



---

## ترجمہ فصل اول

---



احمد مجتبیٰ الکوزی

## PL/SQL پیژندنه

"Procedural Language extensions to the Structured Query Language" د SQL اوس د اړیکو لرونکو ډیټابیسونو لپاره یوه عامه ژبه ده چې د معلوماتو د پوښتنې (query) او تازه کولو (update) لپاره کارول کېږي

Oracle Corporation دا ژبه PL/SQL معرفي کړه خو د SQL ځینې محدودیتونه له منځه یوسي او هغو کسانو ته بشپړه برنامه‌سازي حل لاره برابر کړي چې غواړي مهم او حیاتي اپلیکیشنونه د Oracle ډیټابیس پر بنسټ جوړ کړي

دا فصل PL/SQL د هغې سرچینه او مختلف نسخې یې معرفي کوي همداراز دا د Oracle Database 12c په نوي نسخه کې د PL/SQL یوه لنډه خلاصه وړاندې کوي په پای کې دا فصل PL/SQL پراختیا ورکونکو ته د نورو سرچینو لپاره لارښود او ځینې مشورې هم وړاندې کوي

## PL/SQL څه شی دی؟

د Oracle PL/SQL ژبه ځینې مهمې ځانګړتیاوې لري:

دا ډېره منظم لوست‌وړ او لاس‌رسی‌وړ ژبه ده که تاسو برنامه‌سازي ته نوي یاست PL/SQL یو ښه پیل دی دا ژبه زده کول اسانه ده او ډېر واضح جوړښت او کلیدي کلمې لري چې ستاسو د کوډ هدف ښکاره بیانوي که تاسو د نورو پروګرامنګ ژبو تجربه لرئ نو د PL/SQL نحو (syntax) ته به ژر عادت شئ

دا د Oracle پراختیا لپاره یوه معیاري او انتقال‌ېدونکې ژبه ده که تاسو یو PL/SQL پروسیجر یا فنکشن ولیکئ او هغه د خپل لپ‌تاپ Oracle ډیټابیس کې اجرا کړئ کولای شئ همدغه پروسیجر پرته له بدلون څخه د خپل شرکت په شبکې کې په بل Oracle ډیټابیس کې هم اجرا کړئ (البته د نسخو توافق ته په پام سره)

"یوه واره ولیکه هرچېرې یې وچلوه" دا شعار د Java نه مخکې د PL/SQL و البته دلته "هرچېرې" مطلب دا دی هر ځای چې Oracle ډیټابیس وي دا یوه ځای کړل شوې (Embedded) ژبه ده

PL/SQL داسې طرحه شوې نه ده چې یو خپلواکه (standalone) ژبه وي بلکه دا باید د یوه میزبان چاپیریال (host environment) له دننه وغوښتل شي

د بېلګې په توګه تاسې کولای شئ PL/SQL پروګرامونه د ډیټابیس له دننه څخه لکه د SQL\*Plus له لارې (اجرا کړئ

یا هم تاسې کولای شئ PL/SQL پروګرامونه د Oracle Developer فورم یا راپورونو (Reports) له لارې تعریف او اجرا کړئ) دا طریقه د "client-side PL/SQL" په نوم یادېږي.

خو تاسې نشئ کولای یو داسې PL/SQL اجرا کېدونکی فایل جوړ کړئ چې یوازې په خپله وچلېږي

دا یوه لوړ فعالیت لرونکې، بشپړ یوځای شوې ډیټابیس ژبه ده

نن سبا د Oracle ډیټابیس لپاره د پروګرام لیکلو لپاره تاسو ګڼ انتخابونه لرئ لکه Java او JDBC، Visual Basic او ODBC، یا هم Delphi، C++ او نور.

خو تاسو به ومومئ چې د Oracle ډیټابیس ته د لاسرسي لپاره په PL/SQL کې لوړ مؤثریت لرونکی کوډ لیکل اسانه دي نسبت نورو ژبو ته

په ځانګړي ډول Oracle ځینې PL/SQL ته ځانګړي اصلاحات وړاندې کوي لکه FORALL بیان چې کولی شي د ډیټابیس فعالیت څو برابره لوړ کړي

د PL/SQL سرچینه (Origins)

Oracle Corporation د پروګرام جوړونې او ډیټابیس ډیزاین لپاره د اعلاميې محور (declarative) او غیر 절차ي (nonprocedural) طریقه په وړاندې کولو کې د سافټویر صنعت مشري کړې ده

د Oracle سرور ټکنالوژي د نړۍ له پیاوړو پرمختللو او باثباته اړیکو لرونکو ډیټابیسونو څخه شمېرل کېږي.

