدنه یک پرس جوشامل دستوراتی است جهت انجام محاسبات ریاضی ، برخوردصحیح بامقادیرنول (unll) ، تعیین اسامی مستعار (alias) برای ستونها ، بهم چسباندن (concatenate) ستونهای کاراکتری ونیز شامل فرامینی است برای مرتب کردن سطرهای بازیابی شده از جدول.

عبارت where برای محدود کردن سطرهای حاصل ازیک پرسوجو براساس شرایط مشخص ، مورد استفاده قرار می گیرد.

دستور SELECT میتواند اطلاعات راازیک بانک اطلاعاتی بازیابی کرده وتمامی عملگرهای (operators) جبررابطه ای رابرروی آنها اعمال نماید. ساده ترین شکل آن شامل عبارتهای زیراست.

- یک عبارت SELECT بهمراه لیست ستونهایی که بایدنمایش داده شوند.
- یک عبارت From که مشخص کننده نام جدول است .

به عنوان مثال برای نشان دادن ، شماره دپارتمان ، نام کارمندوشماره مدیرازجدول EMP بایددستورزیررا واردنمایید.

```
SELECT  DEPTNO, ENAME,MGR
FROM  EMP ;
```

برای نمایش همه ستونهای یک جدول بایدازعلامت * استفاده نمود.

علاوه برموارد ذکرشده یک خط SELECT می تواندشامل عناصرزیرنیزباشد.

- عبارتهای محاسباتی
- اسامی مستعار (ahas) برای ستونها
- ستونهای بهم چسبیده (concatenated)
- رشته های متنی – عددی (literal)
- توابع

عبارتهای محاسباتی :

عبارتهای محاسباتی می تواندشامل اسامی ستونها ، مقادیرعددی ثابت وعملگرهای محاسباتی باشد.

عملگر های محاسباتی	
+	جمع
-	تفریق
*	ضرب
/	تقسیم

عملگرهای * و / دارای اولویت بالاتری نسبت به + و - بوده وبرای عملگرهای بااولویت یکسان نیز ، محاسبه از چپ به راست خواهندبود میتوان ازپرانتزها برای تعیین اولویت اجرای محاسبات استفاده نمود.

```
SELECT  ENAM,SAL*12, COMM
FROM  EMP ;
```

اسامی مستعار (alias) برای ستونها:

SQL*Plus بهنگام نمایش نتایج حاصل ازیک پرس وجو بطورمعمول عنوان ستون رابانام آن مشخص میسازد بسیاری ازاوقات بهتراست که عنوان موردنظر باکلمه گویاتری ، تغییریابداین کارباتعیین یک نام مستعار (alias) صورت می پذیرد. درستور SELECT نام مستعار پس ازنام ستون قرارمی گیرد درخروجی نام مستعار بطورپیش فرض دارای حروف بزرگ خواهدبود مگرآنکه بین دوعلامت گیومه (" ") قرارگیرد.

```
SELECT  ENAME,SAL*12 ANNSAL, COMM
FROM  EMP;
```

ستونهای بهم چسبیده (concatenated):

ازعملگر || میتوان برای الصاق ستونها ، عبارتهای محاسباتی ویامقادیرثابت به یکدیگر جهت ایجاد یک عبارت رشته ای ، استفاده نمود بعنوان مثال برای الصاق ENAME, EMPNO ودادن نام مستعار EMPLOYEE به حاصل ترکیب بایددستورزیررادرج نمایید.

```
SELECT  EMPNO || ENAME EMPLOYEE  FROM EMP;
```

رشته های متنی – عددی (LITERAL):

یک رشته متنی – عددی عبارت است ازمجموعه ای ازکاراکترویااعدادموجود دریک دستور SELECT که مبین نام ستونی نباشد چنین رشته ای درتمامی سطرهای حاصل ازیک پرس وجو تکرار میگردد رشته های کاراکتری وتاریخ ، بایدبین دوعلامت کوتیشن (' ') قرارگیرند ولی رشته های عددی نیازی به کوتیشن ندارند.

```
SELECT  EMPNO||'-'||ENAME EMPLOYEE ,'WORK IN DEPARTMENT',DEPTNO
FROM  EMP;
```

توابع (functions) :

هرتابع دارای تعدادی آرگومان است که درداخل پرانتز ودرجلوی نام آن قرارمی گیرند این آرگومانها می توانند شامل مقادیرستونها ویارشته های متنی- عددی (literal) باشند.

نحوه کاربامقادیر نول (NULL) :

نول یعنی خالی وستونی که مقدارنگرفته باشد ، می گویننددارای مقدار نول است وواضح است که این مفهوم ، متمایز ازمقدارصفرویا بلانک(blenk) است. درزبان SQL مقادیرنول به روشی صحیح ، موردپردازش قرارمی گیرند اگریک عبارت شامل ستونی باشد که دارای مقدارنول است حاصل آن عبارت نیز ، معادل نول خواهد بود مثال زیر ، گویای این مطالب است:

```
SELECT  ENAME,SAL*12 + COMM ANNUAL_SAL,COMM,SAL FROM  EMP ;
```

```
SQL> select ename,sal*12+comm annual_sal,comm,sal from emp;

ENAME      ANNUAL_SAL       COMM        SAL
---------- ---------- ---------- ----------
SMITH                             800
ALLEN           19500        300       1600
WARD            15500        500       1250
JONES                            2975
MARTIN          16400       1400       1250
BLAKE                            2850
CLARK                            2450
SCOTT                            3000
KING                             5000
TURNER          18000          0       1500
ADAMS                            1100

ENAME      ANNUAL_SAL       COMM        SAL
---------- ---------- ---------- ----------
JAMES                             950
FORD                             3000
MILLER                           1300

14 rows selected.
```

اگربخواهیم ستون ANNUAL_SAL به ازای هیچیک ازسطرها خالی نباشد بایدمقدارنول را به یک عدد تبدیل نماییم. اینکارباستفاده از تابع NVL صورت می پذیرد. برای تبدیل مقادیرنول به صفر درعبارت فوق بایدآن رابصورت زیر ، اصلاح نماییم.

SELECT ENAME, SAL*12 + NVL(COMM,0) ANNUAL_SAL,COMM,SAL FROM EMP;

```
ENAME      ANNUAL_SAL
---------- ----------
SMITH            9600
ALLEN           19500
WARD            15500
JONES           35700
MARTIN          16400
BLAKE           34200
CLARK           29400
SCOTT           36000
KING            60000
TURNER          18000
ADAMS           13200

ENAME      ANNUAL_SAL
---------- ----------
JAMES           11400
FORD            36000
MILLER          15600

14 rows selected.
```

تابع NVL دارای دوآرگومان است یک عبارت ویک مقدار غیرنول.

ازتابع NVL بتوان برای تبدیل مقادیرنول ، تاریخ ویارشته های کاراکتری به مقادیرمتناظر بانوع داده ای آنها استفاده نمود.

NVL(DATECOLUMN,01-JAN-88)

NVL(NUMBERCOLUMN.9)

ممانعت از انتخاب سطرهای تکراری بااستفاده از عبارت DISTLNCT :

دریک پرس وجو برای جلوگیری ازنمایش سطرهای تکراری برحسب تمام ستونهایی کـه نمـایش داده مـی شـوند ، بایـدازعبارت DISTINCT دردستور SELECT استفاده نمایید . بعنوان مثال برای ممانعت ازنمایش مقـادیرتکراری DEPTNO بهنگـام پرس وجو ازجدول EMP بایددستور زیرراوارد نمایید:

```
SQL> SELECT  DISTINCT DEPTNO FROM  EMP ;

    DEPTNO
----------
        10
        20
        30
```

درعبارت DISTINCT میتوان بیش ازیک ستون رامشخص نمود که دراین حالت عبارت DISTINCT برروی تمامی ستونهای مربوطه ، اعمال می گردد بعنوان مثال ، برای جلوگیری ازنمایش مقادیرتکراری DEPTNO و JOB بایددستور زیرراوارد نماید.

SELECT DISTINCT DEPTNO, JOB FROM EMP;

```
SQL> select distinct deptno,job from emp;

    DEPTNO JOB
---------- ---------
        10 CLERK
        10 MANAGER
        10 PRESIDENT
        20 ANALYST
        20 CLERK
        20 MANAGER
        30 CLERK
        30 MANAGER
        30 SALESMAN

9 rows selected.

SQL> Power By www.wbs.ir
```

فهرست مذکور ، ترکیبات مختلف شغلها وشماره دپارتمانها رانشان می دهد.

عبارت ORDER EY :

بطورمعمول ، نمایش سطرها دریک پرس وجود دارای ترتیب مشخصی نیست . ازعبارت ORDER BY می تـوان بـرای مرتـب کردن سطرهای حاصل ازیک پرس وجو استفاده نمود. این عبارت ، همیشه درانتهای دستور SELECT قرارمی گیرد بعنـوان مثال برای مرتب کردن نتایج پرس وجو برحسب ENAME باید دستورزیررا وارد کنید.

SELECT ENAME,JOB, SAL*12, DEPTNO FROM EMP
ORDER BY ENAME;

بطورپیش فرض ، اطلاعات ، بصورت صعودی مرتب می شوند که دراینحالت نخست اعداد ، وسپس تاریخ ، وبدنبال آن رشته های کاراکتری ، نمایش داده می شوند برای تبدیل حالت صعودی به نزولـی ، بایدبعـدازنام سـتون درعبـارت ORDER BY کلمـه DESC رابکاربریم درمثال زیر ، اطلاعات برحسب ستون HIREDATE (تاریخ استخدام) بطورنزولی مرتب شده اند.

SELECT ENAME, JOB , HIREDATE
FROM EMP ORDER BY HIREDATE DESC;

می توان ازعبارت ORDER BY بابیش ازیک ستون استفاده نمود. ستونها باویرگول (,) ازهم جدامی شوند دراینحالت سطرهایی که ستون اول ORDER BY درآنها یکسان باشد برحسب ستون دوم به بعد ، مرتب می گردند درمثال زیر اطلاعات ، بترتیب برحسب ستونهای DEPTNO و SAL بصورت نزولی مرتب شده اند.

```
SELECT  DEPTNO, JOB, ENAME, SAL
FROM  EMP
ORDER BY DEPTNO DESC , SAL DESC;
```

```
SQL> select deptno,job,ename,sal
  2  from emp
  3  order by deptno desc,sal desc;

    DEPTNO JOB       ENAME             SAL
---------- --------- ---------- ----------
        30 MANAGER   BLAKE            2850
        30 SALESMAN  ALLEN            1600
        30 SALESMAN  TURNER           1500
        30 SALESMAN  WARD             1250
        30 SALESMAN  MARTIN           1250
        30 CLERK     JAMES             950
        20 ANALYST   SCOTT            3000
        20 ANALYST   FORD             3000
        20 MANAGER   JONES            2975
        20 CLERK     ADAMS            1100
        20 CLERK     SMITH             800

    DEPTNO JOB       ENAME             SAL
---------- --------- ---------- ----------
        10 PRESIDENT KING             5000
        10 MANAGER   CLARK            2450
        10 CLERK     MILLER           1300

14 rows selected.

SQL> powerd by www.wbs.ir
```

برای مرتب کردن برحسب یک یاچندستون ، نیازی به انتخاب آنها درعبارت SELECT نیست.

عبارت WHERE :

برای محدودکردن نتایج حاصل ازیک پرس وجو ازاین عبارت ، بهمراه درج شرایط موردنظر ، استفاده می شود درصورت استفاده ازآن ، شکل دستورچنین خواهد بود.

```
SELECT  columns
FROM  table
WHERE  certain conditions are met
```

درجلوی عبارت WHERE ازهریک ازسه عنصر زیرمی توان استفاده نمود.

۱) نام یک ستون

۲) یک عملگر مقایسه ای

۳) نام یک ستون ، یک مقدار ثابت ویالیستی از مقادیر

عملگر های مقایسه ای بردوقسمند : منطقی (Logical) و SQL

عملگر های منطقی:

عملگر های منطقی	
-	مساوی با
>	بزرگتراز
>=	بزرگتریامساوی
<	کوچکتراز
<=	کوچکتریامساوی

رشته های کاراکتری وتاریخ ، در عبارت WHERE ، باید بین دو علامت کوتیشن قرار گیرند بعنوان مثال ،برای نمـایش نـام ، شماره کارمندی وشماره دپارتمان تمام کسانی که شغل آنها منشی گیری (CLERK) است باید دستور زیرادرج نماییم:

```
SELECT  ENAME , EMPNO, DEPTNO
FROM  EMP
WHERE JOB= 'CLERK' ;
```

درمثال دیگر ، برای نمایش نام وشماره دپارتمانهایی که شماره دپارتمان آنها بزرگتراز ۲۰ است بایددستورزیررا درج نماییم:

```
SELECT  DNAME, DEPTNO
FROM  DEPT
WHERE DEPTNO>20 ;
```

```
DNAME              DEPTNO
-------------- ----------
SALES                  30
OPERATIONS             40
```

می توان درهرسطرازجدول ، ستونی را باستون دیگر ، مقایسه نمـود . بعنـوان مثـال بـرای نمـایش کارمنـدانی کـه میـزان حـق ماموریتشان (commission) بیشترازمیزان حقوق آنهاست ، بایددستور زیرادرج نماییم.

```
SELECT  ENAME , SAL,  COMM
FROM  EMP
WHERE COMM> SAL  ;
```

```
ENAME             SAL       COMM
---------- ---------- ----------
MARTIN            1250       1400
```

عملگر های SQL :

چهارنوع عملگر SQL وجوددارد که باهمه انواع داده ای کار می کنند این عملگر ها عبارتنداز:

عملگرهای SQL	
BETWEEN	وجود بین دومقدارمشخص
IN	معادل هر یک ازمقادیرلیست مشخص شده
LIKE	منطبق بایک الگوی کاراکتری
IS NULL	آیا معادل مقدارنول است؟

عملگر BETWEEN :

عملگر BETWEEN وجودمقادیر بین دومحدوده راتست می کند فرض کنید بخواهیم کارمندانی را که حقوق آنها بین ۱۰۰۰ و 2000 است ، نشان دهیم برای اینکار بایددستور زیرراوارد نماییم:

```
SELECT  ENAME , SAL FROM  EMP
WHERE SAL BETWEEN 1000 AND 2000;
```

توجه داشته باشید که دوعدد مشخص شده فوق نیز جزء محدوده بوده وعدداول باید همیشه کوچکترازعددوم باشد.

عملگر IN :

عملگر IN وجودمقادیر معادل بایک لیست مشخص شده رابررسی می کند بعنوان مثال برای نمایش همه کارمندانی که دارای شماره MGR مشخصی باشند بایددستور زیرراوارد نمایید.

```
SELECT  EMPNO , ENAME , SAL , MGR
FROM  EMP
WHERE MGR IN (7902,7566, 7788 );
```

```
SQL> select empno,ename,sal,mgr
  2  from emp
  3  where mgr in(7902,7566,7788);

     EMPNO ENAME             SAL         MGR
---------- ---------- ---------- ----------
      7369 SMITH             800        7902
      7788 SCOTT            3000        7566
      7876 ADAMS            1100        7788
      7902 FORD             3000        7566

SQL>
```

درصورت استفاده از کاراکتر یاتاریخ درلیست فوق ، بایدآنها را بین دو کوتیشن قراردهیم.

عملگر LIKE :

گاهی اوقات ممکن است مقدارواقعی موردجستجو راندانیم. بااستفاده ازاین عملگر، میتوان سطرهایی را که دارای ستونهای مشابه بایک الگوی کاراکتری مشخص هستند بدست آورد ، این الگوی کاراکتری می تواندشامل هریک ازدونماد % یا _ (underscore) باشد. نماد اول بجای رشته ای است به طول صفر کاراکتر یابیشتر ، ونماد دوم نیز دقیقا بجای یک کاراکتر قرارمی گیرد.

مثال (۱) برای نمایش اسامی کارمندانی که نامشان با حرف S شروع میشود بایددستور زیررابکاربریم:

```
SELECT  ENAME  FROM EMP
WHERE ENAME LIKE 'S%' ;
```

مثال (۲) برای نمایش اسامی کارمندانی که نامشان دقیقا چهار کاراکتری است بایددستور زیررابکاربریم:

```
SELECT  ENAME  FROM  EMP
WHERE ENAME LIKE '____' ;
```

عملگر IS NULL :

عملگر IS NULL وجودمقادیرنول راتست می کند . بعنوان مثال برای نمایش کارمندانی که دارای مدیرنیستند (یعنی فیلد MGR آنها خالی است) بایددستورزیررا وارد نماییم :

```
SELECT  ENAME , MGR
 FROM  EMP
WHERE MGR IS NULL ;
```

منفی کردن عبارتها:

ازعملگرهای زیربرای انجام تست درحالت نفی استفاده می گردد.

عملگر های SQL برای تست حالت نفی ، عبارتنداز :

عملگر های نفی SQL	
NOT BETWEEN	عدم وجود بین دومقدارمشخص
NOT IN	عدم وجوددرلیست مشخص شده
NOT LIKE	عدم انطباق بایک الگوی کاراکتری
IS NOT NULL	آیا دارای مقدارنول نیست.

