

# PL / SQL के आधार बिन्दु

---

## पाठ्यपुस्तक के प्रश्न

### वस्तुनिष्ठ प्रश्न

**प्रश्न 1. जो एक PL/SQL का हिस्सा नहीं है**

- (अ) Declare
- (ब) BEGIN
- (स) Start
- (द) End

**उत्तर:** (स) Start

**प्रश्न 2. PL/SQL द्वारा विकसित की है**

- (अ) IBM
- (ब) ORACLE
- (स) Microsoft
- (द) इनमें से कोई नहीं

**उत्तर:** (ब) ORACLE

**प्रश्न 3. शब्द का चयन करें जो Select कथन के साथ उपयोग किया जाना चाहिए**

- (अ) Goto
- (ब) Into
- (स) Do
- (द) all

**उत्तर:** (ब) Into

**प्रश्न 4. कर्सर कितने प्रकार के हैं?**

- (अ) 2
- (ब) 4
- (स) 5
- (द) 1

**उत्तर:** (अ) 2

### **प्रश्न 5. % FOUND का काम विपरीत है**

- (अ) % CURSOR
- (ब) % NOT COUNT
- (स) % NOT FOUND
- (द) % FOUND COUNT

**उत्तर:** (स) % NOT FOUND

## **अतिलघु उत्तरीय प्रश्न**

### **प्रश्न 1. PL/SQL क्या है?**

**उत्तर-** PL/SQL का अर्थ प्रसिजरल लेंग्वेज के एक्सटेंशन से है। इसके महत्वपूर्ण बिन्दु SQL पर आधारित हैं। इसका निर्माण ऑरेकल कम्पनी ने किया है।

### **प्रश्न 2. PL/SQL ब्लॉक में कितने भाग हैं?**

**उत्तर-** PL/SQL ब्लॉक में तीन भाग हैं

- घोषणा (Declaration) भाग
- लागूकरण (Execution) भाग
- अपवाद संचालन (Exception handling) भाग

### **प्रश्न 3. PL/SQL में Declare का उपयोग क्यों करते हैं?**

**उत्तर-** PL/SQL में Declare का उपयोग Constants, records, variables और Cursors को घोषित करने में होता है।

### **प्रश्न 4. PL/SQL में & का उपयोग क्या है?**

**उत्तर-** PL/SQL में & का उपयोग किसी Variable के लिए User से इनपुट कराने के लिए होता है।

### **प्रश्न 5. PL/SQL में Variables कहाँ घोषित किए जाते हैं?**

**उत्तर-** PL/SQL में Variables घोषणा (Declaration) भाग में घोषित किये जाते हैं।

### **प्रश्न 6. PL/SQL select कथन का प्रयोग कैसे किया जाता है?**

**उत्तर-** Syntax SELECT Column name INTO variable name FROM table\_name [where condition];

### **प्रश्न 7. Exception ब्लॉक का क्या उपयोग है?**

**उत्तर-** PL/SQL खण्ड का वह भाग जो कि गैर-जरूरी है, आरक्षित Keyword EXCEPTION से शुरू होता है और प्रोग्राम में त्रुटियों के संचालन के उपयोग में लाया जाता है।

### **प्रश्न 8. PL/SQL में Variable के प्रकार को बताइए।**

**उत्तर-** Variable एक storage area (क्षेत्र) को दिया हुआ नाम होता है जिसे हमारा प्रोग्राम मनीपुलेट (manipulate) करता है। प्रत्येक वेरिएबल का एक खास (specific) डेटा टाइप होता है; जैसे-Number, char, varchar, date, long. घोषणा के आधार पर variable दो प्रकार के होते हैं

- Local Variables : ये variables एक inner block में घोषित (declared) किए जाते हैं और Outside block से Referenced नहीं किए जा सकते हैं।
- Global Variables : ये Variables outer block में घोषित (declared) किए जाते हैं और ये Inner तथा Outer block दोनों के द्वारा Referenced किए जा सकते हैं।

### **प्रश्न 9. ट्रिगर क्या है?**

**उत्तर-** ट्रिगर (Trigger) PL/SQL की वह संरचना है जिसका उपयोग DML वाक्य : जैसे Insert, Delete, Update के डेटाबेस तालिका पर Execution के समय किया जाता है। यह Automatically प्रयुक्त होता है जब उपरोक्त DML वाक्यों का Execution होता है।

### **प्रश्न 10. ट्रिगर का उपयोग कैसे करते हैं?**

**उत्तर-**

- ट्रिगर के उपयोग CREATE [OR REPLACE] TRIGGER name\_of\_trigger\_In PL/SQL में ट्रिगर जिसका नाम दिया गया हो, के निर्माण हेतु या फिर ट्रिगर जो अस्तित्व में है, को ओवरराइट करने हेतु इस क्लॉज का उपयोग होता है।
- {BEFORE | AFTER | INSTEAD OF} यह क्लॉज दर्शाता है कि ट्रिगर का उपयोग किस समय किया जाना चाहिए उदाहरणतः टेबल के UPDATE के पहले और पश्चात् INSTEAD OF का उपयोग View के समय ट्रिगर के निर्माण के लिए किया जाता है। Before और After का उपयोग View के समय ट्रिगर निर्माण में नहीं किया जा सकता है।

- {INSERT [OR] | UPDATE [OR] | DELETE} यह क्लॉज ट्रिगरिंग की घटना (event) को बताता है। एक से अधिक ट्रिगरिंग की घटनाओं के एक साथ होने की स्थिति में घटनाओं को पृथक् करने हेतु Keyword का उपयोग होता है। सारी ट्रिगरिंग की घटनाओं के समय ट्रिगर उपयोग होता है।

## लघु उत्तरीय प्रश्न

### प्रश्न 1. % TYPE गुण और % ROWTYPE गुण प्रकार के बीच क्या अन्तर है?

**उत्तर-** % TYPE और % ROWTYPE दोनों डेटा स्वतन्त्रता (date independence) प्रदान करते हैं और प्रोग्राम को नए व्यावसायिक आवश्यकता के कारण डेटाबेस में परिवर्तन करने की अनुमति मिलती है। % TYPE उसी Data type का Variable घोषित करने में उपयोग है जैसे कि टेबल में किसी Column का Data type  
% ROWTYPE को एक प्रकार के साथ एक रिकॉर्ड वैरिएबल घोषित करने के लिए उपयोग किया जाता है।