د دې د اپلیکیشن جوړونې وسایل لکه Oracle Forms د "خپل سکرین رسم کړه" طریقه کاروي چې ډېر چمتو فعالیتونه لري او پراختیا ورکونکي له ډېر کوډ لیکلو ژغوري

د PL/SQL لومړني کلونه

د Oracle په لومړيو کلونو کې د SQL اعلاميې محور طریقه او د اړیکو (relational) ټکنالوژۍ نوښت کافي و چې پراختیا ورکونکي پرې قانع شي خو کله چې صنعت وده وکړه غوښتنې زیاتې شوې او اړتیاوې پیچلې شوې پراختیا ورکونکو ته اړتیا وه چې د محصولاتو "لاندې سطحه" ته لاړ شي هغوی غوښتل پیچلې فارمولې، استثناوې او اصول خپلو فورمونو او ډیټابېس کوډونو کې ځای کړي

په 1988 کال کې، Oracle د خپل ډیټابېس ټکنالوژۍ یو مهم پرمختګ خپور کړ Oracle : نسخه 6 .

د دې نسخې مهمه برخه د "procedural option" یا PL/SQL معرفي وه په نږدې وخت کې Oracle د SQL\*Forms نسخې 2.3 ته اوږده تمه شوې ارتقا هم خپره کړه) دا هغه محصول دی چې اوس د Oracle Forms په نوم پېژندل کېږي .  
په SQL\*Forms نسخه 3 کې د لومړي ځل لپاره د PL/SQL انجن د وسیلو (tools) برخه وگرځېد او پراختیا ورکونکو ته یې دا اجازه ورکړه چې خپله منطقي کوډونه په طبیعي ساده طریقه ولیکي.

د PL/SQL لومړۍ نسخه ډېر محدود امکانات لرل د سرور په اړخ کې، PL/SQL یوازې د "batch processing" لپاره د پروسیجرونو او SQL بیانونو د اسکرېپتونو جوړولو لپاره کارېده نه مودیولر اپلیکیشن جوړولای شو نه د سوداګریزو اصولو ذخیره کولای شو د کلاینټ اړخ کې SQL\*Forms v3.0 اجازه ورکوله چې پروسیجرونه او فنکشنونه جوړ شي، خو د فنکشنونو ملاتړ مستند نه و نو ډېر پراختیاکوونکي کلونه کلونه له دې څخه استفاده نه کوله. سربېره پر دې دا نسخه د آرې (array) ملاتړ نه درلود او له عملیاتي سیستم سره تعامل نه شو کولای (نه د داخلي لپاره، نه د وتلو لپاره) دا له یوه بشپړ پروګرامنګ ژبې څخه ډېر لرې وه

خو د ټولو محدودیتونو سره سره PL/SQL د پراختیاکوونکو له خوا په تود هرکلي سره ومنل شو. د دې لپاره ډېر تنده وه چې یو ساده IF بیان د SQL\*Forms دننه ولیکل شي د څو SQL بیانونو د batch پروسس کولو اړتیا ډېره وه

هغه څه چې ډېر پراختیاکوونکي نه پوهېدل دا وو چې د PL/SQL اصلي انگیزه او لید له دې

څخه ډېره پراخه وه چې يوازې د SQL\*Forms په څېر محصولاتو کې پروگرامي کنټرول ورکړي. د Oracle د ډيټابېس او وسيلو د ژوند په لومړيو پړاوونو کې، Oracle Corporation دوه مهمې ستونزې پېژندلې وې: د انتقال وړتيا نشتوالی او د اجرا واک ستونزې.

د اپليکيشن د انتقال وړتيا بڼه والی

د انتقال وړتيا اندېښنه ممکن هغو کسانو ته عجيبه ښکاري چې د Oracle Corporation د بازار موندنې او تخنیکي ستراتيژيو سره بلد دي د Oracle حل يوه مهمه ځانگړنه له ۱۹۸۰ مو کلونو راهيسې همدا انتقال وړتيا وه کله چې PL/SQL راڅرگند شو، د C پر بنسټ ډيټابېس پر گڼو عملياتي سيستمونو او هارډوېرونو چلېده SQL\_Plus او SQL\_Forms په اسانۍ سره له بېلابېلو ترمينلونو سره برابرېدل. خو له دې ټولو سره سره، لا هم داسې اپليکيشنونه وو چې COBOL ، C ، او FORTRAN په څېر ژبو له خوا وړاندې شوي دقيق کنټرول ته اړتيا لرله. کله چې پراختياکوونکی له Oracle وسيلو څخه بهر قدم کېښود، اپليکيشن نور انتقال وړ نه و

PL/SQL ژبه دا هدف درلود (او لري) چې د اپليکيشن اړتياوې پراخې کړي خو دا ټول د عملياتي سيستم څخه بې واسطه پروگرامنگ وسيلو له لارې پوره شي نن ورځ، Java او نورې ژبې ورته انتقال وړتيا وړاندې کوي خو PL/SQL په دې برخه کې يو مخکښ pioneer گڼل کېږي او لا هم پراختياکوونکو ته دا توان ورکوي چې ډېر انتقال وړ اپليکيشن کوډ وليکي.