مثال (۱) برای نمایش کارمندانی که میزان حقوقشان دریک محدوده معین نباشد بایددستور زیررادرج نماییم:

```
SELECT  ENAME, SAL FROM  EMP
WHERE SAL NOT BETWEEN 1000 AND 2000 ;
```

```
ENAME                SAL
---------- ----------
SMITH                800
JONES               2975
BLAKE               2850
CLARK               2450
SCOTT               3000
KING                5000
JAMES                950
FORD                3000

8 rows selected.
```

مثال (۲) برای نمایش آن دسته از کارمندانی که شغل آنها ، باحرف M شروع نمی شود بایدعبارت زیرراوارد نماییم:

```
SELECT  ENAME , JOB  FROM  EMP
WHERE JOB NOT LIKE 'M%' ;
```

مثال (۳) برای نمایش کارمندانی که دارای مدیرهستند (فیلد MGR آنها ، خالی نیست) باید دستورزیررا واردنماییم.

```
SELECT  ENAME, MGR
FROM  EMP
WHERE MGR IS NOT NULL ;
```

پرس وجوی داده ها بابیش ازیک شرط :

میتوان از عملگر های AND (اشتراک) و OR (اجتماع) برای ایجاد عبارتهای منطقی ترکیبی استفاده نمود . حاصل اشتراک (AND) دوعبارت ، وقتی درست (true) است که هردوعبارت ، درست (true) باشند. نیزحاصل اجتماع(OR) دوعبارت ، وقتی نادرست (false) است که هردوعبارت نادرست (false) باشند.

مثال (I) می خواهیم کارمندانی راکه هم منشی بوده وهم حقوقشان بین ۱۰۰۰ و ۲۰۰۰ است نشان دهیم:

```
SELECT  EMPNO , ENAME, JOB, SAL
FROM  EMP
WHERE SAL BETWEEN 1000 AND 2000
AND JOB='CLERK' ;
```

مثال (۲) برای نمایش کارمندانی که منشی بوده وبامیزان حقوقشان بین ۱۰۰۰ و ۲۰۰۰ است بایدعبارت زیررا درج نماییم:

```
SELECT ENPNO , ENAME , JOP , SAL
FROM  EMP
WHERE SAL BETWEEN 1000 AND 2000
OR  JOB = 'CLERK'  ;
```

ازآنجائیکه اولویت عملگر AND ، بیشتراز OR است بنابراین مثال زیر همه مدیرانی را که حقوقشان بیشتراز 1500 دلاراست ونیزتمامی کارمندانی را که شغل آنها ، فروشندگی است ، نشان می دهد :

```
SELECT  ENPNO , ENAME , JOB , SAL , DEPTNO
FROM  EMP
WHERE  SAL >1500
AND JOB = 'MANAGER'
OR   JOB = 'SALESMAN' ;
```

اگربخواهیم همه مدیران وفروشندگانی را که حقوقشان بیشتراز 1500 دلاراست نشان دهیم ، بایددستور زیررادرج نماییم :

```
SELECT  EMPNO , ENAME , JOB , SAL , DEPTNO
FROM  EMP
WHERE SAL >1500
AND ( JOB = MANAGER OR   JOB = SALESMAN) ;
```

اولویت عملگر ها :

عملگر های موجوددریک عبارت ، برحسب اولویتشان اجرامی شوند عملگر هایی که دارای اولویت یکسان باشند ، ازچپ به راست اجرا می گردند ترتیب اولویت عملگر ها ازبالا به پایین ، چنین است :

۱)تمام عملگر های SQL ومقایسه ای دارای اولویت یکسانند.

=,!=,<,>,<=,>=, BETWEEN ... AND ,IN , LIKE, IS NULL

۲) NOT (برای نفی کردن یک عبارت منطقی

۳) AND

۴) OR

هرگاه دراولویت عملگر ها دچار تردید شدید می توانید از پرانتزها استفاده نمایید که دراینحالت برگویایی عبارت افزوده خواهدشد بعنوان مثال برای نشان دادن تمامی مدیران دپارتمانها ، بهمراه همه افرادمنشی موجود دردپارتمان شماره 10 باید ازدستور زیراستفاده نماییم.

```
SELECT  *
FROM  EMP
WHERE JOB = 'MANAGER'  OR ( JOB='CLERK' AND DEPTNO =10)
```

البته درعبارت فوق ، نیازی به استفاده از پرانتزنیست زیرا اولویت عملگر AND ، بالاتراز OR است ولی همانگونه که ملاحظه می گردد استفاده ازآن ، برگویایی عبارت ، کاملا افزوده است.

فصل دوم : اجرای پرس وجوهای استاندارد باجایگذاری متغیرها

متغیرهای جایگزین (Substitution variables) :

دردستور SELECT می توان ازمتغیرهای جایگزین ، برای نمایش مقادیری که باید بهنگام اجرا فراهم آید ، استفاده نمود. چنین متغیری دارای پیشوند & می باشد درمثال زیر بهنگام اجرای دستور ، شماره دپارتمان از کاربر درخواست می گردد.

```
SELECT  EMPNO , ENAME , SAL
FROM  EMP
WHERE DEPTNO = &DEPARTMENT_NUMBER;
Enter value for department_number :10
```

```
SQL> select empno,ename,sal
  2  from emp
  3  where deptno=&department_number;
Enter value for department_number: 10
old   3: where deptno=&department_number
new   3: where deptno=10

     EMPNO ENAME              SAL
---------- ---------- ----------
      7782 CLARK            2450
      7839 KING             5000
      7934 MILLER           1300

SQL>
```

درمثال فوق ، بهنگام اجرا ، ازشرط WHERE DEPTNO=10 استفاده خواهد شد درصورت استفاده ازیک علامت & به ازای هرباراجرای دستور ، مقدار متغیر ، ازکاربر درخواست می گردد.

مقادیر کاراکتری وتاریخ ، بهنگام اجرا ، باید بین دوعلامت کوتیشن قرارگیرند. اگربخواهیم ازدرج کوتیشن ، درزمان اجرا ، صرفنظرکنیم ، بایدخود متغیر را بین دو کوتیشن قراردهیم . مثال زیر گویای این مطلب است :

```
SELECT  ENAME, DEPTNO , SAL*12
FROM  EMP
WHERE JOB = & JOB_TITLE;
Enter value for job_title :MANAGER
```

درمثال زیر ، از کاربر درخواست می گردد که بهنگام اجرا ، یک عبارت محاسباتی را وارد نماید.

```
SELECT  DEPTNO , & ARTTHMETIC_EXPRESSION
FROM  EMP ;
Enter value for & arithmetic expression: sal/12
```

اگر متغیری دارای پیشوند && باشد دراینحالت SQL*Plus فقط یکبار مقدارآن را از کاربر درخواست می کند چنین متغیری درسطح سیستم تعریف شده ومقدارآن به ازای هرباراجرای دستور SQL مجددا مورداستفاده قرارمی گیرد.

```
SELECT  ENAME , DEPTNO , JOB
FROM  EMP
WHERE DEPTNO =&& DEPTNO_PLEASE;
Enter value for deptno_please :10
```

می توان ازدستور DEFINE برای تعیین مقداریک متغیر استفاده نمود. نحوه ارجاع به این متغیر با پیشوند& امکان پذیر خواهدبود . درمثال زیر یک عبارت محاسباتی بعنوان مقدار یک متغیر تعریف شده است .

```
SQL>DEFINE    REM= 'SAL*12 +NVL (COMM,0)'
SQL> SELECT   ENAME , JOB ,& REM
FROM  EMP
ORDER BY & REM ;
SQL> UNDEFINE REM
```

اگربخواهیم یک متغیر کاراکتری تعریف کنیم باید بصورت زیرعمل نموده و عبارت موردنظر رابین دوگیومه (double qoutation) قراردهیم.

```
SQL>DEFINE REM= " SAL*12 +NVL(COMM,0)"  (CHAR)
SQL> UNDEFINE REM
```

اجرای فایل حاوی فرامینی که شامل متغیرهای جایگزین باشد :

فرض کنید که بخواهیم لیستهای مختلفی ازکارمندان رابرحسب شغل آنها ، داشته باشیم برای اینکار می توان دستوراتی دریک فایل قرارداد که بهنگام اجرای آن فایل ، توسط فرمان START ، مقادیر پارامترها ، ازکاربردریافت گردد. چنین کاری باقراردادن علامت & بهمراه یک شماره (مثل '&1') امکان پذیر خواهد شد .شماره مربوطه ،معرف ترتیب یا سریالیتی پارامتر است . بنابراین '&1' بمعنی پارامتر اول است . برای قراردادن یک عبارت SQL در یک فایل بایدازروال زیرپیروی نمایید.

```
SELECT  ENPNO , ENAME , SAL
FROM  EMP
WHERE JOB ='&1' ;
SQL>SAVE JOB1
```

دراین حالت ، پیغام زیر ، نشان داده خواهد شد .

```
Created file job1
```

اکنون می توانید فایل JOB1 رابا پارامتر CLERK اجرا نمایید.

```
SQL>START JOB1  CLERK
```

یک فرمان ذخیره شده ، حداکثر از نه پارامتر ('&1' تا '&9') میتوانداستفاده نماید . توجه داشته باشید که درصورت اجرای فرمان بادستورRUN نمی توانید ازاین پارامترها استفاده کنید بلکه باید فرامین رادرفایلی ذخیره نموده و سپس بادستور START آن رااجرا نمایید.

فرمان ACCEPT

این دستور موجب می گردد که متغیری ایجادشده ومقداری که بهنگام اجراوارد می شود درآن ذخیره گردد.

این دستور ، اغلب دریک فایل فرمان بکارمی رود. مزایای استفاده ازفرمان ACCEPT عبارتنداز :

- میتوان ازاعلان های دیگری جهت گویایی بیشتراستفاده نمود.
- مقادیرپاسخ داده شده توسط کاربر می تواند پنهان (hidden) باشد .
- نوع داده ای قابل وارسی (check) است .

شکل دستور به صورت زیر است :

ACC[EPT] variable [NUMBER | CHAR][PROMPT | NOPROMPT 'text'] [HIDE]

NUMBER | CHAR : مشخص کننده نوع متغیر است . در صورت نامعتبر بودن مقدار ورودی ، پیغام خطا ظاهر خواهد شد.

PROMPT 'text' : متن مشخص شده را بهنگام اجرا نشان می دهد .

NORRCMPT : مکان نما (cursor) رابه خط بعدی منتقل نموده ومنتظر ورود یک مقدارخواهد بود.

HIDE مقادیرورودی را نشان نمی دهد.

مثال ها :

```
SQL>ACCEPT SALARY NUMAER PROMPT 'Salary Figure :'
Salay figure :3000
SQL>ACCEPT PASSWORD CHAR PROMPT 'Password :'  HIDE
Password :
SQL>ACCEPT COMM  NUMBER NOPROMPT
500
SQL> DEETNE
DEFINE SALARY = 3000 (NUMBER)
DEFINE PASSWORD = 'FREEBIES' (CHAR)
DEFINE COMM = 500 (NUMBER)
```

فصل سوم : توابع

توابع ، برای کارباعناصر داده ای مورداستفاده قرارمی گیرند. هرتابع ، یک یاچندآرگومان ، دریافت کرده ویک مقدار برمی گرداند. آرگومان می تواند یک مقدارثابت ، یک متغیر ویایک ستون جدولی باشد . شکل یک تابع ، چنین است :

Function_Name (argument1, argument2)

ازتوابع ، برای موارد‌زیر، میتوان استفاده نمود :

- اجرای محاسبات برروی داده ها
- اصلاح عناصرداده ای
- تنظیم خروجی برای گروهی از سطرها
- تغییرفرمت تاریخ
- تبدیل نوع داده ای یک ستون

انواع توابع عبارتنداز :

- کاراکتری
- عددی
- تاریخ
- تبدیل
- توابعی که هریک ازانواع داده ای رابعنوان ورودی می پذیرند.
- گروه (GROUP)

برخی ازتوابع فقط برروی یک سطر ، کارکرده وبرخی دیگر نیزمی تواند برروی گروهی ازسطرها ، اعمال شوند.

توابعی که فقط برروی یک سطرکارمی کنند :

- برروی هرسطرحاصل ازیک پرس وجواعمال می شوند.
- به ازای هرسطر ، یک نتیجه برمی گردانند.
- می توانند شامل یک یاچندآرگومان باشند.
- می توانند بصورت تودرتو باشند.
- درهرنقطه ازعبارت SQL قابل استفاده اند.

توابع حرفی یاکاراکتری

این توابع ، دادهای حرفی رابعنوان ورودی پذیرفته ومقادیرحرفی یاعددی رابرمی گردانند :

: LOWER (col | value)

مقادیر کاراکتری حاوی حروف بزرگ را به حروف کوچک تبدیل می کند.

SELECT LOWER(DNAME) FROM DEPT ;

:UPPER(col | value)

مقادیر کاراکتری حاوی حروف کوچک را به حروف بزرگ ، تبدیل می کند.

```
SELECT  ENAME
FROM  EMP
WHERE ENAME= UPPER('&ENAME') ;
```

: INITCAP (col | value)

نخستین حرف هرکلمه را به حروف بزرگ ، و بقیه را به حروف کوچک تبدیل می کند.

```
SELECT  INTTCAP (DNAME), INITCAP (LOC)
FROM  DEPT ;
```

```
INITCAP(DNAME) INITCAP(LOC)
-------------- -------------
Accounting     New York
Research       Dallas
Sales          Chicago
Operations     Boston
```

: LPAD(col | value ,n,'string')

ستون یامقدارموردنظر را ازسمت چپ ، مجموعا به اندازه n کاراکتر گسترش می دهد فضاهای خالی با عبارت string پر می شوند درصورت حذف string فضای موجود با بلانک پرمی شود.

```
SELECT  LPAD(DNAME,20, '*') ,LPAD(DNAME,20), LPAD(DEPTNO,20,'-')  FROM DEPT;
```

```
RPAD(DNAME,20,'*')   RPAD(DNAME,20)       RPAD(DEPTNO,20,'-')
-------------------- -------------------- --------------------
ACCOUNTING********** ACCOUNTING           10------------------
RESEARCH************ RESEARCH             20------------------
SALES*************** SALES                30------------------
OPERATIONS********** OPERATIONS           40------------------
```

: RPAD(col | value ,n,'string')

ستون یامقدارموردنظر را ازسمت راست ، مجموعا به اندازه n کاراکتر ، گسترش می دهد ، فضاهای خالی با عبارت string پر می شوند درصورت حذف string فضای موجود با بلانک پرمی شوند.

```
SELECT  LPAD(DNAME,20, '*'),LPAD(DNAME,20), LPAD(DEPTNO,20,'-') FROM  DEPT ;
```

```
RPAD(DNAME,20,'*')   RPAD(DNAME,20)       RPAD(DEPTNO,20,'-')
-------------------- -------------------- --------------------
ACCOUNTING********** ACCOUNTING           10------------------
RESEARCH************ RESEARCH             20------------------
SALES*************** SALES                30------------------
OPERATIONS********** OPERATIONS           40------------------
```

توجه داشته باشید که ستون دوم درحالت عادی نیز دارای ستونهای بلانک درسمت راست بوده و برای این ستون نیازی به استفاده ازتابع RPAD نیست.

: SUBSTR (col | value,pos,n)

زیرنوشته ای بطول n کاراکتر ازرشته موردنظر راباشروع ازموقعیت POS بر می گرداند.

```
SELECT  SUBSTR('ORACLE',2,4) , SUBSTR(DNAME,2),SUBSTR(DNAME,3,5)
FROM  DEPT ;
```

```
SUBS SUBSTR(DNAME, SUBST
---- ------------- -----
racl CCOUNTING     COUNT
racl ESEARCH       SEARC
racl ALES          LES
racl PERATIONS     ERATI
```

اوراکل ، بطور پیش فرض داده های کاراکتری راازچپ ، ترازمی کند.

INSTR (col | value,'string') :

موقعیت اولین وقوع رشته string رادرمقدار ورودی مشخص می سازد.

INSTR (col | value , 'string', pos,n) :

موقعیت n امین وقوع رشته string رادرمقداروردی وازستون pos به بعد مشخص می سازد.

```
SELECT  DNAME, INSTR (DNAME, 'A'),INSTR(DNAME ,'ES'),
INSTR(DNAME ,'C',1,2)
FROM  DEPT;
```

DNAME	INSTR(DNAME,'A')	INSTR(DNAME,'ES')	INSTR(DNAME,'C',1,2)
ACCOUNTING	1	0	3
RESEARCH	5	2	0
SALES	2	4	0
OPERATIONS	5	0	0

کاربرد معمول INSTR آن است که مشخص گردد آیا مقدار ورودی ، حاوی یک رشته خاص هست یانه ؟ درمثال فوق ، حاصل عبارت INSTR (DNAME,'ES') برای کلمه ACCOUNTING مساوی صفراست ، زیراین کلمه حاوی رشته ES نمی باشد.

LTRIM (col | value , char[s]) :

کاراکتر های موجوددررشته char[s] راازسمت چپ مقداروردی برمی دارد اگررشته char[s] مشخص نگردد ، تمام بلاتک ها ازسمت چپ مقدارمربوطه برداشته می شود.

```
SELECT  DNAME ,LTRIM (DNAME ,'A') , LTRIM(DNAME ,'AS'),
LTRIM(DNAME,'ASOP')
FROM  DEPT;
```

RTRIM (col | value ,char[s]) :

کاراکترهای موجوددررشته char[s] راازسمت راست مقدار ورودی برمی دارد اگررشته char[s] مشخص نگردد.