### प्रश्न 2. PL/SQL में EXIT कथन का क्या उपयोग है?

**उत्तर-** PL/SQL में किसी लूप के अन्दर EXIT Statement के आने पर लूप को तुरन्त समाप्त कर दिया जाता है और लूप के बाद अगले नियन्त्रण में प्रोग्राम नियन्त्रण शुरू होता है।

यदि नेस्टेड लूप का प्रयोग कर रहे हैं तो EXIT Statement, innermost लूप के निष्पादन (execution) को रोक देगा और ब्लॉक के बाद कोड की अगली पंक्ति को निष्पादित (executing) करना शुरू करेगा।

Syntax:

EXIT;

### प्रश्न 3. Before ट्रिगर क्या है?

**उत्तर-** ट्रिगर PL/SQL खण्ड की संरचना है जिसका उपयोग DML वाक्य जैसे Insert, Delete, Update के डेटाबेस तालिका पर Execution के समय किया जाता है। यह Automatically प्रयुक्त होता है जब उपरोक्त DML वाक्यों का execution होता है।

BEFORE ट्रिगर दर्शाता है कि ट्रिगर उपयोग पहले किया जाएगा। उदाहरणतः किसी टेबल के UPDATE के पहले।

### प्रश्न 4. Implicit और Explicit कर्सर के बीच क्या अन्तर है?

**उत्तर-** PL/SQL में कर्सर दो प्रकार के होते हैं  
अंतर्निहित (Implicit) कर्सर – यह पहले से बने होते हैं जब किसी DML वाक्य जैसे INSERT, UPDATE

और DELETE का लागूकरण किया जाता है। किसी SELECT वाक्य जो केवल एक पंक्ति परिणामस्वरूप देता है इसके लिए भी कर्सर उपयोग में जाते हैं।

बाह्यनिहित (Explicit) कर्सर – इनका निर्माण जरूरी है जब एक SELECT वाक्य का लागूकरण होता है एवं ये वाक्य परिणाम स्वरूप एक से ज्यादा पंक्तियों (more than one) को प्रदर्शित करता है तो यह कर्सर एक से ज्यादा तथ्य संग्रहित करता है परन्तु एक समय में एक ही तथ्य का क्रियान्वन संभव है। जिसे वर्तमानकालीन (current) पंक्ति कहा जाता है। जब कभी कोई नयी पंक्ति आती है। तब पुरानी पंक्ति वर्तमान स्थिति से अगली स्थिति में आ जाती है। अंतर्निहित एवं बाह्यनिहित कर्सर की कार्य प्रणाली (functionality) एक समान होती है। परन्तु क्रियान्वन भिन्न हैं।

### प्रश्न 5. For Loop का सिंटैक्स लिखें।

**उत्तर-** फॉर लूप (FOR LOOP)-FOR LOOP के भीतर वाक्यों का तब तक execution होता है जब तक लूप में लिखी संख्या पूरी नहीं हो जाती है। लूप का चलन प्रारम्भिक से अन्तिम तक (integer values given) होता है। काउन्टर के

मूल्य में प्रत्येक बार एक संख्या से वृद्धि होना निश्चित है। अन्तिम मूल्य पर पहुँचने पर काउन्टर लूप से बाहर निकल जाता है।

फॉर लूप का साधारण सिन्टेक्स – FOR counter IN from.....to  
LOOP statements;  
ENDLOOP;

### निबंधात्मक प्रश्न

**प्रश्न 1. कर्सर क्या है? कर्सर का उपयोग क्या है? उदाहरण के साथ explicit कर्सर की व्याख्या करें।**

**उत्तर-** PL/SQL कर्सर-जब कभी किसी SQL वाक्य का लागूकरण होता है। एक अस्थायी work area मशीन Memory में बनता है। जिसे कर्सर कहा जाता है। आँकड़ों की पंक्तियों एवं सलेक्ट वाक्य से सम्बन्धित सूचनायें कर्सर में होती हैं।

उपयोग – यह अस्थायी कार्य क्षेत्र उन आँकड़ों को संग्रहित करता है जो डाटाबेस से ग्रहण किये जाते हैं। कर्सर एक से ज्यादा पंक्तियों पर नियंत्रण रख सकता है। परन्तु एक समय में एक पंक्ति का ही क्रियान्वन संभव है। वो सारी पंक्तियों का समूह जिस पर कर्सर का नियंत्रण होता है क्रियाशील समूह (active set) कहलाता है।

Explicit कर्सर – Explicit कर्सर का वर्णन 'PL/SQL खण्ड के घोषणा भाग में दिया जाता है इसका निर्माण उस SELECT वाक्य के संदर्भ में किया जाता है जो परिणामस्वरूप एक से अधिक पंक्तियों को प्रदर्शित करता है कर्सर को इच्छानुसार उपयुक्त नाम दे सकते हैं।

कर्सर के निर्माण के लिए साधारण रचना

CURSOR cursor\_name IS select\_statement;

- cursor\_name – कर्सर का नाम
- select\_statement – एक सलेक्ट query जो एक या एक से अधिक पंक्तियाँ प्रदर्शित करे।

Explicit कर्सर के उपयोग हेतु बिन्दु –

- घोषणा भाग में कर्सर को घोषित करें।
- Execution भाग में कर्सर को सक्रिय (OPEN) करें।
- Execution भाग में PL/SQL वेरिएबल्स या तथ्यों में कर्सर से आँकड़ें लावें।
- PL/SQL खण्ड की समाप्ति से पहले कर्सर को निष्क्रिय (CLOSE) करें।