د اجرا واک او د معاملي بشپړتيا بڼه والی

د انتقال وړتيا نه هم مهمه موضوع د اجرا واک ده ډيټابېس او SQL ژبه تاسو ته دا توان درکوي چې د هر ځانگړي جدول ته لاسرسی او بدلونونه په کلکه کنټرول کړئ. د بېلگې په توگه د GRANT قوماندې له لارې تاسو ډاډ تر لاسه کولای شئ چې يوازې ټاکلي رولونه او کاروونکي کولای شي په يوه جدول UPDATE ترسره کړي خو دا GRANT قومانده دا تضمين نه کوي چې کاروونکی به د جدولونو د بدلونونو سمه لړۍ تعقيب کړي، کوم چې اکثره سوداگريزو معاملو ته اړتيا وي

PL/SQL ژبه د منطقي معاملو د کنټرول او مدیریت لپاره کلک واک ورکوي یو لاره چې PL/SQL دا کار کوي د اجرا واک پلي کول دي پر خای د دې چې یو رول یا کاروونکي ته د جدول د تازه کولو واک ورکړئ تاسو یوازې د پروسیجر اجرا کولو واک ورکړئ کوم چې د اساسي معلوماتو جوړښتونو ته لاسرسی کنټرولوي دا پروسیجر د یو بل Oracle ډیټابیس سکیمې (د پروگرام "تعریف کوونکي") پورې اړه لري چې هغه ته د اړینو جدولونو د تازه کولو اصلي واک ورکړل شوی وي. نو پروسیجر د معاملې لپاره د "دروازې ساتونکي" رول لوبوي. یوازینی لاره چې یو پروگرام) که Oracle Forms اپلیکیشن وي یا Pro\*C اجرای فایل (دا انتقال ترسره کولای شي د همدغه پروسیجر له لارې ده. په دې ډول د اپلیکیشن د معاملې بشپړتیا تضمین کېږي

د Oracle8i ډیټابیس له پیل سره Oracle د PL/SQL د اجرا واک ماډل ته د AUTHID کلیرې له لارې ډېر انعطاف ورکړ د AUTHID سره تاسو کولای شئ خپل پروگرامونه د تعریف کوونکي د واک ماډل لاندې وچلوئ یا AUTHID CURRENT\_USER د اجرا کوونکي واک) وټاکئ، چې په هغه حالت کې پروگرامونه د اجرا کوونکي سکیمې واک لاندې چلېږي د اجرا کوونکي واک یوازې یو مثال دی چې PL/SQL څنګه پیاوړی او انعطاف‌پذیر شوی دی

عاجز پیل، دوامداره پرمختګ

که څه هم SQL پیاوړې ژبه ده خو پراختیاکوونکو ته دا انعطاف او ځواک نه ورکوي چې بشپړ اپلیکیشنونه جوړ کړي د Oracle PL/SQL ژبه دا تضمین کوي چې موږ کولای شو د عملیاتي سیستم څخه بې‌واسطه د Oracle چاپیریال کې پاتې شو او لا هم ډېر مؤثر اپلیکیشنونه ولیکو چې د کاروونکو اړتیاوې پوره کړي

PL/SQL له خپلو عاجزو پیلونو څخه ډېر پرمختګ کړی دی د [1.0](#) PL/SQL سره دا معمول وه چې پراختیاکوونکي به خپلو مدیرانو ته ویل "دا کار د PL/SQL سره نه شي کېدلای." نن ورځ دا جمله له حقیقت څخه بهانې ته بدله شوې ده که کله تاسو له یوې اړتیا سره مخ شئ او ووايئ "دا کار نشي کېدلای" نو مهرباني وکړئ دا خبره خپل مدیر ته مه کوئ. پر خای یې، ژور تحقیق وکړئ، یا د Oracle له خوا وړاندې شوي PL/SQL پکیجونه وپلټئ ډېر احتمال شته چې نن PL/SQL تاسو ته دا توان درکړي چې هر څه ترسره کړئ

د کلونو په اوږدو کې Oracle Corporation د PL/SQL ژبې ته خپله ژمنتیا بنودلې ده چې دا یې خپله مهمه خصوصي پروگرامنګ ژبه ده د هرې نوي ډیټابیس نسخې سره Oracle د PL/SQL ژبې ته بنسټیز او دوامداره پرمختګونه ورکړي دي دا ژبه ته ګڼ شمېر جوړ شوي (built-in) پکیجونه ور زیات کړي چې PL/SQL ته په بېلابېلو لارو پراختیا ورکړي دا د شې محور وړتیاوې معرفي کړي د آرې په څېر معلوماتي جوړښتونه پلي کړي کمپایلر یې ښه کړی چې هم کوډ ته اصلاح ورکړي او هم د کیفیت او اجرایې ستونزو خبرداری ورکړي او په عمومي ډول د ژبې پراخوالی او ژوروالی یې ښه کړی دی.