تمام بلاتک ها ازسمت راست مقدارمربوطه ، برداشته می شود.

```
SELECT  DNAME,RTRIM(DNAME ,'G'),RTRIM(DNAME ,'GHS'),
RTRIM(DNAME ,'N')  FROM  DEPT;
```

DNAME	RTRIM(DNAME,'G	RTRIM(DNAME,'G	RTRIM(DNAME,'N
ACCOUNTING	ACCOUNTING	ACCOUNTING	ACCOUNTING
RESEARCH	RESEARCH	RESEARCH	RESEARCH
SALES	SALES	SALES	SALES
OPERATIONS	OPERATIONS	OPERATIONS	OPERATIONS

معمولا ازتابع RTRIM برای حذف بلاتکهای اضافی ازانتهای مقداوروردی استفاده می گردد بعنـوان مثـال ، فـرض کنیـد کـه بخواهیم مقادیر ENAME ازجدول EMP راباحذف بلاتک های اضافی ازسمت راست مجددا به آن جدول منتقل نماییم بـرای اینکار بایددستورزیررابکاربریم.

```
UPDATE EMP
SET ENAME=RTRIM(ENAME)
```

IENGTH (col | value) :

طول رشته ورودی را برحسب کاراکتر مشخص می سازد.

```
SELECT  LENGTH(DEPTNO ), LENGTH(DNAME)
FROM  DEPT;
```

LENGTH(DEPTNO)	LENGTH(DNAME)
2	10
2	8
2	5
2	10

TRANSLATE (col | value ,from,to) :

هر کاراکتر ازرشته from رابه کاراکتر متناظرباموقعیت آن دررشته to تبدیل میکند.

```
SELECT  ENAME, TRANSLATE(ENAME,'C','P'),JOB,TRANSLATE(JOB,'AR' ,'IT')
FROM  EMP
WHERE DEPTNO =10;
```

ENAME	TRANSLATE(	JOB	TRANSLATE
CLARK	PLARK	MANAGER	MINIGET
KING	KING	PRESIDENT	PTESIDENT
MILLER	MILLER	CLERK	CLETK

توابع تودرتو (Nested Functions) :

توابع مذکور رامی توان به صورت تودرتو مورداستفاده قرارداد درچنین حالتی ، ابتداتوابع داخلی تراجرا می شوند . فـرض کنیـد که بخواهید تعداددفعات تکرار یک کاراکتر خاص را دریک رشته تعیین نماییم.درمثال زیراین عمل برای کاراکتر s انجـام یافتـه است.

```
SELECT  DNAME ,LENGTH(DNAME),LENGTH(DNAME) -
LENGTH(TRANSLATE(DNAME ,'AS','A'))
FROM  DEPT ;
```

دردستورTRANSLATE کاراکتر متناظربا S نول است بنابراین به ازای هر کاراکتر S طول رشته یک واحدکاهش می یابد وباتفاضل طول این رشته ازطول رشته اصلی تعداد دفعات تکرار S بدست می آید.

توابع عددی

این توابع داده های عددی رابعنوان ورودی پذیرفته ومقادیرعددی نیزبرمی گردانند :

ROUND (col | value ,n) :

مقداروروردی رابه n رقم اعشار گرد می کند . اگرعددn حذف گردد ، ارقام اعشاری ازبین خواهندرفت .

```
SELECT  ROUND(45.923,1),ROUND(45.923), ROUND(45.323,-1)
FROM  EMP
WHERE DEPTNO =10;
```

ROUND(45.923,1)	ROUND(45.923)	ROUND(45.323,-1)
45.9	46	50
45.9	46	50
45.9	46	50

TRUNC (col | value ,n) :

مقداروروردی را بهn رقم اعشار ، برش می دهد اگرعددn حذف گردد ارقام اعشاری ازبین خواهندرفت اگرn منفی باشد به تعداد آن ، ارقام صحیح صفرخواهندشد.

```
SELECT  TRUNC(45.923,1), TRUNC(45.923), TRUNC(45.323,-1)
FROM  EMP
WHERE DEPTNO =10;
```

TRUNC(45.923,1)	TRUNC(45.923)	TRUNC(45.323,-1)
45.9	45	40
45.9	45	40
45.9	45	40

CEIL(col | value) :

کوچکترین عددصحیح بزرگتر را بامساوی مقدار ورودی رابدست می آورد.

```
SELECT  CEIL (SAL ), CEIL (99.9), CEIL (101.76), CEIL (-11.1)
FROM  EMP
WHERE SAL BETWEEN 3000 AND 5000;
```

CEIL(SAL)	CEIL(99.9)	CEIL(101.76)	CEIL(-11.1)
3000	100	102	-11
5000	100	102	-11
3000	100	102	-11

FLOOR (col | value) :

بزرگترین عددصحیح کوچکتر یامساوی مقدار ورودی رابدست می آورد.

```
SELECT  FLOOR(SAL), FLOOR(99.9) , FLOOR(101.76) , FLOOR(-11.1) FROM EMP
WHERE FLOOR(SAL) BETWEEN 3000 AND 5000;
```

DNAME	INSTR(DNAME,'A')	INSTR(DNAME,'ES')	INSTR(DNAME,'C',1,2)
ACCOUNTING	1	0	3
RESEARCH	5	2	0
SALES	2	4	0
OPERATIONS	5	0	0

POWER (col | value , n) :

مقدار ورودی رابه توان n می رساند.

```
SELECT  SAL,POWER(SAL,2), POWER(50,.5)
FROM  EMP
WHERE DEPTNO=10;
```

SAL	POWER(SAL,2)	POWER(50,5)
2450	6002500	312500000
5000	25000000	312500000
1300	1690000	312500000

SQRT (col | value) :

ریشه دوم مقدار ورودی رابدست می آورد.

```
SELECT  SAL , SQRT(SAL),SQRT(40)
FROM  EMP
WHERE COMM>10;
```

SAL	SQRT(SAL)	SQRT(40)
1600	40	6.32455532
1250	35.3553391	6.32455532
1250	35.3553391	6.32455532

SIGN(col | value) :

اگر مقدار ورودی منفی باشد عدد1- ، واگرصفرباشد عددصفر واگرمنفی باشد عدد1برمی گرداند.

```
SELECT  SAL-COMM,SIGN(SAL-COMM)
FROM EMP
WHERE DEPTNO=30;
```

SAL-COMM	SIGN(SAL-COMM)
1300	1
750	1
-150	-1
1500	1

ABS(col | value) :

قدرمطلق مقدار ورودی را برمی گرداند.

```
SELECT COMM-SAL,ABS(COMM-SAL ),ABS(-35)
FROM  EMP
WHERE DEPTNO =30;
```

COMM-SAL	ABS(COMM-SAL)	ABS(-35)
-1300	1300	35
-750	750	35
150	150	35
		35
-1500	1500	35
		35

MOD(value1,valuel2) :

باقیمانده تقسیم value1 رابر value2 بدست می آورد.

```
SELECT  SAL , COMM,MOD(SAL,COMM)
FROM  EMP
WHERE DEPTNO =30
ORDER BY COMM;
```

SAL	COMM	MOD(SAL,COMM)
1500	0	1500
1600	300	100
1250	500	250
1250	1400	1250
2850		
950		

توابع مربوط به تاریخ :

اوراکل ، تاریخ ها را به یک فرمت عددی که حاوی عناصرقرن رسال ، ماه ، روز ، ساعت ، دقیقه وثانیه است تبدیل می کندفرمت پیش فرض برای نمایش تاریخ ، بشکل DD-MON-YY است محدوده تاریخ دراوراکل بین اول ژوئن ۴۷۱۲ پیش ازمیلاد و31 دسامبر4712 بعدازمیلاد است

SYSDATE یک شبه ستون (pseudo-column) است که تاریخ وزمان جاری را برمی گرداند از SYSDATE می توان بعنوان یک ستون معمولی استفاده کرد بعنوان مثال میتوان آن را ازیک جدول غیرحقیقی (dummy) بعنام DUAL بازیابی نمود. مالک (owner) این جدول کاربر SYSTEM است وتوسط همه کاربران نیز قابل دسترسی است این جدول دارای یک ستون بنام DUMMY ویک سطر بامقدار X است ازاین جدول برای نمایش عبارتهایی استفاده میشود که مقدار آنها از یک ازیک جدول برای نمایش عبارتهایی استفاده میشود که مقدار آنهاازیک جدول حقیقی بدست نمی آید بعنوان مثال برای نمایش تاریخ جاری ، باید عبارت زیرراوارد کنیم.

```
SELECT  SYSDATE FROM  SYSTEM.DUAL;
```

دراین دستور ، بجای DUAL میتوان ازجدولی مثل EMP استفاده نمود که درچنین حالتی ، مقدار SYSDATE بـه تعـداد سطرهای موجوددرجدولEMP نمایش داده خواهد شد امادستورفوق فقط یک سطررانمایش می دهد کـه اغلـب اوقـات چنـین وضعیتی مطلوب وموردنظراست

ازآنجائیکه ، تاریخ ، بصورت عددی ذخیره می گردد ، لذا می توان انواع عملیات محاسباتی رابرروی آن انجام داد.

DATE + تعداد روز = DATE

DATE - تعداد روز = DATE

DATE - DATE = تعداد روز

DATE - تعداد روز /24 = DATE

(24/ تعدادروز) معادل تعدادساعت است که درمثال زیرآمده است.

```
SELECT  TO_CHAR(SYSDATE, 'YY MM DD , hhmmss'),
TO_CHAR(SYSDATE-5/24, 'YY MM DD , hhmmss')
FROM  DUAL;
```

```
TO_CHAR(SYSDATE,' TO_CHAR(SYSDATE,'
---------------- ----------------
07 06 07 , 060653 07 06 07 , 060653
```

```
SELECT  HIREDATE, HIREDATE+7, HIREDATE-7,SYSDATE- HIREDATE
FROM  EMP
WHERE HIREDATE LIKE '%JUN%';
```

```
HIREDATE  HIREDATE+ HIREDATE- SYSDATE-HIREDATE
--------- --------- --------- -----------------
09-JUN-81 16-JUN-81 02-JUN-81        9981.6184
```

MONTES_BETWEEN (date1 , date 2) :

تعدادماههای بین دوتاریخ date1 و date2 رابدست می آورد . نتیجه میتواندمثبت یامنفی باشد.

```
SELECT  MONTHS_BETWEEN (SYSDATE,HIREDATE ),
MONTHS_BETWEEN ('01-JAN-84','05-NOV-88')
FROM  EMP
WHERE  MONTHS_BETWEEN (SYSDATE, HIREDATE )>59;
```

```
MONTHS_BETWEEN(SYSDATE,HIREDATE) MONTHS_BETWEEN('01-JAN-84','05-NOV-88')
-------------------------------- ----------------------------------------
                       333.66524                               -58.12903
                       331.56846                               -58.12903
                       331.50395                               -58.12903
                       330.14911                               -58.12903
```

ADD-MONTHS (date1 , n) :

تعداد n ماه را به تاریخ date1 اضافه می کند . (n عددصحیح بوده ومی تواند منفی باشد)

```
SELECT  HIREDATE ,ADD_MONTHS(HIREDATE ,3),
ADD_MONTHS (HIREDATE ,-3)
FROM  EMP
WHERE DEPTNO =10;
```

```
HIREDATE  ADD_MONTH ADD_MONTH
--------- --------- ---------
09-JUN-81 09-SEP-81 09-MAR-81
17-NOV-81 17-FEB-82 17-AUG-81
23-JAN-82 23-APR-82 23-OCT-81
```

NEXT_DAY(date1,char) :

این تابع ، تاریخ روز مشخص شده توسط مقدارchar را که بلحاظ زمانی بلافاصله بعدازتاریخ data1 قرارداشته باشد بدست می آورد ، مقدار char مبین یکی ازروزهای هفته است ومیتواند بصورت عددی هم باشد که دراین حالت بمعنی n امین روز هفته میلادی خواهدبود.

```
SELECT  HIREDATE ,NEXT_DAY( HIREDATE , 'FRIDAY'),
NEXT_DAY(HIREDATE ,6)
FROM  EMP
WHERE DEPTNO =10;
```

```
HIREDATE  NEXT_DAY( NEXT_DAY(
--------- --------- ---------
09-JUN-81 12-JUN-81 12-JUN-81
17-NOV-81 20-NOV-81 20-NOV-81
23-JAN-82 29-JAN-82 29-JAN-82
```

LAST_DAY (date1) :

آخرین روزازماه میلادی متناظربا date1 را بدست می آورد.

```
SELECT  SYSDATE,LAST_DAY( SYSDATE ), LAST_DAY('15-FEB-88')
FROM  EMP
WHERE DEPTNO =10;
```

```
SYSDATE   LAST_DAY( LAST_DAY(
--------- --------- ---------
25-MAY-07 31-MAY-07 29-FEB-88
25-MAY-07 31-MAY-07 29-FEB-88
25-MAY-07 31-MAY-07 29-FEB-88
```

ROUND(date1):

تاریخ وزمان راباقراردادن ساعت12 ظهربعنوان مبنا گرد می کند.یعنی اگرزمان ، قبل ازساعت 12 ظهر باشد به تاریخ همان روز گردشده واگربعدازساعات 12 ظهرباشد به تاریخ روزبعد گردخواهدشد ازاین حالت میتوان برای مقایسه تاریخهایی که دارای زمانهای نامساوی هستند استفاده نمود.

فرمت تاریخ باید بین دو کوتیشن قرارگیرد ومیتواند شامل هریک ازفرمتهای زیرباشد.

فرمت	شرح
SCC یاCC	قرن پیشوند Sباعلامت – قبل ازBCقرارمی گیرد
YYYY یا SYYYY	سال پیشوند Sباعلامت – قبل ازBCقرارمی گیرد
YYY یا YY ویا Y	سه ، دو ویایک رقم آخرسال
YYY,Y	سال همراه باکاما درموقعیت مربوطه
SYEAR یا YEAR	سال
BC یا AD	معرف قبل ویابعداز میلاد
B.C یا A.D	معرف قبل ویابعدازمیلاد همراه باعلامت نقطه
Q	یک چهارم سال
MM	ماه
MONTH	نام ماه
MON	نام ماه بصورت سه حرفی
WW یا W	هفته سال یاماه
DDD یا DD یا D	روزسال یاماه یاهفته
DAY	نام روز
DY	نام روز بصورت سه حرفی
J	روزJulien تعدادروزها از31 دسامبرسال4713 قبل ازمیلاد
AM یا PM	مشخص کننده صبح یابعدازظهر
HH24	ساعت (صفر تا ۲۳)
MI	دقیقه
SS	ثانیه
SSSSS	تعداد ثانی‌ه از نی‌مه شب (صفر تا ۸۶۳۹۹)

بطورپیش فرض DAY وMONTH بصورت 9 کاراکتری نمایش داده می شوند ودرصورت کوتاهتر بودن طول ، بـا بلاتـک پـر می شوند . برای برداشتن بلاتکها باید ازپیشوند FM (Fill Mode) استفاده نمایید.

```
SELECT  TO_CHAR(SYSDATE, 'fmDay,ddth Month YYYY')
FROM DUAL;
```

```
TO_CHAR(SYSDATE,'FMDAY,DDTHMO
------------------------------
Monday,6th October 2008
```

ازپیشوند fm برای برداشتن صفرهای قبل ازاعداد نیزاستفاده می گردد که درمثـال فـوق 05 بـه 5 تبـدیل شـده اسـت ازتـابع TO_CHAR برای جداکردن زمان ونمایش آن به فرمت دلخواه نیز می توان استفاده نمود.

```
SELECT  TO_CHAR(SYSDATE,'HHMISS')
FROM  DUAL;
```

TO_NUMMBR (char) :

رشته کاراکتری حاوی ارقام را به یک مقدار عددی تبدیل می کند.

```
SELECT  EMPNO,ENAME , JOB , SAL
FROM  EMP
WHERE SAL > TO_NUMBER('1500');
```

EMPNO	ENAME	JOB	SAL
7499	ALLEN	SALESMAN	1600
7566	JONES	MANAGER	2975
7698	BLAKE	MANAGER	2850
7782	CLARK	MANAGER	2450
7788	SCOTT	ANALYST	3000
7839	KING	PRESIDENT	5000
7902	FORD	ANALYST	3000

TO_DATE (char , format) :

رشته کاراکتری حاوی تاریخ رابه یکمقدار تاریخ بافرمت مشخص شده تبدیل می کنداگرفرمت راحذف کنیم ، ازفرمت پیش فرض DD-MON-YY استفاده می گردد. بعنوان مثال برای نمایش کارمندانی که تاریخ استخدام آنها June4,1984 (فرمت غیر پیش فرض) است بایددستور زیرراوارد نماییم:

```
SELECT  EMPNO , ENAME , HIREDATE
FROM  EMP
WHERE HIREDATE =TO_DATE ('June 4,1984' ,'Month dd,yyyy');
```

مقدارثابت تاریخ ، به فرمت متناظر ، تبدیل شده وسپس بامقدار HIREDATE مقایسه می گردد.