**उदाहरण –**

1. SQL> set serveroutput on
2. SQL> edit explicit\_cursor
3. DECLARE
4. Cursor c is select \* from emp\_information
5. Where emp\_name = 'Kamal';
6. tmp emp\_information % rowtype;
7. BEGIN
8. OPENC;
9. Loop exit when C% NOT FOUND;
10. FETCHC into tmp;
11. update emp\_information set tmp..emp\_dept = 'webDeveloper'
12. Where tmp. emp\_name= 'Sagar';
13. END Loop;
14. IF C% ROWCOUNT > 0 THEN
15. dbms\_output. put\_line (SQL % ROWCOUNT || 'Rows Updated');
16. ELSE .
17. dbms output. put\_line ('No Rows Updated Found');
18. END IF;
19. CLOSEC;
20. END;
21. /

**Result :**

SQL> @explicit\_cursor

1 Rows updated

## प्रश्न 2. विभिन्न प्रकार के डेटाबेस ट्रिगर्स उदाहरण के साथ समझाइए।

**उत्तर-** डेटाबेस ट्रिगर्स के प्रकार

1. BEFORE Trigger – BEFORE ट्रिगर, ट्रिगरिंग DML कथन (INSERT, UPDATE, DELETE) के निष्पादन (execute) से पहले ही निष्पादित हो जाते हैं। ट्रिगरिंग SQL कथन निष्पादित होंगे या नहीं, यह BEFORE ट्रिगर condition block पर निर्भर करता है।
2. AFTER Trigger – AFTER ट्रिगर, ट्रिगरिंग DML कथन (INSERT, UPDATE, DELETE) के निष्पादन (execute) के बाद ही निष्पादित होते हैं। ट्रिगरिंग SQL कथन ट्रिगर के कोड के आते ही निष्पादित हो जाते हैं।
3. ROW Trigger – ROW ट्रिगर, प्रत्येक रिकॉर्ड के लिए जो कि डेटाबेस टेबल के लिए INSERT, UPDATE, DELETE को perform कर रहे हैं, तुरन्त कार्य करता है।
4. Statement Trigger – Statement ट्रिगर एक कथन पर सिर्फ एक ही बार कार्य करता है।
5. Combination Trigger – Combination ट्रिगर दो ट्रिगर type का मेल होता है। ये निम्न प्रकार के होते हैं
  - Before statement Trigger
  - Before Row Trigger
  - After statement Trigger
  - After Row Trigger

## प्रश्न 3. Exception क्या है? विभिन्न प्रकार की exceptions की व्याख्या करें।

**उत्तर-** Exception – किसी प्रोग्राम के निष्पादन (execution) के दौरान एक error condition को ही Exception कहते हैं। PL/SQL प्रोग्रामर को प्रोग्राम में Exception Block का उपयोग कर ऐसी conditions को पकड़ने और उस error condition के विरुद्ध उचित कारवाई (action) करने को support करता है।

Exception के प्रकार – तीन प्रकार के Exceptions होते हैं।

1. Named System Exceptions
2. Unnamed System Exceptions
3. User-defined Exceptions

### 1. Named System Exceptions

तंत्र के Exceptions automatically ऑरेकल द्वारा चिह्नित किये जाते हैं यदि कोई प्रोग्राम RDBMS के नियम की पालना नहीं करता है और कुछ ऐसे अपवाद बार-बार विघ्न डालते हैं जो कि पूर्व वर्णित एवं ऑरेकल में नामांकित होते हैं तो ऐसे अपवाद नामांकित तंत्र अपवाद कहलाते हैं।

उदाहरण – NO\_DATA FOUND and ZERO\_DIVIDE Named System Exceptions है।  
Named System Exceptions:

- बाह्यनिहित (explicitly) घोषित नहीं होते हैं।
- अंतर्निहित (implicitly) रूप से जाग्रत होते हैं जब कभी ऑरेकल की पूर्व वर्णित त्रुटियाँ होती हैं।
- Exceptions Handling रूटीन में सामान्य नाम के हवाले से चिह्नित किये जाते हैं।

उदाहरण – विचार कीजिए यदि NO DATA\_FOUND Exceptions किसी कार्यपद्धति में जाग्रत होता है। हम Exceptions Handling हेतु निम्नांकित कोड लिख सकते हैं।

```
BEGIN
```

```
Execution part
```

```
EXCEPTION
```

```
WHEN NO DATA FOUND THEN
```

```
dbms_output.put_line ('Using SELECT....INTO did not get anyrow.');
```

```
END;
```

## 2. Unnamed System Exceptions

वो सारे तंत्र Exception जिनके लिए ऑरेकल की तरफ से नाम नहीं दिया जाता है Unnamed System Exceptions कहलाते हैं। ये Exceptions नियमित रूप से जाग्रत नहीं होते हैं इन Exceptions के साथ एक प्रोग्राम और एक सन्देश होता है। Unnamed System Exceptions दो प्रकार से होता है

- WHEN OTHERS Exceptions handler का उपयोग कर या
- Exception कोड के नाम देकर named System Exceptions की तरह उपयोग करें।  
EXCEPTION\_INIT एक पूर्वनिर्धारित Oracle त्रुटि संख्या उपयोग कर हम प्रोग्राम नामक Exception को नाम दे सकते हैं।

Unnamed System Exceptions को उपयोग करते समय कुछ ध्यान रखने योग्य मुख्य बिन्दु इस प्रकार है

- ये अंतर्निहित रूप से जाग्रत होते हैं।
- दूसरों के साथ संचालन न होने की स्थिति में ये बाह्यनिहित रूप से संचालित होते हैं।
- Exceptions को बाह्यनिहित रूप से संचालन हेतु उनकी घोषणा Pragma EXCEPTION\_INIT के उपयोग से हो एवं संचालन उपयोगकर्ता वर्णित Exception के नाम से Exception भाग में होना चाहिए।

EXCEPTION\_INIT का उपयोग करते हुए Unnamed System Exceptions की घोषणा हेतु साधारण रचना :

```
DECLARE
```

```
excep_name EXCEPTION;
```



```
PRAGMA
EXCEPTION_INIT (excep_name, Err_code);
BEGIN
Execution part
EXCEPTION
WHEN excep name THEN
handle the exception
END;
```