## دا دی PL/SQL

که تاسو په بشپړ ډول له پروگرامنګ یا له PL/SQL یا حتی (SQL سره نوي یاست نو د PL/SQL زده کړه به درته یو څه وېرونکې ښکاره شي. که همداسې وي اندېښنه مه کوئ زه باور لرم چې دا به تر هغه څه ډېر اسانه وي چې تاسو یې فکر کوئ زما د خوشبینۍ دوه دلیلونه دي

د کمپیوټر ژبې په عمومي ډول زده کول ډېر سخت نه دي لږ تر لږه د دویم یا درېیم انساني ژبې په پرتله ولې؟ ځکه چې کمپیوټرونه ډېر هوښیار نه دي (هغوی عملیات ډېر ژر ترسره کوي، خو هېڅ ډول خلاقیت نه لري) موږ باید له ډېر سخت نحو (syntax) څخه کار واخلو څو کمپیوټر ته ووايو چې څه باید وکړي نو ژبه هم سخت جوړښت لري (هېڅ استثنا نشته!) او له همدې امله زده کول یې اسانه دي

PL/SQL - واقعاً یوه اسانه ژبه ده د نورو پروگرامنګ ژبو په پرتله دا د "بلاک" جوړښت باندې ولاړه ده چې بېلابېل برخې لري او هره برخه له واضح ځان تشریح کونکو کلیمو سره پېژندل کېږي

راځئ څو بېلګې وګورو چې د PL/SQL جوړښت او فعالیت مهم عناصر ښيي

له SQL سره یوځایوالی

د PL/SQL یوه مهمه ځانګړنه له SQL سره نږدې یوځایوالی دی تاسو اړتیا نه لرئ چې له منځګړي سافټویر لکه (ODBC (Open Database Connectivity یا JDBC

(Database Connectivity) څخه کار واخلئ څو SQL بیانونه د PL/SQL پروګرامونو کې وچلوئ پر ځای یې، تاسو یوازې UPDATE یا SELECT بیانونه مستقیم خپل کوډ کې ځای پر ځای کوئ لکه څنګه چې دلته ښودل شوي

```
```psql
1 DECLARE
2 l_book_count INTEGER;

3
4 BEGIN
5 SELECT COUNT(*)
6 INTO l_book_count
7 FROM books
8 WHERE author LIKE '%FEUERSTEIN, STEVEN%';

9
10 DBMS_OUTPUT.PUT_LINE (
11 'Steven has written (or co-written) ' ||
12 l_book_count ||
13 ' books.');
```

15 او، ما خپل نوم بدل کړی، نو...

```
16 UPDATE books
17 SET author = REPLACE (author, 'STEVEN', 'STEPHEN')
18 WHERE author LIKE '%FEUERSTEIN, STEVEN%';
19 END;
```
```

راځئ چې دا کوډ په تفصیل سره وڅېړو، چې د PL/SQL د جوړښت او فعالیت نور جزئیات روښانه کړي.

د کړنو تشریح



کړېښي ۱-۳: دا د PL/SQL بلاک د اعلان برخه ده چې "بې-نومه" بلاک بلل کېږي دلته یو عدد ډول متغیر اعلان شوی څو د هغو کتابونو شمېر وساتي چې ما لیکلي یا شریک لیکلي دي (د بلاک جوړښت په اړه به په درېیم فصل کې نور تفصیل ورکړل شي)

کړېښه ۴: د `BEGIN` کلیمه د اجرا برخې پیل ښيي—یعنې هغه کوډ چې کله دا بلاک SQL\*Plus ته وسپارل شي اجرا به شي

کړېښي ۵-۸: دلته یو پوښتنه (query) اجرا کېږي څو معلومه شي چې ما څو کتابونه لیکلي یا شریک لیکلي دي. کړېښه ۶ ځانګړې توجه غواړي `INTO`: کلیمه د SQL بیان برخه نه ده بلکې دا د ډیټابېس څخه PL/SQL محلي متغیرونو ته د معلوماتو انتقال وسیله ده

کړېښي ۱۰-۱۳: دلته د `DBMS\_OUTPUT.PUT\_LINE` جوړ شوي پروسیجر څخه استفاده شوې) یعنې د Oracle لخوا ورکړل شوی DBMS\_OUTPUT پکیچ کې پروسیجر ( څو د کتابونو شمېر ښکاره کړي

کړېښه ۱۵: دا یوه کړېښه تبصره ده چې د UPDATE هدف تشریح کوي

کړېښي ۱۶-۱۸: \* دلته لیکوال پرېکړه کړې چې د خپل نوم املا له "STEVEN" څخه "STEPHEN" ته بدله کړي، نو د `books` جدول کې UPDATE اجرا کوي. د `REPLACE` جوړې شوې فنکشن څخه استفاده کوي څو ټول "STEVEN" موارد پیدا کړي او "STEPHEN" ته یې بدل کړي.