توابعی که هریک ازانواع داده ای رابعنوان ورودی می پذیرند:

DECODE(…) : شکل کلی این تابع بصورت زیراست :

```
DECODE (col/ expression,search1, result1,
                [search2 ,result2,…,]
                 Default)
```

این تابع یکی ازپرقدرترین توابع SQL است که نوشتن دستورات شرطی را ، مشابه فرامین case ویا if-then-else تسهیل می کند . مقدارcol/expression باهریک از مقادیرsearch مقایسه شده ودرصورت تساوی مقدار result متناظر ، آن بعنوان خروجی تابع برمی گردد درصورتی که هیچیک از مقادیرsearch مساوی مقدار ورودی نباشند خروجی تابع معادل مقدارdefault خواهدبود نوع داده ای مقداربرگشتی ، معادل نوع داده ای آرگومان result خواهدبود درمثال زیر، فقط شغلهای MANAGER و CLERK دکود شده اندومقدارپیش فرض نیز معادلUNDEFINED می باشد.

```
SELECT      ENAME, JOB, DECODE (JOB ,'CLERK','WORKER'
,'MANAGER','BOSS','UNDEFINED')
 FROM EMP;
```

ENAME	JOB	DECODE(JO
SMITH	CLERK	WORKER
ALLEN	SALESMAN	UNDEFINED
WARD	SALESMAN	UNDEFINED
JONES	MANAGER	BOSS
MARTIN	SALESMAN	UNDEFINED
BLAKE	MANAGER	BOSS
CLARK	MANAGER	BOSS
SCOTT	ANALYST	UNDEFINED
KING	PRESIDENT	UNDEFINED
TURNER	SALESMAN	UNDEFINED
ADAMS	CLERK	WORKER
JAMES	CLERK	WORKER
FORD	ANALYST	UNDEFINED
MILLER	CLERK	WORKER

برای نمایش درصد پاداش (bonus) برحسب رتبه شغلی ، بایدعبارت زیررادرج نمایید.

```
SELECT  GRADE, DECODE(GRADE,'1','15%',
                      '2',' 10%',
                      '3','8%',
                      '5%' ) BONUS FROM   SALGRADE;
```

GRADE	BON
1	15%
2	10%
3	8%
4	5%
5	5%

NVL (col/value ,Val) :

یک مقدار نول را به یک مقدارغیرنول (val) تبدیل می کند نوع داده ای هردو مقدار ورودی بایدیکسان باشد.

```
SELECT  SAL*12 +NVL(COMM,0) ,SAL*12 +NVL(COMM,1000)
FROM  EMP
WHERE DEPTNO =30;
```

SAL*12+NVL(COMM,0)	SAL*12+NVL(COMM,1000)
19500	19500
15500	15500
16400	16400
34200	35200
18000	18000
11400	12400

GREATEST (col | value1,col | value2,…) :

بزرگترین مقدارلیست مشخص شده رابرمی گرداند . نوع داده ای (date type) آرگومانهای دوم به بعد ، پیش از مقایسه به نوع داده ای اولین آرگومان تبدیل می شوند.

```
SELECT  GREATEST (1000,2000),GREATEST( SAL ,COMM)
FROM  EMP
WHERE DEPTNO =30;
```

GREATEST(1000,2000)	GREATEST(SAL,COMM)
2000	1600
2000	1250
2000	1400
2000	
2000	1500
2000	

LEAST (col |value1,col | value2,...) :

کوچکترین مقدارلیست رابرمی گرداند نوع داده ای تمامی آرگومانهای دوم به بعد ، پیش ازمقایسه به نوع داده ای اولین آرگومان تبدیل می شوند.

```
SELECT  LEAST (1000,2000), LEAST(SAL,COMM)
FROM  EMP
WHERE DEPTNO =30;
```

LEAST(1000,2000)	LEAST(SAL,COMM)
1000	300
1000	500
1000	1250
1000	
1000	0
1000	

VSIZE (Col | value):

تعداد بایتهای مقدار ورودی به فانکشن را (ازنظرنمایش داخلی اوراکل) نشان می دهد.

```
SELECT        DEPTNO , VSIZE( DEPTNO ),
              VSIZE ( HIREDATE ), VSIZE( SAL )
FROM  EMP
WHERE DEPTNO =10;
```

DEPTNO	VSIZE(DEPTNO)	VSIZE(HIREDATE)	VSIZE(SAL)
10	2	7	3
10	2	7	2
10	2	7	2

بررسی مجدد توابع تودرتو :

درمثالهای زیرازتوابع تودرتو استفاده شده است

```
SELECT  ENAME,NVL(TO_CHAR(MGR), 'UNMANAGEABLE')
FROM   EMP
WHERE MGR  IS   NULL;
```

```
ENAME      NVL(TO_CHAR(MGR),'UNMANAGEABLE')
---------- ----------------------------------------
KING       UNMANAGEABLE
```

۱) ستون MGR توسط تابع TO_CHAR به یک مقدار کاراکتری تبدیل شده است.

۲) تابع NVL ستون MGR رادرصورت دارابودن مقدارنول به رشته کاراکتری UNMANAGEABLE تبدیل می کند.

برای نمایش تاریخ روزجمعه دوماه بعدازفرمت Day dd month YYYY باید دستور زیرراوارد کنید.

```
SELECT   SYSDATE,TO_CHAR(NEXT_DAY(ADD_MONTHS  (SYSDATE ,2),
                'FRIDAY' ), 'Day dd month YYYY')
FROM  DUAL ;
```

```
SYSDATE   TO_CHAR(NEXT_DAY(ADD_MONTHS
--------- ---------------------------
09-JUN-07 Friday    10 August    2007
```

۱) تابع ADD_MONTHS دوماه به ماه جاری (آگوست) می افزاید.

۲) تابع NEXT_DAY تاریخ روزجمعه دوماه بعدازتاریخ امروزرا پیدامی کند.

۳)تابع TO_CHAR مقدارتاریخ را به فرمت Day dd month YYYY تبدیل می کند.

فصل چهارم : توابع گروه (Group Functions)

این توابع برروی مجموعه ای ازسطرها اعمال می شوند بطورپیش فرض ، تمامی سطرها درحکم یک گروه واحدمی باشند ولی بااستفاده ازعبارت GROUP BY می توان این سطرها را به گروههای کوچکتری تقسیم نمود.

لیست توابع گروه درزیرآمده است تمامی این توابع بجز (*) COUNT ازمقادیرنول صرفنظر می کنند.

نام تابع	مقداربرگشتی
AVG([DISTINCT |ALL] n)	میانگینn راباصرفنظر کردن مقادیرنول بدست می اورد
MAX([DISTINCT|ALL] expr)	حداکثر مقدارعبارت را درسطرهای مورد پرس وجو نشان می دهد
MIN([DISTINCT |ALL]expr)	حداکثر مقدارعبارت را درسطرهای مورد پرس وجو نشان می دهد
STDDEV([DISTINCT |ALL] n)	انحراف معیارn رادرسطرهای مورد پرس وجو باصرفنظرکردن مقادیرنول بدست می آورد
SUM([DISTINCT | ALL] n)	مجموع n رادرسطرهای مورد پرس وجو باصرفنظرکردن مقادیرنول بدست می آورد
VARLANCE([DISTINCT|ALL]n	واریانس n را درسطرهای مورد پرس وجو باصرفنظرکردن مقادیرنول بدست می آورد.
COUNT([DISTINCT |ALL] n)	تعداد n را در سطرهای مورد پرس وجو ، با صرف نظر کردن مقادیر نول ،بدست می آورد

گزینه DISTINCT باعث می گردد که یک تابع گروه فقط برروی مقادیر غیرتکراری اعمال گردد گزینه ALL (حالت پیش فرض) تمامی مقادیررا دربرمی گیرد.

درحالت عادی برای آنکه توابع گروه بتوانند برروی مقدار ورودی نول هم اعمال شوند بهتراست ازدستورNVL استفاده نمایید بعنوان مثال برای محاسبه میانگین حقوق همه کارمندان بایدعبارت زیررادرج نمایید.

```
SELECT  AVG(SAL ) FROM  EMP ;
```

```
  AVG(SAL)
----------
2073.21429
```

درمثال فوق تمامی سطرها دریک گروه قرارگرفته اند.

یک تابع گروه می تواند برروی بخشی ازرکوردها نیزاعمال گردد اینکارباذکر عبارت WHERE انجام می شود بعنوان مثال برای پیداکردن حداقل حقوق افرادمنشی بایدعبارت زیرراوارد نمایید.

```
SELECT  MIN(SAL)
FROM  EMP
WHERE JOB='CLERK';
```

```
  MIN(SAL)
----------
       800
```

برای محاسبه تعداد کارمندان دپارتمان شماره ۲۰ از دستور زیر استفاده میکنیم :

```
SELECT COUNT(*)
FROM EMP
WHERE DEPTNO=20;
```

عبارت GROUP BY :

ازاین عبارت برای تقسیم سطرها دریک جدول به گروههای کوچکتراستفاده می گردد دراینحالت توابع گروه فقط برروی هریک ازگروههای مشخص اعمال خواهندشد بعنوان مثال برای محاسبه میانگین حقوق به ازای هریک ازشغلها باید دستورزیرراوارد نمایید.

```
SELECT  JOB, AVG (SAL )
FROM  EMP
GROUP  BY JOB;
```

```
JOB          AVG(SAL)
---------  ----------
ANALYST          3000
CLERK          1037.5
MANAGER    2758.33333
PRESIDENT        5000
SALESMAN         1400
```

بعنوان مثال دیگر ، برای نشان دادن میانگین حقوق به ازای همه شغلها بجز مدیران (managers) بایددستورزیرراوارد نمایید.

```
SELECT     JOB ,AVG (SAL )
FROM  EMP
WHERE JOB != 'MANAGER'
GROUP BY JOB;
```

```
JOB          AVG(SAL)
---------  ----------
ANALYST          3000
CLERK          1037.5
PRESIDENT        5000
SALESMAN         1400
```

ازعبارت GROUP BY میتوان برای ایجاد گروههایی دردرون گروههای دیگرنیزاستفاده نمود بعنوان مثال برای نمایش میانگین حقوق ماهیانه به ازای هرشغل درداخل یک دپارتمان ، باید دستورزیرراوارد نمایید.

```
SELECT  DEPTNO ,JOB , AVG(SAL )
FROM  EMP
GROUP BY DEPTNO , JOB ;
```

DEPTNO	JOB	AVG(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	3000
20	CLERK	950
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	1400

بهنگام استفاده ازتوابع گروه باید قواعد زیررا بیادداشته باشید.

- هیچگاه ازیک عبارت ویاستون مجزا بعنوان خروجی select استفاده نکنیدمگرآنکه ستون متناظربـاآن ، درعبـارت GROUP BY ظاهرگردد بعنوان مثال دستورزیر دارای وضوح وگویایی نیست.

```
SELECT MAX(SAL) FROM EMP GROUP BY JOB;
```

MAX(SAL)
3000
1300
2975
5000
1600

امامثال زیر ، دارای وضوح وگویایی بیشتری نسبت به مثال قبلی است.

```
SELECT  MAX(SAL), JOB
FROM  EMP
GROUP BY JOB ;
```

MAX(SAL)	JOB
3000	ANALYST
1300	CLERK
2975	MANAGER
5000	PRESIDENT
1600	SALESMAN

- اگردرخروجی یک SELECT ازتوابع گروه بعمراه ستون های مجزای دیگری استفاده شود حتمابایدعبارت GROUP BY رابه ازای آن ، ستونهای مجزا ذکرنماییم. بعنوان مثال ، دستورزیر نادرست بوده وموجب بروزخطا خواهدشد.

```
SELECT  DEPTNO ,MIN(SAL )  FROM  EMP;
```

برای اصلاح دستورفوق ، بایدعبارت GROUP BY رابه آن اضافه نماییم.

```
SELECT  DEPTNO ,MIN(SAL )  FROM  EMP
CROUP BY DEPTNO;
```

در مثال فوق ،ستون DEPTNO دیگر یک ستون مجزا نیست ، بلکه بعنوان نام یک گروه مطرح است .

اگر بیش از یک ستون مجزا در جلوی عبارت SELECT قـرار گیرنـد ،تمـامی آنهـا بایـد در عبـارت GROUP BY ، ذکـر شوندیعنی عمل گروه بندی باید به ازای همه آنها صورت پذیرد.

عبارت HAVING :

ازاین عبارت برای محدودکردن یک گروه استفاده می شود این عبارت باید بعدازدستور CROUP BY نوشته شودبعنوان مثال برای نشاندادن میانگین حقوق افراد همه دپارتمان هایی که دارای بیش از سه کارمند هستندبایددستور زیرراوارد نمود.

```
SELECT  DEPTNO ,AVG (SAL )
FROM  EMP
HAVING COUNT(*)>3
```

بعنوان مثالی دیگر ، برای نمایش آندسته از شغلهایی که حداکثرحقوق آنها بزرگتر ویامساوی 3000 دلاراست بایددستورزیرراوارد نمایید.

```
SELECT  JOB , MAX( SAL )
FROM  EMP
HAVING MAX(SAL ) >=3000
GROUP BY JOB;
```

JOB	MAX(SAL)
ANALYST	3000
PRESIDENT	5000

همانگونه که ملاحظه می کنیدعبارت HAVING رامی توان قبل ازGROUP BY قرارداد ، ولی توصیه می شود که چنین کاری راانجام نداده وازروش معمول استفاده نمایید.

ترتیب بیان عبارت های مختلف دریک دستور SELECT بشرح زیراست.

```
SELECT  column(s)
From table(s)
WHERE  row condition(s)
GROUP BY column(s)
HAVING group of rows condition(s)
ORDER BY column(s) ;
```

درزبان SQL ازعبارت where برای محدودکردن سطرها ازعبارت group by برای تعیین نحوه گروه بندی وازعبارت Having نیزبرای انتخاب گروههای خاص استفاده می شود.

فصل پنجم : استخراج داده ها ازبیش ازیک جدول

استفاده ازپیوندها(Joins) :

هرگاه بخواهیم داده هایی راازچندین جدول موجوددریک بانک اطلاعاتی بازیابی کنیم بایدازامکان پیوند (**Join**) استفاده نماییم. سطرهای یک جدول می توانند برمبنای برخی ازستونهای متناظر ، باسطرهایی ازجدول دیگر پیوند(**Join**) داشته باشند دونوع پیوند(**Join**) وجوددارد.

۱) پیوندبشرط تساوی (Equi- join)

۲) پیوندبشرط عدم تساوی (Non-equi-join)

۳) پیوند بیرونی (Equi-join)

پیوند بشرط تساوی (Equi- join)

برای تعیین اسامی کارمندانی که درهر آپارتمان قراردارند بایدمقادیرستون DEPTNO درجدولEMP با مقادیرستون DEPTNO درجدول DEPT مقایسه گردد ، رابطه بین جدولEMP وجدولDEPT ازنوع پیوند بشرط تساوی (Equi-join)است که درآن مقادیرستون DEPTNO درهردوجدول یکسان می باشد شرط پیوند رابایـد درجلوی عبارت WHERE مشخص نمود.

```
SELECT column(s) FROM EMP
WHERE join condition is…
```

برای پیوند دوجدول EMP و DEPT بایددستورزیرراوارد نمایید.

```
SELECT ENAME, JOB , DNAME FROM EMP , DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO ;
```

```
ENAME      JOB       DNAME
---------- --------- --------------
SMITH      CLERK     RESEARCH
ALLEN      SALESMAN  SALES
WARD       SALESMAN  SALES
JONES      MANAGER   RESEARCH
MARTIN     SALESMAN  SALES
BLAKE      MANAGER   SALES
CLARK      MANAGER   ACCOUNTING
SCOTT      ANALYST   RESEARCH
KING       PRESIDENT ACCOUNTING
TURNER     SALESMAN  SALES
ADAMS      CLERK     RESEARCH

ENAME      JOB       DNAME
---------- --------- --------------
JAMES      CLERK     SALES
FORD       ANALYST   RESEARCH
MILLER     CLERK     ACCOUNTING
```

همانگونه که ملاحظه می کنید درشرط پیوند ، پیش ازنام ستون ، جدول مربوطه نیزمشخص شده است زمانیکه نام ستونهای موجود درشرط پیوند ، یکسان باشند ، ذکرنام جدول ضروری است این ضرورت باید برای تمامی ستونهای همنام ازدوجدول ، که دردستور SELECT بیان می شوند ، اعمال گردد مثال زیر گویای این مطلب است.

```
SELECT  ENAME, DEPT. DEPTNO , DNAME
FROM  EMP , DEPT
WHERE EMP.DEPTNO = DEPT. DEPTNO
ORDER BY DEPT. DEPTNO ;
```

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
KING	10	ACCOUNTING
MILLER	10	ACCOUNTING
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
FORD	20	RESEARCH
SCOTT	20	RESEARCH
JONES	20	RESEARCH
ALLEN	30	SALES
BLAKE	30	SALES
MARTIN	30	SALES

ENAME	DEPTNO	DNAME
JAMES	30	SALES
TURNER	30	SALES
WARD	30	SALES

بیان مکرراسامی جداول دریک دستور SELECT ممکن است تاحدودی ملال آور باشد لذابهتراست بجای آنها ازاسامی موقت ومختصر (که به آن نام مستعار یا alias گفته می شود) استفاده گردد ، ازاین روش در مثال زیراستفاده شده است.

```
SELECT  E.ENAME, D.DEPTNO ,D.DNAME
FROM  EMP E,DEPT D
WHERE   E.DEPTNO = D.DEPTNO
ORDER BY D. DEPTNO ;
```

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
KING	10	ACCOUNTING
MILLER	10	ACCOUNTING
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
FORD	20	RESEARCH
SCOTT	20	RESEARCH
JONES	20	RESEARCH
ALLEN	30	SALES
BLAKE	30	SALES
MARTIN	30	SALES

حاصلضرب سطرها (PRODUCTS) :

اگرشرط پیوند نادرست بوده ویاحذف گردد ، نتیجه کاریک حاصلضرب دکارتی (product) ازسطرها بوده وتمامی ترکیبات ممکن سطرها نمایش داده خواهدشد.