### 3. User-defined Exceptions

System Exceptions के अलावा हम उन exceptions को जो व्यवसायिक नियम पर आधारित है कि बाह्यनिहित रूप से वर्णित कर सकते हैं और इन्हें user-defined exceptions कहते हैं। User-defined exceptions के उपयोग हेतु मुख्य बिन्दु

- इनकी घोषणा बाह्यनिहित रूप से जाग्रत किया जायें।
- user-defined exceptions के नाम का हवाला देते हुए exception खण्ड में संचालन किया जायें।

### प्रश्न 4. PL/SQL में लूप के विभिन्न प्रकार समझाइए।

**उत्तर-** जब हम किसी वाक्य या वाक्यों का Execution एक से अधिक बार करना चाहते हैं तो हम पुनरावृत्ति नियंत्रक वाक्य (iterative control statements) का उपयोग करते हैं। PL/SQL में तीन तरह के लूप होते हैं।

**लूप के प्रकार व रचना इस प्रकार है –**

- साधारण लूप
- व्हाइल लूप
- फॉर लूप

साधारण लूप की रचना –

```
LOOP
Statements;
EXIT;
{or EXIT WHEN condition;}
ENDLOOP;
```

साधारण लूप का उपयोग करते समय कुछ महत्वपूर्ण बिन्दु पर ध्यान दें-

- सदैव लूप के प्रधान भाग से पहले वेरिएबल्स का मूल्यांकित (initialize) करें।
- लूप के भीतर वेरिएबल के मूल्य में इजाफा जरूर करना चाहिए।

- यदि लूप से बाहर निकलना चाहें तो EXIT WHEN वाक्य लिखें। यदि EXIT, WHEN के साथ नहीं लिखा जाता है तो लूप का execution केवल एक बार होता है।

व्हाइल लूप (While Loop) – While Loop के भीतर लिखे सारे वाक्यों का execution तब तक होता है जब तक condition true है। अवस्था का मूल्यांकन, प्रत्येक बार तब तक लूप चलता है के साथ होता है। और तब तक होता है जब तक condition false नहीं होती है।

While Loop लूप का साधारण सिन्टेक्स –

```
WHILE < condition>
LOOP statements;
END LOOP;
```

फॉर लूप (FOR LOOP) – FOR LOOP के भीतर वाक्यों का तब तक execution होता है जब तक लूप में लिखी संख्या पूरी नहीं हो जाती है। लूप का चलन प्रारम्भिक से अन्तिम तक (integer values given) होता है। काउन्टर के मूल्य में प्रत्येक बार एक संख्या से वृद्धि होना निश्चित है। अन्तिम मूल्य पर पहुँचने पर काउन्टर लूप से बाहर निकल जाता है।

फॉर लूप का साधारण सिन्टेक्स

```
FOR counter IN from. . .to
LOOP statements;
END LOOP;
```

- From-Start integer value.
- to-End integer value.

फॉर लूप के execution के लिए कुछ महत्वपूर्ण उपाय-

- काउन्टर वेरियबल घोषणा खण्ड में ही घोषित किया जाता है। घोषणा खण्ड के बाहर घोषित करना अनावश्यक है।
- काउन्टर वेरियबल 1 के द्वारा Incremented है अनावश्यक वृद्धि करने की जरूरत नहीं है।
- EXIT WHEN वाक्य और EXIT वाक्य फॉर (for) लूप के भीतर उपयोग में लाये जा सकते हैं परन्तु यह अनुचित है।

**प्रश्न 5. Function क्या है? यह Procedure से अलग कैसे है? functions और Procedure के लिए सिन्टेक्स समझाइए।**

**उत्तर-** PL/SQL फंक्शन-यह एक तरह की नामांकित PL/SQL Block है जो कि procedure के समान ही है। Function एवं Procedure में एक बड़ा अन्तर यह है कि फंक्शन में सदैव एक मान दिया जाता है पर procedure में यह हो भी सकता है और नहीं भी।

फंक्शन की साधारण रचना –

```
CREATE [OR REPLACE]FUNCTION func_name [parameters list]
RETURN return_datatype
IS
Declaration_part
BEGIN
Execution_part
Return return_variable;
EXCEPTION
exception_part
Return return_variable;
END;
```

- Return Type हैडर वाला भाग प्रक्रिया के रिटर्न टाईप को वर्णित करता है। रिटर्न डाटा टाईप कोई भी उपयुक्त डाटा टाईप हो सकता है। जैसे varchar, number etc.
- Execution भाग एवं exception भाग हैडर में वर्णित डाटा टाईप का ही मान वापस देते हैं।

उदाहरण – हम एक फंक्शन का निर्माण करते हैं जिसका नाम 'student\_info\_func' है।

1. >CREATE OR REPLACE FUNCTION student\_info\_func
2. > RETURN VARCHAR(10);
3. >IS
4. > stu name VARCHAR (20);
5. >BEGIN
6. > SELECT Sname INTO stu\_name
7. > FROM student WHERE Roll\_no = '58';
8. >RETURN stu\_name;
9. >END;
10. >/

उपरोक्त उदाहरण में वो 'Sname' प्राप्त किये जिनका अनुक्रमांक 58 है और उन्हें 'stu\_name' वेरियबल में संग्रहित किया। प्रक्रिया के रिटर्न टाईप का प्रकार VARCHAR है जिसकी घोषणा 2 नम्बर पंक्ति में की गई है। पूरी फंक्शन 'stu name' पंक्ति नम्बर 9 में परिणामस्वरूप देती है। जिसके आँकड़े का प्रकार VARCHAR है।

Executing फंक्श –

- क्योंकि फंक्शन परिणामस्वरूप एक मान देती है। यह वेरिएबल को सौंप सकते हैं।  
student\_name : = student\_info\_func;

यदि 'student\_name' के आँकड़ों का प्रकार VARCHAR हो तो हम विद्यार्थी का नाम फंक्शन के रिटर्न टाईप को इसे सौंप कर संग्रहित कर सकते हैं।

- SELECT वाक्य के एक अंश की तरह  
SELECT student\_info\_func FROM student;
- PL/SQL के कुछ वाक्यों में  
dbms\_output.put\_line (student\_info\_func);  
यह पंक्ति फंक्शन द्वारा रिटर्न मान को प्रदर्शित करती है।