---

کنټرول او شرطی منطق

PL/SQL پراخ ډول بیانونه وړاندې کوي چې موږ ته دا توان راکوي چې د پروګرام کومې کړېښې اجرا شي په کلکه کنټرول کړو دا بیانونه عبارت دي له

IF - او CASE بیانونه دا شرطی منطق پلي کوي د بېلګې په توګه "که د کتاب پاڼو شمېر له ۱۰۰۰ څخه زیات وي نو

-د تکرار لپاره بشپړ کنټرولونه: پکې شامل دي 'FOR' لوپ 'WHILE' لوپ او ساده 'LOOP'.

-د GOTO بیان: هو، PL/SQL حتی 'GOTO' هم لري چې تاسو ته اجازه درکوي له یوه ځای څخه بل ځای ته بی‌شرط انتقال وکړئ خو دا معنا نه لري چې باید واقعاً یې وکاروئ

د پروسیجر بېلگه

دلته یو پروسیجر دی (یعنې د کوډ یو بیا کارېدونکی بلاک چې د نوم له لارې غږول کېږي) چې ځینې له پورته ځانگړنو ښيي

```
```sql
1 PROCEDURE pay_out_balance (
2 account_id_in IN accounts.id%TYPE)
3 IS
4 l_balance_remaining NUMBER;
5 BEGIN
6 LOOP
7 l_balance_remaining:= account_balance (account_id_in);

9 IF l_balance_remaining < 1000
10 THEN
11 EXIT;
12 ELSE
13 apply_balance (account_id_in, l_balance_remaining);
14 END IF;
15 END LOOP;
16 END pay_out_balance;
```
```

راځئ چې دا کوډ نور تفصیل سره وڅېړو څو د PL/SQL د کنټرول منطق او تکرار جوړښتونه ښه درک کړو

د کرښو تشریح

کرښې ۱-۲: دا د یو پروسیجر سرلیک دی چې د حساب پاتې پیسې د پاتې بلونو د پوښښ لپاره ورکوي کرښه ۲ د پروسیجر د پارامیټرونو لیست ښیي چې دلته یوازې یو داخلېدونکی ارزښت دی (د حساب پېژند شمېر)

کرښې ۳-۴: دا د پروسیجر د اعلان برخه ده. پام وکړئ چې دلته د DECLARE کلیمه نه ده کارول شوې لکه پخوانۍ بېلگه کې، بلکې د IS یا AS کلیمه کارول شوې څو سرلیک له اعلانونو څخه جلا کړي

کرښې ۶-۱۵: دا د ساده لوپ یوه بېلگه ده دا لوپ د EXIT بیان (کرښه ۱۱) له لارې پای ته رسېږي FO او WHILE لوپونه د پای شرط په بله طریقه مشخصوي.

کرښه ۷: دلته د `account\_balance` فنکشن ته غږ کېږي څو د حساب پاتې پیسې ترلاسه شي دا د یو بیا کارېدونکي پروګرام دننه بل بیا کارېدونکي پروګرام ته د غږ کولو بېلگه ده کرښه ۱۳ د بل پروسیجر دننه د پروسیجر غږول ښیي

کرښې ۹-۱۴: دلته یو IF بیان دی چې داسې تعبیرېږي که د حساب پاتې پیسې له \$۱۰۰۰ څخه کمې وي نو د بلونو لپاره پیسې ورکول بند کړئ که نه نو پاتې پیسې د بل مصرف لپاره وکاروئ

کله چې ستونزه پېښ شي

د PL/SQL ژبه یو پیاوړی میکانیزم وړاندې کوي د تېروتنو راپور ورکولو او مدیریت لپاره په لاندې پروسیجر کې زه د حساب نوم او پاتې پیسې د ID له لارې ترلاسه کوم بیا گورم