پیوند بشرط عدم تساوی (Non- Equi-join)

رابطه بین جداول EMP و SALGRADE ازنوع پیوند بشرط عدم تساوی (non-equi-join) است که درآن هیچ ستونی ازEMP بطورمستقیم متناظرباستونی درجدول SALGRADE نیست. چنین رابطه ای با استفاده ازعلمگرهایی غیرازعلمگرتساوی(=) بدست می آیدبعنوان مثال ، برای ارزیابی رتبه یک کارمند حقوق وی باید درمحدوده دومقدار مشخص باشد که برای اینکار ازعملگر BETWEEN استفاده می گردد.

```
SELECT  E.ENAME,SAL ,S.GRADE
FROM  EMP E,SALGRADE S
WHERE  E.SAL BETWEEN S.LOSAL AND S.HISAL;
```

ENAME	SAL	GRADE
SMITH	800	1
ADAMS	1100	1
JAMES	950	1
WARD	1250	2
MARTIN	1250	2
MILLER	1300	2
ALLEN	1600	3
TURNER	1500	3
JONES	2975	4
BLAKE	2850	4
CLARK	2450	4

ENAME	SAL	GRADE
SCOTT	3000	4
FORD	3000	4
KING	5000	5

درعبارت فوق از عملگر های => و =< نیز می توانستیم استفاده کنیم. اما عملگر BETWEEN ساده تراز آنهاست.

برای پیوند (**Join**) سه جدول ، به دوشرط پیوند نیازداریم نیز برای پیوند چهارجدول ، حداقل به سه شرط پیوندنیاز است وبعنوان یک قاعده کلی برای پیوند n جدول ، حداقل به n-1 شرط نیاز داریم .

پیوند بیرونی (Outer-join) :

این نوع پیوند ، درحقیقت گسترش یافته پیوند معمولی (Equi- join) است چنین پیوندی تمامی سطرهای حاصل ازیک پیوندہ ساده رابرگردانده وعلاوه برآن سطرهایی ازجدول را که برای آنان ، بطورمستقیم ، متناظری درجدول دیگر وجودنداشته باشد ، برمی گرداند ، شکل کلی دستورپرس وجو برای چنین پیوندی بصورت زیراست.

```
SELECT  …
FROM  table1, table2
WHERE table1.column=table2 .column(+)
```

بعنوان مثال اگربخواهیم تمام کارمندان را ازجدول EMP و دپارتمان های آنها را ازجدول DEPT انتخاب کنیم. دراین صورت ممکن است کارمندانی وجودداشته باشند که دارای دپارتمانی درجدول DEPT نباشد ، سطرهای مربوطه به این کارمندان درصورت استفاده ازشکل ساده پیوند ، نشان داده خواهدشد برای مشاهده چنین سطرهایی دریک پرس وجو بایدحتما ازپیوند بیرونی (outer- join) استفاده می شود.

یک پیوند بیرونی شامل عبارت WHERE بهمراه یک شرط است که این شرط ، به یکی ازدوصورت زیرنوشته می شود.

Table1.column= table2.column(+)

Table1.column(+)=table2.column

اوراکل برای انجام چنین پیوندی ، تعدادی ستون اضافی ، به ازای جدولی که از عملگر (+) استفاده می کندایجادمی نماید تابتواند موجب برقراری پیوندبین سطرهای آن ، باتمامی سطرهای جدول دیگر شود.

بعنوان یک قاعده ، برای برقراری یک پیوند بیرونی ، بایدعلامت (+) رادرجلوی ستون موردنظر ازنام جدولی قراردهید که دارای سطرهای متناظرنیست.

درمثال زیر مجموع حقوق ماهیانه دپارتمان ها ، بدون استفاده ازیک پیوند بیرونی نشان داده شده است.

```
SELECT  DEPT.DEPTNO ,DNAME, SUM(SAL) MONTHLy_SALARY
FROM  EMP ,DEPT
WHERE   EMP.DEPTNO = DEPT.DEPTNO
GROUP BY DEPT.DEPTNO , DNAME
ORDER BY DEPT.DEPTNO ;
```

برای نمایش مجموع حقوق ماهیانه تمام دپارتمان ها حتی آنهایی که دارای کارمندنباشند بااستفاده ازیک پیوند بیرونی ، بایددستورزیرراوارد نمایید.

```
SELECT  DEPT.DEPTNO , DNAME,SUM(SAL )   MONTHLY_SALARY
FROM  EMP ,DEPT
WHERE   EMP.DEPTNO(+) = DEPT.DEPTNO
GROUP BY DEPT.DEPTNO , DNAME
ORDER BY DEPT. DEPTNO ;
```

همانگونه که ملاحظه می کنید سطرمربوطه به دپارتمان OPERATIONS باوجودنداشتن کارمندی درجدول EMP در خروجی ظاهر میشود.

فصل ششم : پرس وجو های فرعی (Subqueries) یاتودرتو

یک پرس وجوی فرعی (subquery) یک عبارت SELECT است که دردرون یک عبارت SELECT دیگرقرارمی گیرد وشکل کلی آن بصورت زیراست.

```
SELECT  column1, column2,…
FROM  EMP
WHERE column=
                (SELECT  column
                 FROM  table
                 WHERE condition )
```

بطورمعمول ، نخست ، دستور SELECT داخلی تراجراشده وسپس حاصل آن بعنوان شرط دستور SELECT اصلی مورداستفاده قرارمی گیرد. استفاده ازاین نوع پرس وجو زمانی مفیداست که بخواهیم سطرهای یک جدول رابراساس شرایط داده های موجوددرهمان جدول ، بدست آوریم.

بعنوان مثال ، برای نمایش اسامی کارمندانی که کمترین حقوق رادریافت می کنند بایدروال زیرااجرا نماییم.

(توجه داشته باشید که شرط کمترین حقوق ، یک کمیت نامشخص است) .

۱)حداقل حقوق رابیابیم.

```
SELECT  MIN(SAL ) FROM  EMP ;
```

۲)کارمندانی را پیدا کنیم که آن مقدار حقوق رادریافت می کنند

```
SELECT  ENAME, JOB , SAL
FROM  EMP
WHERE SAL=( 'کمترین حقوق')
```

می توان دو عبارت فوق را بصورت یک پرس وجوی تودرتو بایکدیگر تلفیق نمود.

```
SELECT  ENAME, JOB , SAL
FROM  EMP
WHERE SAL = (SELECT  MIN(SAL) FROM  EMP ) ;
```

هر عبارت SELECT رامی توان بعنوان یک بلوک پرس وجو (query block) درنظرگرفت . مثال فوق دارای دو بلوک پرس وجو است که یکی اصلی ودیگری فرعی یاداخلی است نخست بلوک داخلی تر اجرامی شود ومقدار800 رابعنوان نتیجه برمی گرداند. پس ازآن بلوک پرس وجوی اصلی اجرامی گردد که ازمقدار 800 بعنوان شرط جستجو ، استفاده می کنددرحقیقت پرس وجوی اصلی ، مشابه عبارت زیرخواهدبود.

```
SELECT  ENAME, JOB , SAL
FROM  EMP
WHERE SAL =800 ;
```

درمثال فوق عدد 800 یک مقدارواحداست براین مبنا پرس وجوهایی که حاصل آنها فقط یک مقدارواحدباشد ، پرس وجو هـای تک سطری نامیده می شوند درچنین حالتی درپرس وجوی اصلی ، می توان ازعلمگرهای تک سـطری ماننـد : = ، > ، < ، >= و... برای مقایسه بامقدارواحدبدست آمده استفاده نمود.

بعنوان مثالی دیگر ، برای پیداکردن کارمندانی که شغلشان مشابه شغل BLAKE است بایدعبارت زیرراوارد نمایید.

```
SELECT  ENAME, JOB
FROM  EMP
WHERE JOB =( SELECT  JOB
                FROM  EMP
                WHERE  ENAME = 'BLAKE') ;
```

پرس وجوی فرعی یا داخلی شغل Blake رابدست آورده وحاصل بعنوان شرط مقایسه ، درپرس وجـوی اصـلی ، مورداسـتفاده قرارمی گیرد.

حاصل یک پرس وجوی فرعـی ، مـی توانـد بـیش ازیـک سطرباشـدبعنوان مثـال درعبـارت زیرمـی خـواهیم کارمنـدانی را کـه درهرآپارتمان کمترین حقوق رادریافت می دارند. می باشند.

```
SELECT ENAME,SAL,DEPTNO
FROM  EMP
WHERE SAL IN ( SELECT  MIN( SAL )
                FROM  EMP
                GROUP BY DEPTNO ) ;
```

همانگونه که ملاحظه می کنید پرس وجوی فوق حاوی یک عبارت GROUP BY است این بدان معنی اسـت کـه نتیجـه آن می تواند بیش ازیک سطر باشد . بنابراین درچنین حالتهایی به عملگر های مقایسه ای چندسطری نیازداریم که درمثال فـوق از عملگر IN استفاده شده است.

درنتیجه بدست آمده ازپرس وجوی مذکور ، نام دپارتمان ها لزوما منطبق بااسامی کارمندانی که درآن کـارمی کننـد ، نیـست . بعلاوه ازآنجائیکه مافقط میزان حقوق را بررسی می کنیم ، لذا پرس وجوی فرعی یاداخلی ، فقط حداقل حقـوق را بـه ازای هـر دپارتمان ، تعیین کرده ولزوما هیچگونه انطباقی بین کارمندودپارتمان وی وجودندارد بنابراین برای برقراری ایـن انطبـاق بایـد پرس وجوی فوق رابصورت زیربازنویسی نمود :

```
SELECT  ENAME ,SAL, DEPTNO
FROM  EMP
WHERE (SAL , DEPTNO ) IN ( SELECT   MIN(SAL),DEPTNO
                            FROM  EMP
                            GROUP BY DEPTNO );
```

(همانگونه که ملاحظه میکنید ، حاصل دوپرس وجو تصادفا مشابه یکدیگر است ولی این بمعنی صحت عملکرد پرس وجوی اول ، تحت همه شرایط نیست) .

درپرس وجوی اخیر ، ازدوستون درمقابل عبارت WHERE بعنوان شرایط جستجو ، استفاده شده است که این دوستون ، با کاماازهم جدا شده اندتوجه داشته باشید که درپرس وجوی فرعی ، ستون های مربوطه ، باید بهمان ترتیبی که درعبارت WHERE از پرس وجوی اصلی آمده اند قرارگیرند. علاوه برآن ، نوع داده ای ستونهای متناظر نیزباید یکسان باشد.

اگرحاصل یک پرس وجوی فرعی ، بیش ازیک سطرباشد ، ولی از عملگر تک سطری جهت انجام مقایسه استفاده شود ، دراینحالت SQL*Plus پیغام خطایی رانشان خواهد داد.

```
SELECT  ENAME ,SAL , DEPTNO
FROM  EMP
WHERE SAL=(SELECT  MIN(SAL )
                FROM  EMP
               GROUP BY DEPTNO ) ;
ERROR : ORA-1427 : single-row subquery returns more than one row
```

اگرپرس وجوی فرعی سطری رابرنگرداند بازهم پیغام خطای فوق را دریافت خواهیدنمود.

```
SELECT  ENAME , JOB
FROM  EMP
WHERE JOB =( SELECT  JOB
                FROM  EMP
                WHERE ENAME ='SMYTHE') ;
ERROR : ORA-1427 single-row sub query returns more than one row
no records selected
```

عملگر های ANY و ALL :

درپرس وجوی های فرعی که حاصل آنها بیش ازیک سطراست می توان از این عملگر ها استفاده نمود. این عملگر ها درعبارتهای WHERE وبهمراه عملگر های منطقی =,!=,<,>, >= و => بکارمی روند.

عملگر ANY یک مقدار را باهریک ازمقادیر حاصل ازیک پرس وجوی فرعی مقایسه می کند بعنوان مثال ، برای نمایش کارمندانی که میزان حقوق آنها بیشتراز حداقل حقوق دریافتی در دپارتمان 30 است ، بایدعبارت زیررا وارد نماییم.

```
SELECT ENAME,SAL,JOB,DEPTNO
FROM  EMP WHERE SAL > ANY( SELECT  DISTINCT SAL
                        FROM  EMP
                       WHERE DEPTNO =30)
ORDER BY SAL DESC;
```

ENAME	SAL	JOB	DEPTNO
KING	5000	PRESIDENT	10
SCOTT	3000	ANALYST	20
FORD	3000	ANALYST	20
JONES	2975	MANAGER	20
BLAKE	2850	MANAGER	30
CLARK	2450	MANAGER	10
ALLEN	1600	SALESMAN	30
TURNER	1500	SALESMAN	30
MILLER	1300	CLERK	10
WARD	1250	SALESMAN	30
MARTIN	1250	SALESMAN	30
ADAMS	1100	CLERK	20

12 rows selected.

در دپارتمان 30 حداقل میزان حقوق ، ۹۵۰ دلار و مربوط به Jmaes است پرس وجوی اصلی ، اسامی کارمندانی راکه حقوقشان بیشتراز این مقداراست بدست می آورد. **'ANY >'** بمعنی **' بیشترازمقدار حداقل '** بوده و' ANY=' نیزمعادل عملگر **IN** است بهنگام بکاربردن عملگر ANY اغلب بایدازکلمه DISTINCT جهت ممانعت ازانتخاب سطرهای تکراری استفاده نمود.

عملگر ALL نیزیک مقدار راباهمه مقادیر حاصل ازیک پرس وجوی فرعی ، مقایسه می کند ولی بلحاظ مفهومی باعملگر ANY متفاوت است بعنوان مثال برای نمایش کارمندانی که میزان حقوق آنها بیشتر از حقوق دریافتی تمامی کارمندان دپارتمان 30 است ، بایدعبارت زیرراوارد نماییم.

```
SELECT  ENAME , SAL , JOB , DEPTNO  FROM EMP
WHERE  SAL > ALL(SELECT  DISTINCT SAL
                 FROM  EMP
                 WHERE DEPTNO =30):
```

در دپارتمان 30 بیشترین میزان حقوق ، 2850 دلاربوده که متعلق به Blake است . پرس جوی مذکور آندسته ازکارمندانی را که حقوقشان بیشترازاین مبلغ است نشان میدهد.' **All >**' بمعنی **'بیشترازمقدار حداکثر'** می باشد.

می توان از عملگر NOT بهمراه عملگر های ALL, ANY, IN استفاده نمود .

استفاده از عبارت HAVING درپرس وجوی تودرتو :

همانگونه که میدانید از عبارت WHERE برای اعمال محدودیت برروی تک تک سطرها ، وازعبارت HAVING نیزبرای اعمال محدودیت برروی گروههایی ازسطرها استفاده می شود بعنوان مثال برای نمایش دپارتمان هایی که دارای میانگین حقوقی کمتراز دپارتمان 30 هستند ، بایدعبارت زیرراوارد نمایید.

```
SELECT  DEPTNO , AVG (SAL )
FROM  EMP
HAVING  AVG (SAL )> ( SELECT  AVG (SAL)
                      FROM  EMP
                      WHERE DEPTNO =30)
GROUP BY DEPTNO ;
```

بعنوان مثالی دیگر برای نشان دادن شغلی که دارای بیشترین میانگین حقوقی است بایدعبارت زیرراوارد نمایید.

```
SELECT  JOB , AVG( SAL )
FROM  EMP
HAVING AVG (SAL) =( SELECT  MAX(AVG (SAL ))
                    FROM  EMP
                    GROUP BY JOB )
GROUP BY JOB;
```

JOB	AVG(SAL)
PRESIDENT	5000

درعبارت فوق ، نخست پرس وجوی فرعی یاداخلی ، میانگین حقوق را به ازای هرشغل ، بدست آورده وبااستفاده ازتابع MAX حداکثراین میانگین ها را مشخص می سازد. مقدار حاصله (5000) نیز ، بعنوان شرط عبارت HAVING درپرس وجوی اصلی، مورداستفاده قرارمی گیرد.

دریک پرس وجوی فرعی ، نمی توان ازعبارت ORDEY BY استفاده نمود . قاعده این است که دریک دستور SELECT فقط یک عبارت ORDER BY می توانیم داشته باشیم وآن نیز بایدآخرین عبارت درفرمان SELECT باشد.

پرس وجوهای فرعی خودمی توانند بصورت تودرتو باشندبعنوان مثال ، برای نمایش نام ، شغل ، وتاریخ استخدام کارمندانی که حقوقشان بیشترازحداکثر میزان حقوقشان در دپارتمان SALES است بایدعبارت زیرراوارد نمایید.

```
SELECT  ENAME , JOB , HIREDATE , SAL
FROM  EMP
WHERE SAL > ( SELECT  MAX( SAL )
                FROM  EMP
                WHERE DEPTNO =( SELECT  DEPTNO
                                FROM  DEPT
                                WHERE DNAME = 'SALES'));
```

ENAME	JOB	HIREDATE	SAL
JONES	MANAGER	02-APR-81	2975
SCOTT	ANALYST	19-APR-87	3000
KING	PRESIDENT	17-NOV-81	5000
FORD	ANALYST	03-DEC-81	3000

یک پرس وجوی اصلی می تواند حداکثر دارای 16 پرس وجوی فرعی درسطح یک باشد. نیزهرسطح می تواند تا 255 سطح بصورت تودرتو ، عمق داشته باشد.

سطح 1 سطح 2 سطح 3 ... سطح 255

```
-------------------------------------------------
SELECT
FROM
WHERE ( SELECT
          FROM
          WHERE (SELECT
                    FROM
                    WHERE ))
SELECT
FROM
WHERE
.
.
```

حداکثر تا ۱۶ سطح

پرس وجوهای موجوددریک سطح ، میتوانند با عملگر هایی نظیر union ازهم جدا شونددراین حالت استفاده از حداکثر16 پرس وجومجاز می باشد.