Function व Procedure में मुख्य अन्तर – फंक्शन को एक मान वापिस करना चाहिए परन्तु Procedure में यह वैकल्पिक है। फंक्शन में केवल इनपुट पैरामीटर हो सकते हैं। जबकि Procedure में इनपुट/आउटपुट पैरामीटर हो सकते हैं। हम Procedure से फंक्शन को call कर सकते हैं जबकि फंक्शन से Procedure को call नहीं किया जा सकता।

#### Procedure

संग्रहित कार्य पद्धति (Stored Procedures) नामांकित PL/SQL खण्ड जो एक या एक से अधिक कार्यों का क्रियान्वन करे संग्रहित Procedures या साधारण Procedures कहलाता है। Procedures में एक हैडर व बॉडी होता है। जहाँ हैडर में Procedures का नाम और पैरामीटर या वेरिएबल्स का नाम होता है जो कि Procedures को पास किये जाते हैं। बॉडी में एक declaration भाग एवं एक exception भाग होता है। जो कि साधारण PL/SQL खण्ड के समान ही है। परन्तु एक से अधिक बार उपयोग में लाने हेतु Procedures को नामांकित किया जाता है।

Procedures में पैरामीटर को पास करना – तीन प्रकार से पैरामीटर को Procedures में पास कर सकते हैं।

- IN-parameters
- OUT-parameters
- IN OUT-parameters

Procedures के क्रियान्वन हेतु साधारण रचना –

```
CREATE [ORREPLACE] PROCEDURE proce_name [parameterlist]
```

```
IS
```

```
Declaration section
```

```
BEGIN
```

```
Exception section
```

```
EXCEPTION
```

```
Exception section
```

```
END;
```

IS : Procedures की बॉडी के प्रारम्भ को चिह्नित करता है एवं PL/SQL के घोषणा भाग के समान ही है। IS एवं BEGIN के मध्य लिखा हुआ भाग ही घोषणा भाग कहलाता है।

जो भाग बड़े कोष्ठक [] में लिखा जाता है गैर जरूरी होता है। हम CREATE या REPLACE को एक साथ तभी प्रयुक्त करें जब समान नाम से कोई और procedures न हो या अस्तित्व में जो procedures है वर्तमान कोड से बदली जाये।

## अन्य महत्वपूर्ण प्रश्न

### अतिलघु उत्तरीय प्रश्न

**प्रश्न 1. PL/SQL का निर्माण किस कम्पनी ने किया है?**

**उत्तर-** PL/SQL का निर्माण ऑरेकल कम्पनी ने किया है।

**प्रश्न 2. PL/SQL ब्लॉक क्या होता है?**

**उत्तर-** यह मुख्यतः SQL और PL/SQL स्टेटमेन्ट्स से मिलकर बनता है। PL/SQL प्रोग्राम में इनका उपयोग होता है।

**प्रश्न 3. Exception भाग क्या होता है?**

**उत्तर-** PL/SQL खण्ड का वह भाग जो कि गैर जरूरी है, आरक्षित Keyword EXCEPTION से शुरू होता है और प्रोग्राम में त्रुटियों के संचालन में उपयोग में लाया जाता है।

**प्रश्न 4. प्लेसहोल्डर का प्रमुख कार्य क्या होता है?**

**उत्तर-** प्लेसहोल्डर आँकड़ों को अस्थायी रूप से संग्रहित करता है।

**प्रश्न 5. Execution भाग के भागों के नाम बताइए।**

**उत्तर-** प्रोग्रामेटिक Construct Loops, सशर्त कथन और SQL कथन Execution भाग के विभिन्न भागों के नाम हैं।

**प्रश्न 6. नेस्टिंग किसे कहते हैं?**

**उत्तर-** PL/SQL में, प्रोग्राम के खण्ड एक-दूसरे के साथ लिखे जा सकते हैं जिसे नेस्टिंग कहते हैं। प्रत्येक खण्ड । सम्पूर्ण कार्य के छोटे या बड़े अंश के लिए उत्तरदायी होता है।

### **प्रश्न 7. PL/SQL वेरिएबल्स क्या होते हैं?**

**उत्तर-** PL/SQL वेरियबल्स वो प्लेसहोल्डर्स हैं जो वैल्यूस को संग्रहित करते हैं तथा जो PL/SQL खण्ड के जरिए बदली जा सकती है।

### **प्रश्न 8. कॉन्स्टेन्ट से आप क्या समझते हैं?**

**उत्तर-** यदि PL/SQL खण्ड में प्रयुक्त Value पूरे प्रोग्राम में परिवर्तित न हो तो इस तरह का Value कॉन्स्टेन्ट कहलाता

### **प्रश्न 9. VALUE से आप क्या समझते हैं?**

**उत्तर-** VALUE वह मूल्य या मूल्यांकन है जो घोषित होते समय किसी भी कॉन्स्टेन्ट को दिया जाता है।

### **प्रश्न 10. Exception Handling किसे कहते हैं?**

**उत्तर-** PL/SQL की एक सुविधा यह है कि यह PL/SQL खण्ड में आये Exception की Handling करता है, जिसे Exception handling कहा जाता है। का उपयोग कर हम वर्तमान प्रोग्राम या कोड की जाँच कर सकते हैं।

### **प्रश्न 11. Exception कितने प्रकार के होते हैं? नाम बताइए।**

**उत्तर-** Exception तीन प्रकार के होते हैं

- Named System Exception
- Unnamed System Exception
- User-defined Exception

### **प्रश्न 12. Unnamed System Exceptions किसे कहते हैं?**

**उत्तर-** वे सारे तंत्र Exceptions जिनके लिए ऑरेकल की ओर से नाम नहीं दिया जाता है Unnamed System Exceptions कहलाते हैं। ये Exceptions नियमित रूप से जाग्रत नहीं होते हैं, इन Exceptions के साथ एक प्रोग्राम और सन्देश होता है।