چې آيا پاتې پيسې ډبرې كمې دي. كه داسې وي زه په واضح ډول يوه استثنا راپورته كوم  
چې پروگرام له دوام څخه بندوي

```
```sql
1 PROCEDURE check_account (
2 account_id_in IN accounts.id%TYPE)
3 IS
4 l_balance_remaining NUMBER;
5 l_balance_below_minimum EXCEPTION;
6 l_account_name accounts.name%TYPE;
7 BEGIN
8 SELECT name
9 INTO l_account_name
10 FROM accounts
11 WHERE id = account_id_in;

13 l_balance_remaining:= account_balance (account_id_in);

15 DBMS_OUTPUT.PUT_LINE (
16 'Balance for ' || l_account_name ||
17 ' = ' || l_balance_remaining);

19 IF l_balance_remaining < 1000
20 THEN
21 RAISE l_balance_below_minimum;
22 END IF;

24 EXCEPTION
25 WHEN NO_DATA_FOUND
26 THEN
```

-- 27 د دي ID لپاره حساب ونه موندل شو

28 log\_error (...);

29 RAISE;

30 WHEN l\_balance\_below\_minimum

31 THEN

32 log\_error (...);

33 RAISE VALUE\_ERROR;

34 END;

...

راځي چې د دي کوډ د تېروتنې مدیریت اړخونه په تفصیل سره وڅېړو

د کرښو تشریح

کرښه ۵: دلته زه خپله ځانگړې استثنا اعلانوم چې نوم يې

`l\_balance\_below\_minimum` دی Oracle. یو شمېر مخکې ټاکل شوي استثناوې

وړاندې کوي لکه `DUP\_VAL\_ON\_INDEX` خو زه داسې استثنا ته اړتیا لرم چې زما

اپلیکیشن ته ځانگړې وي نو باید دا خپله اعلان کړم

کرښې ۸-۱۱: دا پوښتنه (query) د حساب نوم ترلاسه کوي که د ورکړل شوي ID لپاره حساب موجود نه وي دیتابیس د مخکې ټاکل شوي `NO\_DATA\_FOUND` استثنا راپورته کوي چې پروگرام بندوي

کرښې ۱۹-۲۲: که پاتې پیسې دېرې کمې وي زه خپله اعلان شوي استثنا په واضح ډول راپورته کوم ځکه چې دا د حساب له جدي ستونزې سره مخ دی

کرښه ۲۴: د `EXCEPTION` کلیمه د اجرائي برخې پای او د استثنا مدیریت برخې پیل ښيي چې پکې تېروتنې اداره کېږي

کرښې ۲۵-۲۸: دا د تېروتنې مدیریت برخه د کله چې حساب ونه موندل شي که `NO\_DATA\_FOUND` استثنا راپورته شوې وي دلته نیول کېږي او تېروتنه د

`log\_error` پروسیجر له لارې ثبتېږي بیا هماغه استثنا بیا راپورته کېږي څو بهرنۍ بلاک ته دا معلومه شي چې د حساب ID سره هېڅ حساب نه و

کرنې ۳۰-۳۳: دا د تېروتنې مدیریت برخه ده کله چې د حساب پاتې پیسې ډېرې کمې شي (یعنې زما ځانګړې استثنا). که `l\_balance\_below\_minimum` راپورته شي دلته نیول کېږي، تېروتنه ثبتېږي او بیا د سیستم له خوا ټاکل شوې `VALUE\_ERROR` استثنا راپورته کېږي، څو بهرنۍ بلاک ته ستونزه ور وښيي

شپږم فصل تاسو د PL/SQL د تېروتنې مدیریت میکانیزمونو پراخ سفر ته بیايي البته د PL/SQL په اړه ډېر نور څه ویل کېدای شي—همدا وجه ده چې تاسو د دې کتاب کې سلگونه نور مخونه مطالعه کوئ دا ابتدایي بېلګې باید تاسو ته دا احساس درکړي چې د PL/SQL سره به څه ډول کوډ ولیکئ د دې ژبې مهم نحوي عناصر څه دي او دا چې د PL/SQL کوډ لیکل او لوستل څومره اسانه دي

د PL/SQL نسخو په اړه

د Oracle د ډیټابېس هره نسخه له خپلې ځانګړې PL/SQL نسخې سره راځي کله چې تاسو د PL/SQL نوې نسخې کاروئ نو تاسو ته د فعالیتونو پراخي اسانتیاوې وړاندې کېږي زموږ د PL/SQL پروګرامرانو لپاره یوه ستره ننگونه دا ده چې له پرمختګونو سره ځان همغږی وساتو. موږ باید تل ځانونه د هرې نسخې له نوو ځانګړنو سره بلد کړو—دا زده کړو چې څنګه یې وکاروو، څنګه یې خپلو اپلیکېشنونو کې پلي کړو، او دا وپېژنو چې کومې نوې تخنیکونه دومره ګټور دي چې باید موجود اپلیکېشنونه هم ورسره اصلاح کړو