قواعد کلی درباره پرس وجو های موجوددرتودرتو :

- پرس وجوی فرعی یاداخلی ، بایددرداخل پرانتزبوده ودرسمت راست عبارت شرط، قرارگیرد.
- پرس وجوی فرعی ، نباید حاوی عبارت ORDER BY باشد.
- ستون هایی که درمقابل عبارت SELECT از پرس وجوی فرعی قرارمی گیرند ، بایدبهمان ترتیبی باشند که در عبارت شرط پرس وجوی اصلی بکارمی روند . نوع داده ای وتعداد ستونها نیزبایدیکسان باشد.
- دریک پرس وجوی فرعی می توان از عملگر های مجموعه استفاده نمود.
- پرس وجوی فرعی بغیراز پرس وجو های همبسته(correlated) همیشه باشروع ازپایین ترین عمق ، اجرا می شوند.
- میتوان از عملگر های منطقی و SQL ، مشابهANY و ALL ، استفاده نمود.

پرس وجوی فرعی میتوانند :

- ازعبارت GROUP BY ویاتوابع گروه استفاده کنند.
- یک یا جند ستون برگردانند .
- بصورت AND یاOR همراه با پرس وجو پرس وجوی اصلی بکاررروند.
- بین جداول پیوند ایجادکنند .
- داده ها را ازجدولی غیرازجدول موجود در پرس وجوی اصلی بازیابی کنند.
- درعبارتهای CREATE TABLE, INSERT,DELETE, UPDATE, SELECT ظاهرشوند .
- بایک پرس وجوی اصلی، همبسته (correlate) شوند.

پرس وجوهای هم بسته (correlated subqueries) :

این پرس وجو ، ازنوع تودرتو (nested subqury) است که به ازای هرسطر از پرس وجوی اصلی یکباراجراشده و درآن ازستونی که در پرس وجوی اصلی بکاررفته ، استفاده می گردد و همین امرباعث می شود که نحوه پردازش اینگونه پرس وجو ها متفاوت از پرس وجو های تودرتوی معمولی باشد یک پرس وجو همبسته بابکاربردن ستونی از پرس وجوی اصلی دربخش شرایط مربوط به پرس وجوی فرعی ، مشخص میگردد دریک پرس وجو تودرتوی معمولی ، نخست پرس وجوی داخلی تر یکباراجرا شده ونتیجه بدست آمده درپرس وجوی اصلی مورداستفاده قرارمی گیرد امادر پرس وجوهای همبسته ، قضیه بدین شکل نیست دراینجا پرس وجوی داخلی تر به ازای هرسطراز پرس وجوی اصلی ، یکبار اجرا می شود مراحل اجرای یک پرس وجو همبسته عبارتنداز :

۱)هرسطر از پرس وجوی اصلی بدست می آید.

۲)به ازای هریک از سطرهای واکنشی شده ، پرس وجوی فرعی یکبار اجرامی گردد.

۳)مقادیرحاصل از پرس وجوی فرعی جهت تعیین واحدشرایط بودن سطرهای بدست آمده ازمرحله اول ، مورداستفاده قرار می گیرند.

۴)روال فوق بابررسی تمامی سطرهای موجود درپرس وجوی اصلی ادامه می یابد.

بعنوان مثال برای پیداکردن کارمندانی که میزان حقوقشان بیشترازمیانگین حقوق دپارتمان آنهاست ،باید عبارت زیررا وارد نمایید.

```
SELECT  EMPNO , ENAME , SAL , DEPTNO
 FROM  EMP E
WHERE SAL > ( SELECT  AVG( SAL )
                FROM  EMP
                WHERE DEPTNO = E.DEPTNO )
ORDER BY DEPTNO ;
```

EMPNO	ENAME	SAL	DEPTNO
7839	KING	5000	10
7566	JONES	2975	20
7788	SCOTT	3000	20
7902	FORD	3000	20
7499	ALLEN	1600	30
7698	BLAKE	2850	30

6 rows selected.

توجه داشته باشید که استفاده ازنام مستعار (alias) برای برخی ازستونها ، بجهت عدم ایجاد ابهام بسیارضروری است برای تحلیل مثال فوق آن را به دوبخش تقسیم می کنیم:

درپرس وجوی اصلی :

۱) اولین رکورد (اسمیت ،دپارتمان 20 ومیزان حقوق 800 دلار) انتخاب می گردد.

۲) جدول EMP دارای نام مستعار E بوده که ازآن درستون deptno از پرس وجوی فرعی استفاده شده است

۳) عبارت WHERE عدد800 را بامقدار بدست آمده از پرس وجوی فرعی مقایسه می کند .

درپرس وجوی فرعی :

۴) میانگین حقوق ، به ازای دپارتمان مربوط به کارمندمشخص شده محاسبه می گردد.

۵) ستون E.DEPTNO در پرس وجوی فرعی ازطریق ستون DEPTNO در پرس وجوی اصلی ، مقدار میگردد.

۶) میانگین حقوق به ازای دپارتمان اسمیت (20) ، معادل 2175 دلاراست .

۷) سطربدست آمده در پرس وجوی اصلی ، باشرایط بدست آمده دراینحالت سازگار نبوده ولذا نادیده گرفته می شود.

۸) روال فوق برای سایررکوردهای حاصل از پرس وجوی اصلی تکرارمی گردد

یک دستور Update نیزمی تواند حاوی پرس وجو های همبسته باشد :

```
UPDATE EMP E
SET ( SAL,COMM)= ( SELECT  AVG(SAL ) *1.1,AVG (COMM)
                    FROM  EMP
                    WHERE DEPTNO= E.DEPTNO),
HIRDATE = '11-JUN-85' ;
```

عملگر ها (Operators) :

در پرس وجو های تودرتو استفاده ازتمامی عملگر ها نظیرANY و ALL معتبرخواهدبود .علاوه برهمه آنها می توان از عملگر EXISTS نیزاستفاده نموداین عملگر غالبا در پرس وجو های همبسته ، بکارمی رود کاراین عملگر بررسی وجود ویاعدم وجود برخی ازمقادیراست . اگرمقدارمربوطه وجودداشته باشد حاصل این عملگر TRUE بوده ودرصورت عدم وجودیک مقدار ، حاصل آن FALSE خواهدبود. بعنوان مثال برای پیداکردن کارمندانی که خود ، یک مدیربوده ودارای افراد زیردست باشند ، بایدعبارت زیرراوارد نمایید :

```
SELECT  EMPNO, ENAME ,JOB , DEPTNO
FROM  EMP E
WHERE EXISTS ( SELECT  EMPNO
                FROM  EMP
                WHERE EMP.MGR= E.EMPNO)
ORDER BY EMPNO;
```

EMPNO	ENAME	JOB	DEPTNO
7566	JONES	MANAGER	20
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7902	FORD	ANALYST	20

6 rows selected.

بعنوان مثالی دیگر ، برای یافتن همه کارمندانی که دپارتمان آنها درجدول DEPT وجودندارد بایدعبارت زیرراواردکنید.

```
SELECT  EMPNO , ENAME, DEPTNO
FROM  EMP
WHERE NOT EXISTS( SELECT  EMPNO
                    FROM  EMP
                    WHERE DEPT.DEPTNO = EMP.DEPTNO ):
```

قابل ذکراست عبارت فوق ، رکوردی را برنمی گرداند.

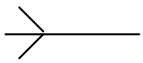
فصل هفتم : مبانی طراحی یک بانک اطلاعاتی رابطه ای

هرسیستم اطلاعاتی ، بابخشی از جهان واقعی سروکاردارد طراحی یک سیستم بانک اطلاعاتی عبارت است از تبدیل مدلی ازدنیای واقعی به صورت یک نرم افزار کاربردی . هرسیستم بانک اطلاعاتی دارای مراحل تعیین استراتژی ، تحلیل ، طراحی وایجاد می باشدخروجی هرمرحله ، بعنوان ورودی مرحله بعد ، بکارخواهدرفت . در این مبحث ، نحوه ایجاد جدوال بانک اطلاعاتی بهمراه مبانی طراحی سیستم موردبررسی قرارمی گیرد.

هدف طراح سیستم ، ایجاد یک طرح فیزیکی برای پیاده سازی سیستم براساس نتایج بدست آمده از مرحله تحلیل سیستم می باشد . درمرحله تحلیل نمودار رابطه موجودیتها یا ERD (Entity Relationship Diagram) تولید می گردد . این مدل یک ابزار ارتباطی بوده که درآن ، رابطه بین موجودیتها نظیرشخص ، شی فیزیکی ، فعالیت و... نشان داده می شود (موجودیت آن چیزی است که درارتباط با آن داده هایی نگهداری می شود) و به آن مدل مفهومی (Conceptual Model) نیزگفته می شود.

خطوط وعلائم بکاررفته دریک نمودارERD عبارتنداز :

رابطه یک بیک ———

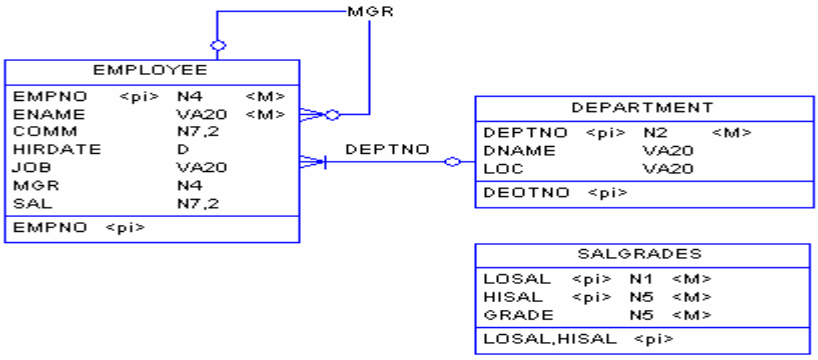
رابطه یک به چند 

(# یا pi) : بخشی از معرفه یونیک O : فیلداختیاری (* یا M) : معرف فیلداجباری

مفهوم نمودار رابطه بین جدول کارمندان (Employee) ودپارتمانها (Deparment) چنین است :

- هر دپارتمان میتواند شامل یک یاچند کارمند باشد .
- هرکارمند باید فقط وفقط ، عضو یک دپارتمان باشد.
- هرمدیر، خودیک کارمند بوده ومی تواند چندین کارمندزیردست نیزداشته باشد.

درفازطراحی ، مدل مفهومی ، به یک مدل منطقی تبدیل شده و توسط یک نمودار جداول (Table Diagram) نشان داده می شود . درنمودار جداول جدولهای موردنیاز تشریح می شوند. درچنین نموداری، هرجدول ، توسط یک مستطیل نشان داده شده وارتباط بین آنها نیزتوسط خطوطی مشخص می گردد ستونی (column) که بوجود آورنده ارتباط است برروی این خطوط ظاهرمی گردد مشخصه ستونهای یک جدول نیز درداخل مستطیل نشان داده می شود.



درنمودار جدولی فوق ، N بمعنی عدد(Number) ، VA بمعنی کاراکتر (Variable Character) ، D بمعنی تاریخ (Data) وf بمعنی کلیدخارجی (Foreign key) می باشد.

علامت pi بمعنی آن است کـه سـتون مربوطـه ، بخـشی ازیـک کلیداصـلی (primary key) اسـت کلیداصـلی یـک سـتون ویاترکیبی ازستونهایی است که مقادیرآنها منحصربفرد (unique)بودن سطرها را دریک جدول ، مشخص مـی سـازد ، بعنـوان مثال ، ستون EMPNO درجدولEMP سطرها را ازهم متمایز می کندونباید دارای مقدارتکراری (duplicate) باشـد دریـک جدول ، ستونی که کلیداصلی است حتماباید دارای مقدار غیرنول باشد اگرکلیداصلی ، شامل بـیش ازیـک سـتون باشـد درایـن صورت هیچک ازآن ستونها نیزنمیتوانند دارای مقدارنول باشند.

علامتf درنمودارجدولی ، مشخص می سازد که ستون مربوطه به یک کلید خارجی (foreign key) اسـت یـک کلیـدخارجی ، ستون ویاستونهایی است که محتویات آن ، ازطریق مقادیرکلیداصلی درجدول دیگر بدست می آیـد ، کلیـدهای خـارجی امکـان پیوند بین جداول رافراهم می سازند. ستون DEPTNO درجدول EMPیک ، کلیدخارجی است ودارای مشخـصه f مـی باشـد مقدار این ستون بایدمتناظربایکی ازمقـادیر موجـود درسـتون DEPTNO ازجـدول DEPT (کـه کلیداولیـه بـوده و دارای مشخصه pi است) باشد.

دراوراکل ، کلیدهای اصلی دارای شاخص های منحصر بفرد (unique index) می باشند تامانع ازدرج مقادیرتکراری شوند.

اگرستونی دارای مشخصه M باشدمعنی آن است که این ستون ، حتمابایدمقداربگیرد.

پس ازتعیین نمودارهای جدولی ، بایدآنهارا به دستورات ایجادجدول (CREATE TABLE) تبدیل نماییم.

زبان تعریف داده ها ودیکشنری داده ها (Date Definition Language & Data Dictionary)

- جدول درهرزمانی میتواند ایجاد شوند حتی اگر کاربران درحال استفاده ازبانک اطلاعاتی باشند.
- طول داده ها متغیرمی باشد واین بدان معنی است که داده ها فقط به اندازه طولشان ذخیـره مـی شـوند وبلاتکهـای ابتداو انتها ذخیره نخواهندشد.
- نیازی به تعیین اندازه جدول نیست این امر ، بصورت کلی ، باتعیین میزان فضای لازم برای بانـک اطلاعـاتی تعیـین می گردد.
- مقادیرنول فضایی را اشغال نمی کنند.
- ساختارجدول رامی توان بهنگام استفاده On-Line ازآنها اصلاح نمود.

مراحل ایجاد یک جدول :

۱) نام جدول باید بایکی ازحروفA-Z ویا a-z شروع شود.

۲) این نام می تواندشامل حروف ، اعداد و کاراکتر زیرخط (underscore) باشد.

۳) حروف بزرگ وکوچک درتعیین نام جدول ، یکـسانند بعنـوان مثـال اسـامی eMp,emp,EMP معـرف نـام یـک جـدول می باشند.

۴) نام یک جدول می تواندحداکثر دارای30 کاراکتر باشد.

۵) نام یک جدول نباید مشابه نام جدول ویانمایه دیگری درهمان بانک اطلاعاتی باشد.

مثالهایی از اسامی مجاز وغیرمجاز برای جداول درزیرآمده است :

EMP35 : مجاز است

85EMP: غیرمجاز است زیرابایک حرف ، شروع نشده است.

FIXED_ASSETS مجازاست.

FIXED ASSETS : غیرمجازاست زیرابین حروف آن فاصله وجوددارد.

UPDATE : غیرمجاز است زیرایک کلمه رزرو شده (reserved word) است.

TABLE1 : مجازاست ، ولی بهتراز چنین نامهایی استفاده نشود و بجای آن اسامی گویاتری بکاررود.

نوع داده ای ستونهای یک جدول (Column Types) :

برای ایجاد یک جدول ، بایدنوع داده ای (Data Type) ستون های آن رامشخص نمایید. جدول زیر مهمترین انواع داده ای را نشان می دهد درجلوی هرنوع داده ای ، میتوان طول ستون موردنظررا توسط اعدادی در داخل پرانتز مشخص نمود این عدد معرف حداکثر طول مجازی است که مقدارمتناظر بااین ستون ، می تواندداشته باشد ستونهای ازنوع CHAR حتمابایـد دارای مشخصه طول باشند ولی برای ستون های ازنوع NUMBER ، این امر ، اجباری نیست.

نوع داده ای	محتویات
CHAR(w)	حاوی مقادیر کاراکتری ، شامل حروف بزرگ وکوچک و کاراکترهای خاص (نظیر +,-,%, SELECT,&) است طول مقادیراین ستون نمی تواند بیشتراز 240 کاراکتر باشد.
NUMBER	حاوی مقادیر عددی شامل ارقام 9-0 وعلامتهای +,- نقطه اعشار است حداکثرطول مقادیراین ستون (بدون درنظرگرفتن علامت ونقطه اعشار) 40 رقم است
NUMBER(w)	مشابه نوع داده ای قبلی است که درآن حداکثر طول باw مشخص میگردد مقدارw نمیتواند بیشتراز40 باشد.
NUMBER(W,d)	مشابه نوع داده ای قبلی است که درآن حداکثر طول ستون باw و حداکثر تعداد ارقام اعشاری آن با d مشخص می گردد .
DATE	مقادیرحاوی تاریخ از1 ژانویه زاویه 4712 قبل ازمیلاد تا31 دسامبر4712 پس ازمیلاد
LONG	مشابه نوع داده ای CHAR بوده ولی می تواند دارای طول حداکثر65535 کاراکتر باشد دریک جدول بیش ازیک ستون ازنوع LONG نمی توان تعریف نمود.

مثالهایی ازتعریف نوع داده ای درزیرآمده است :

CHAR(12) : ستون متناظر میتواند دارای حداکثر 12 کاراکتر باشد.

NUMBER(4) : ستون متناظر میتوانددارای حداکثر 4رقم باشد.

NUMBER(8,3) : ستون متناظر میتواند دارای حداکثر۸رقم باشد(5 رقم صحیح و 3رقم اعشار)

برای ایجاد یک جدول ، دربانک اطلاعاتی اوراکل ، باید ازدستورCreate Table استفاده نمایید.

```
CREATE TABLE TABLE_NAME
(column_name type(size) [ NULL/NOT NULL],
 column_name type(size) [NULL/NOT NULL] ,
…) ;
CREATE TABLE DEPT
( DEPTNO   NUMBER(2) NOT NULL ,
DNAME  CHAR(12) ,
LOC CHAR(12)) ;
```

گزینه NULL بمعنی آن است که ستون متناظر میتواند دارای مقدار نول باشد . یک ستون بهنگام تعریف به طور پیش فرض دارای گزینه NULL بوده ولذا نیازی به ذکر آن نیست.