**प्रश्न 13. डाटाबेस ट्रिगर के कितने अंश होते हैं? नाम बताइए।**

**उत्तर-** डाटाबेस ट्रिगर के तीन अंश होते हैं

- ट्रिगरिंग इवेन्ट
- शर्त (condition)
- गतिविधि (action)

## **लघु उत्तरीय प्रश्न**

**प्रश्न 1. PL/SQL के Declaration भाग पर संक्षिप्त टिप्पणी लिखिए।**

**उत्तर-** PL/SQL Declaration भाग-खण्ड का यह भाग जो कि गैर जरूरी है DECLARE नाम के Keyword जो कि आरक्षित है, से शुरू होता है। इस खण्ड का उद्देश्य Constants, records, variables और Cursors को घोषित करना है। उपरोक्त सभी विचारार्थ शब्द (प्लेसहोल्डर) Execution भाग में आँकड़ों का हेरफेर करने के लिए उपयोग में लिए जाते हैं। प्लेसहोल्डर (constants, variable और records) आँकड़ों को अस्थायी रूप से संग्रहित करते हैं। कर्सर भी इस खण्ड में घोषित किए गए हैं।

**प्रश्न 2. PL/SQL के Execution भाग पर संक्षिप्त टिप्पणी लिखिए।**

**उत्तर-** Execution भाग-PL/SQL (block) का यह भाग आरक्षित keyword BEGIN और END से शुरू और खत्म होता है। Execution भाग किसी कार्य को सफलतापूर्वक पूर्ण करने के लिए वहाँ जरूरी भाग है जहाँ प्रोग्राम का तथ्य लिखा जाता है। प्रोग्रामेटिक constructs Loops, सशर्त कथन और SQL कथन Execution भाग के भाग हैं।

**प्रश्न 3. PL/SQL खण्ड की बनावट उदाहरण सहित समझाइये।**

**उत्तर-** PL/SQL खण्ड की बनावट (Structure of PL/SQL Block)

```
DECLARE  
Variable declaration  
BEGIN  
Program Execution  
Exception  
Exception handling  
END;
```

**उदाहरण-**

साधारण PL/SQL प्रोग्राम का खण्ड (block) जो "भारत" शब्द को प्रदर्शित करता है।

```
SQL> set serveroutput on
```

1. SQL> begin
2. dbms\_output.put\_line ('BHARAT');
3. end;
4. /

BHARAT

#### **प्रश्न 4. PL/SQL प्रोग्राम के महत्वपूर्ण बिन्दु बताइए।**

**उत्तर-** PL/SQL प्रोग्राम के कुछ महत्वपूर्ण बिन्दु निम्न प्रकार हैं

- PL/SQL प्रोग्राम खण्ड (block) का भाग keyword Begin से शुरू होता है एवं keyword END से समाप्त किया जाता है जो कि PL/SQL खण्ड का Executable भाग कहलाता है।
- PL/SQL प्रोग्राम खण्ड कथनों से मिलकर बनता है। प्रत्येक कथन सेमीकॉलन (;) से समाप्त होता है।
- PL/SQL प्रोग्राम खण्ड के अन्त में फॉरवर्ड स्लैश (/) लगाया जाता है जो कि प्रोग्राम खण्ड के कथनों के लागूकरण के लिए उत्तरदायी है।

#### **प्रश्न 5. PL/SQL खण्ड के प्रमुख लाभ बताइए।**

**उत्तर-** PL/SQL के लाभ-PL/SQL खण्ड के कुछ लाभ निम्न प्रकार हैं –

PL/SQL खण्ड की बनावट – PL/SQL में, प्रोग्राम के खण्ड एक-दूसरे के साथ लिखे जा सकते हैं जिसे नेस्टिंग कहते हैं। प्रत्येक खण्ड सम्पूर्ण कार्य के छोटे या बड़े अंश के लिए उत्तरदायी होता है। PL/SQL खण्ड डाटाबेस में संग्रहित किये जाते हैं और पुनः उपयोग में भी लाये जाते हैं।

बेहतर प्रदर्शन – PL/SQL इंजन एकसाथ कई SQL कथनों को एक खण्ड की तरह संचालित करता है। जो बेहतर प्रदर्शन और कम नेटवर्क traffic के लिए उत्तरदायी है।

पद्धतिकरण (Procedural) भाषा की क्षमता – PL/SQL को और शक्तिशाली बनाने हेतु कुछ procedural language कन्स्ट्रक्ट का उपयोग होता है; जैसे-(if else) और (for loops).

#### **प्रश्न 6. PL/SQL के प्लेसहोल्डर्स के विषय में बताइए।**

**उत्तर-** PL/SQL प्लेसहोल्डर्स-प्लेसहोल्डर्स आँकड़ों को अस्थायी रूप से संग्रहित करने के लिए जगह देते हैं जो कि PL/SQL खण्ड के execution के दौरान हेरफेर करने के लिए उपयोग किए जाते हैं। वेरिएबल्स कॉन्स्टेन्ट और रिकॉर्ड्स PL/SQL प्लेसहोल्डर्स हैं।



प्लेस होल्डर्स को नाम और datatype से वर्णित किया जाता है यह परिभाषा उन आँकड़ों पर निर्भर करती है जो हम संग्रहित करते हैं। कुछ आँकड़ों के प्रकार (datatype) जो प्लेस होल्डर्स को वर्णित करते हैं निम्नलिखित हैं

Number (n, m), Char (n), Varchar2 (n), Date, Long, Long raw, Raw, Blob, Clob, Nclob and Bfile.