د لاندې جدول (Table 1-1) له لارې د PL/SQL د پخوانیو او اوسنیو نسخو مهم عناصر لنډیز شوي دي (یادونه: د ډیټابېس په لومړیو نسخو کې د PL/SQL نسخې شمېرې له ډیټابېس نسخو سره توپیر درلود خو له Oracle8 ډیټابېس وروسته دا شمېرې سره برابرې شوې دي.) دا جدول د هرې نسخې نوې ځانګړنې په لنډ ډول ښيي د جدول وروسته تاسو به د Oracle Database 12c په وروستۍ نسخه کې د PL/SQL نوو ځانګړنو تفصیلي تشریح ومومئ

د Oracle Developer محصولاتو مجموعه هم له خپلې PL/SQL نسخې سره راځي خو دا معمولاً له هغه نسخې وروسته وي چې د Oracle ډیټابیس کې موجوده وي دا فصل (او ټول کتاب) د سرور اړخ PL/SQL پروگرامنگ تمرکز لري

جدول 1-1: Oracle ډیټابیس او PL/SQL نسخې

د Oracle ډیټابیس نسخه د PL/SQL نسخې ځانګړنې \*6.0\* د PL/SQL ابتدايي نسخه (1.0) یوازې د SQL\_Plus لپاره د اسکرېپټ ژبې په توګه کارېده (لا تر اوسه نوملرونکي بیا کارېدونکي او غږېدونکي پروګرامونه جوړېدل نه کېدل) او همداراز د SQL\_F Forms 3 کې د پروګرامنگ ژبې په توګه \*7.0\* د لوی پرمختګ (نسخه 2.0 د 1.0 PL/SQL ته د ذخیره شوو پروسیجرونو فنکشنونو پکیجونو د پروګرامر لخوا تعریف شوو ریکارډونو PL/SQL جدولونو) اوس ورته collections ویل کېږي، او ګڼو پکیج پراخونو ملاتړ وړ اضافه کړ \*7.1\* د نسخه (2.1) د پروګرامر لخوا تعریف شوي فرعي ډولونه ملاتړ کړل د ذخیره شوو فنکشنونو کارول د SQL بیانونو دننه ممکن کړل او د DBMS\_SQL پکیج له لارې متحرک SQL وړاندې شو له 2.1 PL/SQL سره تاسو کولای شئ د SQL DDL بیانونه د PL/SQL پروګرامونو دننه اجرا کړئ \*7.3\* د نسخه (2.3) د collections فعالیتونه پراخ کړل د remote dependency مدیریت ښه کړ د UTL\_FILE پکیج له لارې د فایل O/اورتیاوې وړ زیاتې کړې او د cursor متغیرونو پلي کول بشپړ کړل \*8.0\* د PL/SQL نوې نسخه (8.0) د Oracle هڅه ښيي چې د اړوندو محصولاتو نسخه شمېرې سره همغږي کړي 8.0 PL/SQL د Oracle8 ډیټابیس پرمختګونه ملاتړ کړل چې پکې لوی شیان (LOBs) شې محوړ ډیزاین او پراختیا (VARRAYs) او nested tables او د Oracle/Advanced Queuing اسانتیا (Oracle/AQ) شامل وو \*8.1\* د Oracle د i-series لومړۍ نسخه اړونده PL/SQL نسخه ډېرې اغېزناکې نوې ځانګړنې وړاندې کړې پکې د متحرک SQL نوې نسخه د ډیټابیس دننه د Java ملاتړ د اجرا کوونکي واک ماډل د اجرا واک اختیار خپلواکي معاملي (autonomous transactions) او د لوړ فعالیت DML "bulk" او پوښتنې شاملې وې

د Oracle ډیټابیس نسخې او PL/SQL ځانګړنې

د Oracle ډیټابیس نسخه د PL/SQL ځانګړنې \*9.1\* د Oracle9i لومړۍ نسخه ژر له

پخوانۍ نسخې وروسته خپره شوه دا نسخه د object-type توکو د وراثت ملاتړ د جدول فنکشنونه ( cursor expressions د PL/SQL فنکشنونو موازي اجرا ته اجازه ورکوي) خو-سطحي collections او CASE بيانونه وړاندې کرل \*9.2\*. د Oracle9i دويمه نسخه XML ته ډېر تمرکز ورکړ خو PL/SQL پراختياکوونکو ته يې هم نوې اسانتياوې ورکړې لکه associative arrays چې د VARCHAR2 رسته وو له لارې هم index کېدای شي ( record-based DML لکه د record له لارې insert کول) او د UTL\_FILE پکيچ ډېر پرمختگونه \*10.1\*. د Oracle Database 10g لومړۍ نسخه په 2004 کې معرفي شوه چې د grid computing ملاتړ يې تمرکز و د PL/SQL له نظره، مهمې ځانگړنې optimized compiler