گزینه NOT NULL بمعنی آن است که ستون متناظر حتمابایددارای یک مقدارغیرنول باشد درصورتیکه بخواهید سطری رابدون دادن مقدار به ستون های NOT NULL درج کنید. پیغام خطایی توسط SQL*Plus ظاهرمی شود.

مثالی ازایجاد یک جدول (بنام EMP) درزیرآمده است :

```
CREATE TABLE EMP (
EMPNO  NUMBER(4)  NOT NULL ,
ENAME CHAR(10) ,
JOB CHAR(10)  ,
MGR  NUMBER(4) ,
HIREDATE DATE  ,
SAL  NUMBER  ,
COMM  NUMBER(7,2)  ,
DEPTNO  NUMBER(2)  NOT NULL ) ;
```

بااجرای دستورفوق ، پیغام Table created نشان داده می شود.

برای مشاهده نحوه تعریف جدول ، باید ازدستورزیراستفاده نمایید :

```
DESCRIBE  EMP ;     OR     DESC EMP ;
```

NAME	NULL؟	EMP
EMPNO	NOT NULL	NUMBER(۴)
ENAME		CHAR(۱۰)
JOB		CHAR(۱۰)
MGR		NUMBER(۴)
HIREDATE		DATE
SAL		NUMBER(۷،۲)
COMM		NUMBER(۷،۲)
DEPTNO	NOT NULL	NUMBER(۲)

درمثال مذکور EMPNO بصورت NOT NULL تعریف شده وستونی است که معرف منحصر بفردبودن هرسطر ازجدول است.

بعنوان مثالی دیگر ، برای ایجادجدولی که باید حاوی ستونها نام ، حقوق ورتبه کارمندان باشد بایددستور زیررا واردنمایید

```
CREATE TABLE EMP_SALS(NAME , SALARY,GRADE)   AS
SELECT  ENAME , SAL GRADE
FROM  EMP , SALGRADE
WHERE EMP.SAL BETWEEN LOSAL AND HISAL  ;
Table created
```

DESC EMP_SALS;

NAME	NULL؟	EMP
NAME		CAHR(۱۰)
SALARY		NUMBER(۷،۲)
GRADE		NUMBER

نحوه اصلاح ساختاریک جدول

بااستفاده ازدستورALTER TABLE میتوان نحوه تعریف یک جدول راتغییرداد :

۱) برای افزودن یک ستون جدید به جدول بایدازگزینه ADD استفاده نمود.

ALTER TABLE name
ADD (column type)

بعنوان مثال ، برای افزودن ستونی برای نگهداری نام همسرهریک ازکارمنـدان درجـدول EMP بایدازدسـتور زیراسـتفاده نمایید :

ALTER TABL EMP
ADD (SPOUSE_NAME CHAR(10) ;
Table altered.

نحوه ایجاد یک جدول بااستفاده ازجدول دیگر

میتوان یک جدول را بااستفاده ازجدول دیگر تعریف نمود که دراینحالت سطرهای آن نیز ، توسط سطرهای جدول مبدا پرمی شوند.

CREATE TABLE table_name
[(column_name [NULL/NOT NULL],…)]
AS SELECT statement

× جدول جدید باستونهای مشخص شده ایجادشده وعمل درج درسطرهای آن ازطریق دستور SELECT انجام می گردد.

× اگرستونهای جدول جدید کاملا مشابه ستون های جدول مبدا باشند نیازی به ذکراسامی ستونها دردستور فوق نخواهدبود.

× درصورت مشخص کردن ستونها ، تعداد آنها باید مساوی تعداد ستونهای متناظر دردستور SELECT باشد.

بعنوان مثال ، برای ایجاد جدولDEPT30 که باید حاوی ستونهای شماره ، نام ، شغل وحقوق کارمندان دپارتمـان 30 باشـد ، بایددستور زیرراوارد نمایید.

CREATE TABLE DEPT30 AS
SELECT EMPNO , ENAME, JOB , SAL
FROM EMP
WHERE DEPTNO =30 ;

برای مشاهده نحوه تعریف جدول ، بایدازدستور زیراستفاده نمایید.

DESC DEPT30 ;

NAME	NULL؟	EMP
EMPNO	NOT NULL	**NUMBER(4)**
ENMAE		**CHAR(10)**
JOB		**CHAR(9)**
MGR		**NUMBER(4)**
HIREDATE		DATE
SAL		**NUMBER(7,2)**
COMM		**NUMBER(7,2)**
DEPTNO		**NUMBER(2)**
NAME_SPOUSE		**CHAR(10)**

2) برای تغییریک ستون بایدازگزینه MODIFY استفاده نمود :

ALTER TABLE name

MODIFY(column type [NULL)]

بعنوان مثال برای تغییرطول ستون ENAME درجدول EMP به مقدار 25 کاراکتر باید ازدستورزیراستفاده نمایید :

ALTER TABLE EMP

MODIFY (ENAME CHAR(25)) ;

Table altered

DESC EMP;

NAME	NULL؟	EMP
EMPNO	NOT NULL	**NUMBER(4)**
ENMAE		**CHAR(25)**
JOB		**CHAR(9)**
MGR		**NUMBER(4)**
HIREDATE		DATE
SAL		**NUMBER(7,2)**
COMM		**NUMBER(7,2)**
DEPTNO		**NUMBER(2)**
NAME_SPOUSE		**CHAR(10)**

سه نوع تغییررا نمی توان برروی جداول موجود ، اعمال نمود.

۱) نمی تواند ستونی را که حاوی مقادیرنول است بصورت NOT NULL تغییردهید.

۳) نمیتوانید اندازه یک ستون راکاهش داده ویانوع داده ای آن راتغییردهید مگرآنکه ستون مربوطه شامل هیچ داده ای نباشد.

نحوه حذف یک جدول

بااستفاده از دستورDROP TABLE میتوان یک جدول اوراکلی راحذف نمود.

```
DROP TABLE EMP ;
```

بهنگام حدف یک جدول باید به نکات زیرتوجه نمایید :

- تمامی داده های موجوددرآن جدول ، بهمراه شاخص (index) های مربوطه ازبین خواهندرفت.
- نمایه ها (VIEWS) ومترادفهای (SYNONYMS) متناظرباجدول باقی مانده ولی غیرمعتبر خواهندبود.
- تمامی تراکنشهای معوق (Pending Transaetions) ، ثبت نهایی (commit) میگردند.
- فقط ایجادکننده یک جدول ، میتواند آن راحذف نماید.

نحوه تغییر نام یک جدول

بااستفاده از دستورRENAME میتوان نام یک جدول راتغییرداد.

```
RENAME old TO new ;
```

بعنوان مثال برای تغییرنام جدول EMP ، به EMPLOYEE بایددستور زیرراوارد نمایید :

```
RENAME EMP TO EMPLOYEE ;
```

دیکشنری داده ها دراوراکل

دیکشنری داده ها (Data Dictionity) یکی ازمهمترین بخشهای سیستم مدیریت بانک اطلاعاتی اوراکل است وشامل جداول ونمایه هایی است که حاوی اطلاعات باارزشی درباره کاربران (USERS)،حق دستیابی کاربران (USER PRIVILEGES)، جدول ها (TABLES)، شاخص ها (INDEXES) ونحوه بازرسی (AUDITING) می باشد.

یکسری داده ها بهنگام نصب اوراکل ، بطورخودکار ایجادشه ومالک آنها نیزکاربر SYS است این جداول بطوراتوماتیک بوسیله سیستم مدیریت بانک اطلاعاتی اوراکل ، نگهداری واصلاح شده و غالبا توسط کاربران عادی ، مورددستیابی قرارنمی گیرند زیرا اطلاعات موجود درآن ها بسادگی ، قابل فهم نیستند.

نمایه ها (views) دیکشنری داده ها ، شامل اطلاعاتی است که فهم آن برای یک کاربرعادی ساده تر بوده ودارای کاربردهای بیشتری نیزمی باشند مالک این نمایه ها کاربر system است. این نمایه ها بطور اتوماتیک بهنگام نصب اوراکل ، ایجاد نمی شوند وبرای ایجاد آنها بایدفایل CATALOG.ORA توسط کاربر SYSTEM اجراگردد.

نگهداری واصلاح جداول ونمایه های سیستمی درفاصله های زمانی مشخص توسط سیستم مدیریت بانک اطلاعاتی اوراکل انجام می شود بعنوان مثال اگریک اجازه دستیابی (grant) به کاربرجدیدی داده شود در اینحالت یک سطرجدید به جدول SYSUSERAUTH افزوده میگردد اگرکاربری یک جدول جدیدایجاد نماید دراینحالت سطرجدیدی درجدول SYS.TABLES درج می گردد.

بااجرای فایل CATALOG.ORA جدولی بنام DTAB ایجادمی گردد این جدول دارای دوستون TNAME (نام جدول) و REMARKS (شرح جدول) بوده ولیستی از همه جداول ونمایه های سیستمی درآن قراردارد.

برای مشاهده تمامی جداول سیستمی موجود باید دستور زیرراوارد نمایید :

```
SELECT * FROM DTAB ;
```

ازمیان جداول سیستمی موجود ، جدولهای زیردارای کاربردبیشتری هستند :

CATALOG : حاوی تمامی جداول ونمایه های قابل دسترسی برای یک کاربراست

COL : نشان دهنده ستونهای موجوددرجداول ایجادشده توسط یک کاربراست.

COLUMNS: حاوی ستون های موجود در تمامی جداول قابل دسترسی توسط یک کاربراست

INDEXES : شامل تمامی شاخص های ایجاددشه توسط یک کاربراست.

PRIVATSYN : تمامی مترادف های (Synonyms) ایجادشده توسط یک کاربر رادرخوددارد.

TAB: معرف تمامی جداول ایجادشده توسط یک کاربراست.

VIEWS: نشان دهنده تمامی نمایه های ایجادشده توسط یک کاربراست.

فصل هشتم : زبان کار با داده ها (Data Manipulation Language)

درج سطرهای جدید دریک جدول

برای افزودن سطرهای جدید به یک جدول بایدازدستور INSERT استفاده نمایید شکل کلی این دستوربصورت زیراست :

```
INSERT INTO tablename[(column1, column2,…)]
Values (value1 , value2 ,…)
```

میتوان دستورفوق را بدون ذکرنام ستونها نیزنوشت ، مشروط به آنکه مقادیرموردرج ، متناظر با نحوه تعریف ستونهای جدولی باشد (به ترتیبی که توسط دستورdesc نشان داده می شود) ولی بهتراست که همیشه اسامی ستونها را مشخص نمایید.

```
INSERT INTO DEPT ( DEPTNO , DNAME, LOC)
VALUES(50 ,'MARKETING', 'TEHRAN') ;
```

برای درج یک دپارتمان جدید ، بدون تعین نام آن دپارتمان بایددستورزیرراوارد نمایید :

```
INSERT INTO DEPT( DEPTNO ,LOC)
VALUES, (50 , 'TEHRAN') ;
```

دستورفوق را بصورت زیر نیزمیتوان نوشت :

```
INSERT INTO DEPT ( DEPTNO , DNAME, LOC)
VALUES (50 , NULL ,'TEHRAN') ;
```

مقادیر کارکتری وتاریخ باید بین دو کوتیشن باشند.

عمل درج رامیتوان با استفاده از متغیرهای جایگزین ، ازطریق ورود اطلاعات توسط کاربر انجام داد :

```
INSERT INTO DEPT ( DEPTNO , DNAME, LOC)
VALUES (&D_NUMBER, &D_NAME, &LOCATION)
```

با هربار اجرای فرمان فوق مقادیر ورودی ، ازکاربردرخواست می گردد.

برای درج مقدار تاریخ ، معمولا ازفرمت DD-MON-YY استفاده می شود دراین فرمت ، بطورپیش فرض قرن بیستم موردنظراست (19nn) بخشی ازمقدار تاریخ حاوی اطلاعاتی درمورد زمان می باشد برای نمایش تاریخ درقرن دیگر ونیزمشخص نمودن زمان بایدازتابع TO_DATE استفاده نمایید.

```
INSERT INTO EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO)
VALUES (7658,'MASON','ANALYST',7566,
TO_DATE ('24/06/2084 9:30','DD/MM/YYYY HH:MI'),3000,NULL,20);
```

با استفاده از دستور INSERT می توان سطرهای یک جدول را با استفاده از جدول دیگر کپی نمود :

```
INSERT INTO table[(column, column,…)
SELECT  select-list
From table(s);
```

بعنوان مثال برای کپی تمامی اطلاعات دپارتمان 10 بداخل جدول D10HISTORY بایددستور زیرراوارد نمایید :

```
INSERT INTO DI0HISTORY (EMPNO,ENAME, SAL , JOB , HIREDATE)
SELECT EMPNO,ENAME, SAL , JOB , HIREDAT  FROM EMP
WHERE DEPTNO=10;
```

دراینجا همانگونه که ملاحظه میکنیداز کلمه VALUES استفاده نشده است.

اصلاح سطرهای یک جدول

برای تغییر مقادیرسطرهای یک جدول بایدازدستور UPDATE استفاده نمایید شکل کلی این دستور ، بصورت زیراست :

```
UPDATE  table [alias]
SET column [, column…] =(expression, subquery)
[ WHERE condition] ;
```

بعنوان مثال برای تغییر سطرمربوط به scott ، باید دستورزیررا واردنمایید.

```
UPDATE EMP
SET JOB = 'SALESMAN' , HIREDATE= SYSDATE, SAL = SAL*1.1
WHERE ENAME = 'SCOTT' ;
```

اگرعبارت WHERE حذف شودتمامی سطرهای جدول EMP اصلاح خواهندشد دراین دستور میتوان از پرس وجوهای فرعی (subquery) و پرس وجو های فرعی همبسته (correlated subquery) نیزاستفاده نمود.

فرض کنید که جدولی ازترکیب جدیدحق ماموریت بانام COMMISSION برای تعدادی ازکارمندان داشته باشیدوبخواهید ازآن جدول ، برای اصلاح سطرهای خاصی ازجدول EMP استفاده نمایید.

COMMISSION

EMPNO	COMM
۷۴۹۹	۱۱۰۰
۷۶۵۴	۵۰۰
۷۸۴۴	۳۵۰۰
۷۸۴۴	۲۰۰۰
۷۸۴۴	۱۵۰۰

مقادیرموجوددرجدول COMMISSION ، بااستفاده ازیک پرس وجوی فرعی همبسته و یک پرس وجوی فرعی تودرتو می تواند به جدول EMP اعمال گردد :

```
UPDATE EMP
SET COMM= ( SELECT C.COMM  FROM COMMISION C
                WHERE C.EMPNO= EMP.EMPNO)
WHERE EMPNO IN ( SELECT  EMPNO FROM COMMISSION) ;
```

همانگونه که ملاحظه می کنیدجدول COMMISSION شامل بیش ازیک سطر ، به ازای یک کارمندخاص است اگربخواهید مقادیرستون COMM درجدول EMP رابامجموع حق ماموریت ها به ازای هرکارمند درجدول COMMISSION جایگزین نمایید. بایدازدستور زیراستفاده نمایید :

```
UPDATE EMP
SET  COMM= ( SELECT  SUM(C.COMM) FROM COMMISSION C
                WHERE C.EMPNO= EMP.EMPNO)
WHERE EMPNO IN ( SELECT  EMPNO FROM COMMISSION):
```

دراینحالت جدول EMP شامل مقادیرتغییریافته حق ماموریت خواهدبود.

EMP

EMPNO	COMM
۷۴۹۹	۱۱۰۰
۷۶۵۴	۶۰۰
۷۸۴۴	۳۵۰۰

احتمال دیگرآن است که بخواهید مقادیرحق ماموریت درجدول COMMISSION رابه مقادیر متناظرموجوددر جدول EMP بیافزایید. برای اینکار بایددستور زیرراوارد نمایید :

```
UPDATE EMP
SET  COMM= (SELECT  SUM(COMM)+ EMP.COMM
FROM   COMMISSION C
WHERE C.EMPNO= EMP.EMPNO
WHERE EMPNO IN(SELECT  EMPNO FROM COMMISSION);
3 recorde updated
```

EMP

EMPNO	COMM
7844	**3500**
7499	**1400**
7654	**2000**

حذف سطرهای یک جدول

برای حذف سطرهای یک جدول بایدازدستورDELETE استفاده نمایید شکل کلی این دستور بصورت زیراست :

```
DELETE FROM  table
[WHERE condition];
```

بعنوان مثال برای حذف تمامی اطلاعات مربوط به دپارتمان 10 ازجدول EMP بایددستور زیرراوارد نمایید :

```
DELETE FROM  EMP
WHERE DEPTNO =10;
```

پردازش تراکنشی (Transaction processing) :

یک تراکنش عملی است برروی بانک اطلاعاتی که متشکل ازمجموعه ای ازتغییرات بوجودآمده برروی یک یاچندجدول است دونوع تراکنش وجوددارد :

۱) تراکنشهای DML که می تواند شامل تمامی دستورات DML باشد واوراکل باهمه آنها بصورت یک واحد مستقل برخورد می کند.

۲) تراکنشهای DDL که فقط می توانند شامل یک دستور DDL باشند درهرتراکنش کاری بصورت نیمه ، رهانمی شود به ازای هرتراکنش یاهمه فعالیتها بطورکامل برروی بانک اطلاعاتی ثبت شده ویاتمامی آن ها نادیده گرفته می شوند.