PL/SQL वेरिएबल्स-ये वो होल्डर्स हैं जो वेल्यू को संग्रहित करते हैं जो PL/SQL खण्ड के जरिए बदली जा सकती है।

### **प्रश्न 7. PL/SQL में वेरिएबल्स को घोषित करने की सिन्टेक्स बताइए तथा उदाहरण भी दीजिए।**

**उत्तर-** वेरिएबल्स को घोषित (declare) करने की साधारण सिन्टेक्स  
variable\_name datatype [NOT NULL : = value];

variable\_name is the name of variable. डाटा टाईप यहाँ एक उचित PL/SQL डेटाटाईप है। NOT NULL वेरिएबल्स के लिए एक गैर जरूरी व्याख्या है। Value एवं डिफाल्ट वेल्यू भी एक गैर जरूरी व्याख्या है जहाँ पर आप वेरिएबल्स को इनिशियलाइज करते हैं। प्रत्येक वेरिएबल्स की घोषणा एक अलग वाक्य है जो कि सेमीकॉलन से समाप्त होता है।

#### **उदाहरण-**

यदि आप एक शिक्षक का वर्तमान वेतन संग्रहित करना चाहते हैं तो वेरियबल्स उपयोग में लायेंगे।

DECLARE

salary\_variable number (6);

“salary\_variable” एक वेरिएबल है जिसके datatype नम्बर हैं और जिसकी 6 लम्बाई है। जब कभी वेरिएबल्स का ब्यौरा NOT NULL से दिया जाता है। उस वेरिएबल को Initialize करें जब उसे घोषित कर दिया जाता है।

### **प्रश्न 8. PL/SOL कॉन्स्टेन्ट क्या होता है? इसका उपयोग व सिन्टेक्स बताइए।**

**उत्तर-** PL/SQL कॉन्स्टेन्ट-यदि PL/SQL खण्ड में प्रयुक्त Value पूरे प्रोग्राम में परिवर्तित न हो तो इस तरह का Value कॉन्स्टेन्ट कहलाता है।

उदाहरणतः-यदि एक प्रोग्राम जो एक शिक्षक का वेतन 10 प्रतिशत बढ़ा दे तो आप इस पूरे प्रोग्राम में एक स्थिर मूल्यांकन घोषित कर सकते हैं। अगली बार यदि फिर से वेतन में वृद्धि करनी पड़े तो उस स्थिर मूल्य परिवर्तित करने से लक्ष्य प्राप्ति होगी जो कि निसंदेह उस प्रक्रिया से सरल है। जहाँ आपको पूरे प्रोग्राम में Value बदलना पड़े।

किसी कॉन्स्टेन्ट या स्थिर मूल्यांकन की घोषणा

name\_constant CONSTANT datatype = VALUE;

name\_constant मुख्यतः कॉन्स्टेन्ट का नाम है जो कि वेरिएबल नाम के ही समान है।

constant (reserved word) नामक शब्द किसी भी स्थिर वेल्ड की घोषणा हेतु उपयोग किया जाता है।

**प्रश्न 9. Exception Handling से आप क्या समझते हैं। इसकी साधारण रचना भी लिखिए।**

**उत्तर-** PL/SQL की एक सुविधा यह है कि यह PL/SQL खण्ड में आये Exceptions की Handling करता है, जिसे Exception handling कहा जाता है।

जब कभी कोई Exception आता है तो Exception के कारण की व्याख्या का एक Messages प्राप्त किया जाता है। PL/SQL Exception messages के तीन भाग होते हैं

1. Type of Exception
2. An Error Code
3. A message

Exception Handling से यह निश्चित है कि PL/SQL खण्ड एकाएक समस्या का कारण नहीं बनता है।

Exception भाग की साधारण रचना

DECLARE

Declaration part

BEGIN.

Exception part

EXCEPTION

WHEN excep 1 name THEN

Error handling statements

WHEN excep2name THEN

Error handling statements

WHEN Others THEN

Error handling statements

END;

**प्रश्न 10. PL/SQL ट्रिगर Execution Hierarchy बताइए।**

**उत्तर-** PL/SQL ट्रिगर Execution Hierarchy -जब ट्रिगर उपयोग में आता है तो निम्नलिखित हाइराइकी उपयोग में लायी जाती है

- सबसे पहले जब BEFORE वाक्य ट्रिगर उपयोग में आता है।
- उसके बाद जब BEFORE रो लेबल ट्रिगर उपयोग में आता है।

- उसके बाद जब AFTER रो लेबल ट्रिगर उपयोग में आता है। यह घटना BEFORE और AFTER रो लेबल ट्रिगर के मध्य अल्टरनेट देती है।
- सबसे अन्त में AFTER स्टेटमेन्ट ट्रिगर उपयोग में आता है।

## निबंधात्मक प्रश्न

**प्रश्न 1. PL/SQL SET Serveroutput ON क्या है? इसका उपयोग उदाहरण सहित समझाइए।**

**उत्तर-** PL/SQL SET SriveroutputON

“SET Serveroutput ON” हमेशा PL/SQL शुरू करने से पूर्व लिखना चाहिए। PL/SQL प्रोग्राम का Execution ऑरेकल इंजन में होता है, इसलिए सर्वर आउटपुट को स्क्रीन पर प्रदर्शित करने हेतु उसकी आवश्यकता पड़ती है अन्यथा परिणाम प्रदर्शित नहीं होगा।

SQL>सर्वर आउटपुट के उपयोग को समझने के लिए हम एक उदाहरण लेते हैं। प्रोग्राम की पहली लाइन सर्वर आउटपुट को शुरू करती है। तत्पश्चात् वेरिएबल्स और कॉन्स्टेन्ट का वर्णन होता है। प्लेसहोल्डर वर्णित करने के पश्चात् dbms output.put line नामक निर्देश का उपयोग वर्णित वेरिएबल के नाम को छापने हेतु होता है।

उदाहरण-

```
SQL >set serveroutput on
```

```
SQL>DECLARE
```

```
Sno number (4):NOT NULL:=3
```

```
SnameVarchar2(14):= 'Hari';
```

```
Sclass CONSTANT varchar2 (10): = '9th';
```

```
BEGIN
```

```
dbms_output.put_line ('Declared Value:');
```

```
dbms_output.put_line ('Student Numbr: || Sno||' Student Name: ||Sname);
```

```
dbms_output.put_line ('Constant Declared:');
```

```
dbms_output.put_line (student Class : '|| Sclass);
```

```
END;
```

परिणाम प्रदर्शित करने के लिए “set serveroutput on” नामक निर्देश का लागूकरण आवश्यक है।  
उपरोक्त कोड का आउटपुट

Declared Value:

Student Number: 3 Student Name: Hari

Constant Declared:

student Class: 9th

**प्रश्न 2. PL/SQL के नियमबद्ध (Conditional) वाक्य के विषय में विस्तारपूर्वक बताइए।**

## अथवा

### IF THEN ELSE वाक्य को उदाहरण सहित समझाइए।

**उत्तर-** PL/SQL के नियमबद्ध (Conditional) वाक्य नियमबद्ध वाक्यों के प्रकार एवं रचना निम्न प्रकार है

IF THEN ELSE – वाक्य-इस प्रकार के वाक्यों में जब अवस्था अनुकूल या TRUE होती है तब 1 वाक्य का Execution होता है, 2 वाक्य को छोड़ दिया जाता है परन्तु जब अवस्था FALSE होती है तब 1 वाक्य छोड़कर 2 वाक्य का Execution होता है।

वाक्य रचना (Syntaxes)

#### 1. IF condition

THEN

statement 1;

ELSE

statement 2;

END IF;

#### 2. IF condition 1

THEN

statement 1;

statement 2;

ELSEIF condition2 THEN

statement 3;

ELSE

statement 4;

ENDF

### प्रश्न 3. अंतर्निहित कर्सर क्या होता है? इसके उदाहरण दीजिए व गुणों के विषय में प्रोग्राम सहित बताइए।

**उत्तर-** अंतर्निहित (Implicit) कर्सर-यह पहले से बने होते हैं। जब किसी DML वाक्य; जैसे-INSERT, UPDATE और DELETE का लागूकरण किया जाता है। किसी SELECT वाक्य जो केवल एक पंक्ति परिणामस्वरूप देता है, के लिए भी कर्सर उपयोग में लाए जाते हैं।

अंतर्निहित कर्सर के उदाहरण – DML वाक्य; जैसे-INSERT, DELETE, UPDATE एवं SELECT के लागूकरण के लिए अंतर्निहित कर्सर उपयोग में लाये जाते हैं।

अंतर्निहित कर्सर के कुछ गुण (attributes) हैं जो DML की संचालन क्रिया (operations) की स्थिति बताते हैं। ये गुण %FOUND, %NOTFOUND, %ROWCOUNT और %ISOPEN हैं।

**उदाहरणतः** INSERT, UPDATE और DELETE वाक्यों के लागूकरण के समय कर्सर के गुण हमें ये बताते हैं कि इन वाक्यों का पंक्तियों पर प्रभाव हुआ है और कितनी पंक्तियों प्रभावित हुई हैं। जब कभी SELECT... INTO वाक्य का लागूकरण होता है तो अंतर्निहित कर्सर ये पता लगाने के लिए उपयोगी हैं कि एक भी पंक्ति उपरोक्त वाक्य के द्वारा प्रदर्शित हुई है या नहीं। यदि कोई पंक्ति नहीं चुनी गई है तो PL/SQL error होता है।

%FOUND गुण-प्राप्त परिणाम TRUE होगा यदि DML वाक्य; जैसे-INSERT, UPDATE, DELETE और SELECT...INTO कम-से-कम एक पंक्ति का संचालन करते हैं। परिणाम False या अनुचित होगा यदि उपरोक्त वाक्यों से एक भी पंक्ति का संचालन नहीं होता।

%Not Found गुण-परिणाम असत्य या अनुचित INSERT, UPDATE होगा। यदि DML वाक्य; जैसे-DELETE और SELECT INTO कम-से-कम एक पंक्ति का संचालन करें। यदि उपरोक्त वाक्यों से एक भी पंक्ति का संचालन न हो तो प्राप्त परिणाम उचित व सत्य होगा।

%ROWCOUNT गुण-परिणामस्वरूप उन सभी पंक्तियों की संख्या देता है जिनका संचालन INSERT, DELETE UPDATE एवं SELECT के द्वारा हुआ है।

**उदाहरणतः** विचार करें कि एक PL/SQL खण्ड जो अंतर्निहित कर्सर के गुणों का उपयोग करता है जो कि निम्नांकित

```
DECLARE rows_var number (7);
BEGIN
UPDATE Teacher
SET salary= salary+100;
IF SQL%NOTFOUND THEN
dbms_output.put_line ('salaries not updated');
ELSIFSQL% FOUND
rows-var;= THEN
ELSIF SQL% ROWCOUNT;
dbms_output.put_line ('Salaries for' || rows_var || teachers are updated');
END IF;
END;
```

उपरोक्त PL/SQL खण्ड में टीचर नामक टेबल में उपस्थित सभी शिक्षकों के वेतन को UPDATE किया गया है। यदि किसी शिक्षक का वेतन UPDATE नहीं होता है, तो हमें त्रुटि होने का सन्देश मिलता है जो कि

'salaries not updated' है। अन्यथा हमें सन्देश मिलता है कि 'Salaries for 100 teachers are updated' । यदि 'Teacher' टेबल में 100 पंक्तियाँ हैं।

**प्रश्न 4. PL/SQL में Procedure को कितने प्रकार से लागू किया जा सकता है। Procedure के निर्माण का एक उदाहरण दीजिए।**

**उत्तर-** निम्न दो प्रकार से Procedure का लागूकरण कर सकते हैं

1. SQL prompt से  
EXECUTE [or EXEC]procedure\_name;
2. दूसरी Procedure के अन्तर्गत-Procedure के नाम से procedure\_name;

**उदाहरणतः-** निम्नलिखित उदाहरण एक Procedures जिसका नाम 'student\_info' है का निर्माण करता है एवं इस Procedures का कार्य विद्यार्थियों की सूचनाएँ प्रदान करना है।

1. >CREATE OR REPLACE PROCEDURE student\_info
2. >IS
3. >CURSOR stu\_cur IS
4. >SELECT roll\_no, Sname, age FROM Student;
5. > stu\_rec stu\_cur% rowtype;
6. >BEGIN
7. >FOR stu\_rec in 1..7
8. >LOOP
9. >dbms\_output.put\_line (stu\_cur. Roll\_no|| " || stu\_cur.Sname
10. >|| "||stu\_cur.age);
11. >ENDLOOP;
12. >END;
13. >/