او compile-time warnings وې چې پراختياکوونکو ته په شفاف ډول وړاندې شوي \*10.2\*. د Oracle Database 10g دويمه نسخه په 2005 کې خپره شوه چې PL/SQL ته لږ شمېر نوې ځانگړنې ورکړې تر ټولو مهمه يې د preprocessor syntax ملاتړ و چې د Boolean expressions پر بنسټ د پروگرام برخې مشروط ډول compile کول ممکنوي \*11.1\*. د Oracle Database 11g لومړۍ نسخه په 2007 کې راغله د PL/SQL لپاره تر ټولو مهمه ځانگړنه د function result cache، سربېره پر دې compound triggers، CONTINUE بيان، او native compilation چې ماشين کوډ توليدوي هم پکې شامل وو \*11.2\*. دا نسخه د 2009 په مې کې خپره شوه تر ټولو مهمه ځانگړنه يې edition-based redefinition وه چې مديرانو ته اجازه ورکوي "hotpatch" اپليکيشنونه پداسې حال کې چې کاروونکي يې کاروي \*12.1\*. د Oracle Database 12c لومړۍ نسخه په جون 2013 کې خپره شوه دا د پروگرام واحدونو او views ته د لاسرسي او واکونو مدیریت کې پرمختگونه وړاندې کوي SQL او PL/SQL ژبې سره نږدې کوي په ځانگړي ډول د VARCHAR2 اعظمي اوږدوالي او dynamic SQL binding برخه کې د ساده فنکشنونو تعريف د SQL بيانونو دننه ممکنوي او د UTL\_CALL\_STACK پکيچ معرفي کوي چې د اجرايي stack تېروتنې stack او backtrace ته دقيق لاسرسی ورکوي

د Oracle Database 12c د PL/SQL نوې ځانگړنې

Oracle Database 12c د PL/SQL فعاليت او کارونې اسانتيا ډېره کړې او د ژبې ځينې نيمگړتياوې يې اصلاح کړې دلته د PL/SQL پراختياکوونکو لپاره مهم بدلونونه لنډيز شوي دي



-د PL/SQL ځانگړي ډولونه د SQL سره يوځای شوي: مخکې له [12.1](#) تاسو نشو کولای  
PL/SQL ځانگړي (datatype لکه associative array د dynamic SQL بيان کې  
وکاروئ اوس دا ممکنه ده چې دا ډول ارزښتونه د PL/SQL anonymous blocks  
فکشنونو CALL بيانونو او SQL پوښتنو کې bind شي

ACCESSIBLE\_BY - کلیمه: تاسو اوس کولای شئ دا کلیمه د پکیج specification کې  
وکاروئ خو مشخص کړئ چې کوم پروگرام واحدونه کولای شي د پکیج  
subprogram ونه وغږوي دا ځانگړنه د "سپین لیست (whitelisting)" په څېر عمل کوي

Implicit statement results: مخکې له Oracle 12c ، PL/SQL subprogram ونه  
باید د SQL پوښتنو پایلې د OUT REF CURSOR يا RETURN له لارې ښکاره کړي اوس  
subprogram ونه کولای شي پایلې په غیر مستقیم ډول د DBMS\_SQL پکیج له لارې  
ورکړي چې د نورو ډیټابېسونو (لکه Transact SQL څخه Oracle ته د اپلیکېشنونو انتقال  
اسانه کوي

views: BEQUEATH CURRENT\_USER - مخکې له Oracle 12c view تڼل د definer  
rights واحد په توگه عمل کاوه حتی که د invoker rights واحد دننه اشاره شوې وه اوس  
view کولای شي (BEQUEATH DEFINER اصلي حالت) يا BEQUEATH  
CURRENT\_USER وي چې د invoker rights واحد په څېر عمل کوي

-د رولونو ورکول پروگرام واحدونو ته: مخکې له Oracle 12c invoker rights واحد تڼل  
د اجرا کوونکي واک سره چلېده که اجرا کوونکي له مالک څخه ډېر واک درلود نو دا واحد  
ممکن ناڅاپي يا منع شوي عملیات ترسره کړي له [12.1](#) وروسته، تاسو کولای شئ رولونه  
ځانگړو PL/SQL پکیجونو يا subprogram ونو ته ورکړئ دا واحدونه اوس د اجرا  
کوونکي او رولونو واک سره چلېږي خو د تعريف کوونکي اضافي واک نه لري دا واحد  
یوازې هغه وخت د اجرا کوونکي واک سره چلېدای شي که مالک يې د INHERIT  
PRIVILEGES يا INHERIT ANY PRIVILEGES و