یک تراکنش DML بااجرای اولین فرمان DML آغازشده وزمانی پایان می یابد که یکی ازوقایع زیررخ دهد.

- دستور COMMIT ویا ROLLBACK اجراگردد.
- یک فرمان DDL صادرشود.
- عمل خروج از سیستم اوراکل (Log off) انجام گیرد .
- اشکالی در کامپیوتربروز نماید.

یک دستورDDL بطوراتوماتیک ثبت نهایی (commit) شده ولذا بطورضمنی موجب خاتمه یک تراکنش میگردد.

تغییرات ایجادشده جهت دائمی شدن باید ثبت نهایی (commit) گردند . فرمان COMMIT این تغییرات را بصورت دائمی برروی بانک اطلاعاتی ، ثبت کرده وفرمان ROLLBACK نیزباعث نادیده گرفته شدن تمامی تغییرات ، بعدازآخرین دستورCOMMIT میگردد. درحقیقت ، تغییرات ایجادشده بین دوفرمان commit یک تراکنش رابوجودمی آورند . زمانیکه خطایی جدی مثل بروز نقص درسیستم کامپیوتر ، روی می دهد آخرین تراکنش موجود نادیده گرفته شده وعمل rollback بطورخودکار انجام می شود بااین کار جامعیت بانک اطلاعاتی بخوبی حفظ می گردد. خطاهایی نظیرذکرنادرست نام یک ستون ویااجرای یک دستور توسط کاربری که اجازه مربوطه راندارد ونظایرآن موجب عمل rollback نمی گردند. زیرااینگونه خطاها ، پیش ازاجرای دستوربررسی شده ودرصورت بروز اشکال ، اساسا اجرانمی گردند. درست بلافاصله بعدازانجام یک commit یا rollback بااجرای نخستین دستورDML یا DDL ، تراکنش جدیدی آغاز می شود.

دستورات COMMIT یا ROLLBACK راهم می توان بصورت دستی واردنمود وهم می توان بافعال کردن گزینه AUTOCOMMTT ازفرمان SET باعث اجرای اتوماتیک آنها ، بعدازانجام هرفرمان درج ، اصلاح ویاحذف گردید.

```
SET AUTO[COMMIT]  ON
SET AUTO[COMMIT]  OFF
```

کاربران یک بانک اطلاعاتی دونوع دسترسی به آن بانک اطلاعاتی دارند :

- عملیات خواندن ، نظیرفرمان SELECT
- عملیات نوشتن ، نظیر درج ، اصلاح وحذف

کسانی که داده های موجود برروی یک بانک اطلاعاتی رامی خوانند ویا بروی آن می نویسند بایدبلحاظ مشاهده صحیح داده ها ، دارای سازگاری باشند خوانندگان نباید داده ای را که درحال تغییراست مشاهده نماید. نویسندگان نیزنیاز دارند که تغییرات برروی بانک اطلاعاتی به روشی سازگار اعمال گردد بگونه ای که تغییرات اعمال شده توسط یک کاربر ، باانجام چنین عملی

توسط کاربردیگر تلاقی نداشته باشد . این امر بانگهداشتن یک کپی ازبانک اطلاعاتی در فایل (Before Image) ، پیاده سازی میگردد . وقتی که عمل درج ، اصلاح ویاحذف ، صورت می گیرد اوراکل بلافاصله یک کپی ازداده ها رابیش ازتغییر درفایل BI قرارمی دهد دراین حالت تمامی کاربران بجزکاربری که درحال تغییرداده هاست داده ها را بصورت تغییرنیافته مشاهده می کنند ودرواقع تصویری از داده ها راکه درفایلBI قراردارد ، می بینند امافقط کاربری که درحال تغییر دادهاست ، حتی پیش ازآنکه عمل commit صورت پذیرد. داده ها را بصورت تغییریافته مشاهده می کند باcommit شدن یک فرمان DML تغییرات موجود ، برروی بانک اطلاعاتی اعمال شده واز این به بعد ، دیگرکاربران نیزشاهد داده های تغییریافته خواهندبود درچنین حالتی ، فضای اشغال شده توسط فایل BI جهت استفاده مجدد ، آزاد خواهدشد .

اگرعمل تراکنش ، rollback شود ، تغییرات نادیده گرفته خواهدشددرچنین حالتی :

- داده های موجود در فایل BI برروی بانک اطلاعاتی کپی خواهند شد.
- تمامی کاربران بانک اطلاعاتی را بگونه ای که پیش ازشروع تراکنش بوده ، مشاهده خواهندکرد.

همزمانی (concnrrency) و Locking

اوراکل ، امکان دستیابی همزمان چندین کاربر را جهت خواندن داده های یکسان فراهم می آورد زیراعمل خواندن ، داده ها را تغییرنمی دهند امادریک زمان ، فقط یکنفر می تواند برروی یک جدول بنویسد. عمل lock مکانیزمی است که برای کنترل دستیابی همزمان به داده ها ، دریک سیستم چندکاربره مورداستفاده قرارمی گیرد چنین مکانیزمی مانع از اصلاح همزمان داده ها توسط بیش ازیک کاربرخواهدشد.

انواع Lock

دراوراکل دو نوع lock وجوددارد :

1) EXCLUSIVE MODE :

- دریک زمان فقط یک کاربر می تواند داده های یک جدول رااصلاح کند.
- مانع از ایجاد LOCK برروی جدول ، توسط دیگرکاربران می گردد.
- دریک زمان فقط یک کاربرمی تواند این نوع LOCK رااعمال نماید.

2) SHARE UPDATE MODE :

- بااین نوع lock بروی یک جدول ، کاربرمیتواند یک سطر ویامجموعه ای ازسطرها رابرای اصلاحات بعدی دراختیار بگیرد.
- امکان اعمال این lock را بر روی همان جدول ، به دیگر کاربران نیز میدهد .
- مانع ازاعمال lock ازنوع EXCLUSIVE بر روی همان جدول می گردد.

بطورپیش فرض ، عملیات درج ، اصلاح وحذف رکورد ، موجب اعمال Lock ازنوع EXCLUSIVE برروی کل جدول میگردند دیگرکاربران فقط می توانند عملیات پرس وجو رابر روی این جدول اجرا کرده ونمی تواند جدول مربوطه رااصلاح کرده ویاآن را lock نمایند.

بطورپیش فرض مکانیزم lock بهنگام اصلاح محتویات جداول بانک اطلاعاتی بصورت اتوماتیک توسط اوراکل انجام می شود اما می توان بااستفاده از دستور LOCK TABLE آن را بصورت دستی نیزاعمال نمود :

LOCK TABLE table_name IN mode;

```
LOCK TABLE emp  IN exclusive  mode;
```

اعمال lock از نوع EXCLUSIVE ، در حالیکه موجب سازگاری داده ها می گردد ولی باعث ایجاد محدودیت در استفاده همزمان از جداول می گردد. در یک محیط چند کاربره باید از نوعی lock استفاده شود که در آن هریک از کاربران مختلف بتوانند بطور مستقل یک یا چند سطر از یک جدول مشترک را اصلاح نمایند که به آن SHARE MODE گفته می شود. برای اعمال این lock بصورت دستی باید دستوراتی مشابه زیر را وارد نمایید :

```
LOCK TABLE dept IN share mode;
LOCK TABLE emp, dept IN share mode
```

این فرمان ، مانع ازاعمال lock ازنوع EXCLUSIVE برروی کل جدول می گردد . چندین کاربرمی تواند بطورهمزمان lock ازنوع SAHRE رابرروی یک جدول اعمال کرده وسطرهای موردنیاز خودرا بااستفاده از دستورselect for update جهت اصلاح دراختیاربگیرند. به این عملRow Level Reservation می گویند.

```
SELECT  ENAME, JOB , HIREDATE , SAL
FROM  EMP
WHERE ENAME ='SCOTT'
FOR UPDATE OF
JOB , HIREDATE , SAL
```

بادستورفوق ، سطرهای مربوطه دراختیار کاربرقرارگرفته تابرای اصلاحات بعدی نظیر اجرای دستور زیرمورداستفاده قرارگیرند :

```
UPDATE EMP
SET JOB = 'SALESMAN' , HIREDATE = SYSDATE, SAL =1.1* SAL
WHERE ENAME ='SCOTT'
```

وقتی فرمان update صادرمی شود جدول lock شده ، وکاربران دیگر ، تازمان ثبت نهایی تغییرات توسط این کاربر نمی تواند تغییراتی را برروی جدول اعمال کنناددرچنین حالتی اگردرخواست یک کاربرمبنی براعمال lock برروی یک جدول یاسطر ، بجهت اعمال توسط کاربردیگر ، عملی نگردد ، سیستم مدیریت بانک اطلاعاتی ، بطورمعمول ، کاربرمربوطه را درحالت انتظار نگه میدارد تا lock اعمال شده با اجرای فرمان commit ویا rollback آزاد گردد ، میتوان با استفاده از گزینه NOWALT ازبه انتظار نشستن کاربر ، اجتناب نمود و موجب عدم اجرای درخواست مربوطه گردید.

```
LOCK TABLE EMP IN SHARE UPDATE MODE MOWALT;
```

سطرهای یک جدول رامی توان با استفاده ازدستور SELECT…FOR UPDATE جهت اصلاح ، رزرو نمود.

دراین حالت ، کاربران دیگرنمیتوانند این سطرهای رزرو شده را اصلاح کنند ولی می توانند اقدام به اصلاح سطرهای رزرو نشده ، نمایند. دستور SELECT …FOR UPDATE یک lock از نوع SHARE UPDATE برروی جدول قرارداده ومانع ازاعمال lock از نوع EXCLUSIVE توسط دیگر کاربران می گردد درچنین وضعیتی اگرکاربری ازدستور SELECT…FOR UPDATE استفاده نکندقادربه اجرای عملیات درج ، اصلاح ویاحذف سطرهای جدول مربوطه نخواهد بود.

استفاده از ROWID

بدیهی است که پس ازاجرای یکسری ازدستورات DML ، عمل commit ، بایدهرچه سریعتر انجام شود . استفاده از ROWID (متغیر داخلی اوراکل ، برای تعیین موقعیت سطرها درجدول) باعث تسریع درعمل commit خواهدشد.

جهت بحداقل رسانیدن اثرات اصلاح سطرها برروی دیگر کاربرانی که ازیک جدول مشترک استفاده می کنند باید قواعد زیررا مدنظر قراردهید :

- ازدستورات SELECT...FOR UPDATE ویا LOCK TABLE table IN SHARE UPDATE MODE برای رزرو کردن آندسته از سطرهایی که می خواهید اصلاح کنید استفاده نمایید.
- ازآنجائیکه دسترسی دیگر کاربران به سطرهای دیگرجدول ، تاثیری برعملیات موردنظرشما ندارد ، لذا می توانید درزمان مناسب به اصلاح داده های lock شده اقدام نمایید.
- دستورات اصلاح و SELECT ... FOR UPDATE را با استفاده از ROWID بنویسید.
- عملیات اصلاح جدول را با استفاده از دستور commit یا rollback کامل نمایید.

فصل نهم : - کابران اوراکلی ومسئله امنیت داده ها

نحوه ایجاد کاربران اوراکلی

کاربرDBA می تواند با استفاده ازدستورGRANT کاربرجدیدی راایجادکرده ویاکلمه رمز یک کاربرراتغییردهد.

```
CREATE USER user_name IDENTIFIED BY password ;
GRANT option TO user IDENTIFIED BY password ;
```

گزینه option میتواندیکی ازمواردزیرباشد.

- CONNECT : قابلیت ورود (LOG ON) به اوراکل
- RESOURCE : قابلیت ایجاد جداول
- DBA : قابلیت ایجاد کاربران دیگر ونادیده گرفتن مسئله محافظت دربرابردسترسی داده ها

توجه داشته باشید که اولویتهای DBA حتی الامکان بایدمحدودگردد.

مثالی ازایجاد یک کاربراوراکلی ، دردستورزیرآمده است :

```
GRANT CONNECT , RESOURCE TO C030G1  IDENTIFIDED  BY G1;
```

همه کاربران اوراکلی بایددارای کلمه رمز باشند.

اگرکاربر SCOTT بخواهدکلمه رمزخود را به LION تغییردهدباید دستورزیرراواردکند.

```
GRANT CONNECT TO SCOTT IDENTIFIDED BY LION;
```

هرکاربر مالک جداول (table) ، نمایه ها (views) ویامترادف هایی (synonyms) است که خود ، ایجادمی کند وی می تواند آبجتهای خود را برای کاربران دیگرنیز به اشتراک بگذارد ویامنحصر به خودوراهبر بانک اطلاعاتی نماید برای دادن حق دستیابی به آبجکتهای بانک اطلاعاتی خود به دیگر کاربران باید ازدستور GRANT استفاده نمایید.

```
Grant privileges on object to user;
```

درجدول زیر ، اولویتهای موجود جهت دستیابی به جداول ویانمایه ها نشان داده شده است.

اولویت یاحق دستیابی	آبجکت
SELECT	داده های موجود دریک جدول یانمایه
INSERT	سطرهای موجود دریک جدول
UPDATE	مقادیرموجود دریک جدول یانمایه
DELETE	سطرهای یک جدول یانمایه
ALTER	تعریف ستون های یک جدول
INDEX	ایندکس های موجود برروی یک جدول
ALL	تمام موارد فوق

توجه داشته باشید که گزینه های ALTER وINDEX برای نمایه ها قابل استفاده نیستند.

ساده ترین حالت GRANT ، دادن یک اولویت به یک کاربراست به عنوان مثال اگربخواهیم اولویت یـا مجـوز SELECT بـه جدول DEPT رابه کاربر ADAMS بدهیم ، بایددستورزیرراواردنماییم.

GRANT SELECT ON DEPT TO ADAMS;

Grant succeeded

پیغامی که ظاهر میشود حاکی ازآن است که عمل مربوطه باموفقیت انجام شده است.

برای دادن چندین اولویت بایک دستور بایدآنها را باکاما ازهم جداکنید بطورمشابه برای دادن اولویت به چندین کاربر باید اسامی را باکاما ازهم جدا نمود.

بعنوان مثال بـرای دادن اولویـت INSERT و UPDATE بـرروی جـدول DEPT ، بـه هردوکـاربر ADAMS و JONES بایددستور زیرراوارد نمایید.

GRANT INSERT , UPDATE ON DEPT TO ADAMS , JONES;

برای دادن تمامی اولویتها به کاربر ADAMS باید دستورزیررا اجرا کنید.

GRANT ALL ON DEPT TO ADAMS;

اگربادستور GRANT ، به کاربری ، اولویتهای مشخصی داده شود کاربرمربوطه بطورمعمول اجـازه دادن همـان اولویتهـا را بـه دیگران ندارد مگرآنکه دردستورGRANT ، ازگزینه WITH GRANT OPTION استفاده شود.

بعنوان مثال برای دادن مجوز SELECT برروی جدول EMP به کـاربر ADAMS یـاحق دادن مجـوز بـه دیگـران ، بایـد دستور زیر را وارد نمایید :

GRANT SELECT ON EMP TO ADAMS WTTH GRANT OPTION;

برای دادن اولویت به همه کاربران می توان ازعبارت PUBLIC استفاده نمود.

GRANT SELECT ON EMP TO PUBLIC ;

اگربخواهید عملی را که مجاز به انجام آن نیستید برروی جدولی اعمال کنید باخطای ارواکل مواجه خواهید شد بعنـوان نمونـه خطای table or view does not exist می تواند حاصل یکی از دوعامل زیرباشد.

- نام جدولی را که وجودندارد بکاربرده اید.
- اقدام به اجرای عملی برروی یک جدول ویانمایه کرده اید که اجازه آن را نداشته اید.

برای بازپس گرفتن اولویتهای موجود برای یک کاربر باید ازفرمان REVOKE استفاده نمایید :

REVOKE privileges on table/view FROM users;

درچنین حالتی تمامی اولویتها ازاین کاربر ونیزهمه کاربرانی که توسط وی دارای اولویت های لازم شده اند از بین می رود.

برای بازپس گرفتن همه اولویتهای موجود برروی جدول DEPT ازکاربر ADAMS بایددستور زیرراوارد نمایید.

REVOKE ALL ON DEPT FROM ADAMS;

اولویت های PUBLIC نیزتوسط دستورREVOKE قابل بازپس گیری است.

REVOKE SELECT ON EMP FROM PUBLIC;

نحوه ایجادمترادف (SYNONYM) برای یک جدول

برای ارجاع به جدولی ، که مالک آن کاربردیگری است بایدحتما از پیشوند نام کاربر وعلامت نقطه بهمـراه نـام جـدول اسـتفاده نمود بعنوان مثال برای ارجاع به جدول EMP که مالک آن SCOTT است باید ازدستور زیراستفاده نمایید :

```
SELECT  *  FROM  SCOTT.EMP;
```

برای فراهم کردن امکان ارجاع به جدول EMP بدون نیاز به پیشوند SCOTT میتوان ازمترادفها استفاده کرد .

حال میتوان بسادگی ازدستورزیر استفاده نمود :

```
SELECT  *
FROM  EMP ;
```

فقط راهبرسیستم (DBA) میتواند مترداف های PUBLIC ایجادنماید.

```
CREATE PUBLIC SYNONYM  synonym_name   FOR [owner.]object_name ;
```

بادستور زیرمیتوان یک مترداف راحذف نمود.

```
DROP  [PUBLIC] SYNONYM synonym_name ;
```