

Βάσεις δεδομένων

(10^ο μάθημα)

Ηρακλής Βαρλάμης
varlamis@hua.gr

Περιεχόμενα

- Ευρετήρια
- Σκανδάλες
- PL/SQL
- Δείκτες/Δρομείς



Αποθήκευση δεδομένων

- Πρωτεύουσα αποθήκευση
 - Κύρια μνήμη (main memory) ή κρυφή μνήμη (cache)
 - Άμεση προσπέλαση, περιορισμένος χώρος
- Δευτερεύουσα αποθήκευση
 - μαγνητικοί δίσκοι, ταινίες, δισκέτες
 - για την επεξεργασία των δεδομένων **απαιτείται η μεταφορά των δεδομένων στην πρωτεύουσα αποθήκευση**
 - πιο αργή προσπέλαση, μεγάλη χωρητικότητα, μικρότερο κόστος



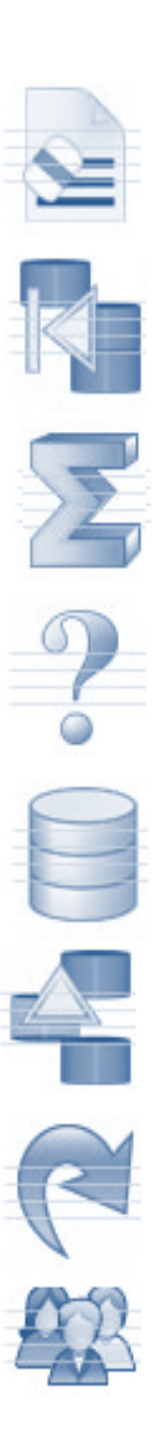
Οργάνωση αρχείων

- Τα δεδομένα αποθηκεύονται σε αρχεία στο δίσκο
 - Κάθε αρχείο χωρίζεται σε τμήματα (block) και έχει και μια επικεφαλίδα (file header) με πληροφορίες για την αρχή του κάθε block.
 - Σε μια επερώτηση, πλειάδες (εγγραφές) ανακτώνται από τα αρχεία έρχονται στην κύρια μνήμη και επεξεργάζονται
 - Τι γίνεται αν η κύρια μνήμη δεν επαρκεί για να αποθηκεύσει όλες τις πλειάδες;
 - Οι πλειάδες έρχονται ομαδοποιημένες σε μονάδες block (σελίδες – pages)
- π.χ. μια σελίδα χωρά 5 ακριβώς πλειάδες, η επερώτηση χρειάζεται 12 άρα θα ανακτηθούν 3 σελίδες από το αρχείο. Μόλις τελειώνει η επεξεργασία μιας σελίδας (πεντάδας πλειάδων) αρχίζει η επόμενη



Ευρετήριο - Index

- Βοηθητική δομή αρχείου που κάνει πιο αποδοτική την αναζήτηση μιας εγγραφής σε ένα αρχείο
- Καθορίζεται (συνήθως) σε ένα γνώρισμα του αρχείου που καλείται πεδίο ευρετηρίασης (**indexing field**)
- Τι αποθηκεύεται
 - στο κανονικό αρχείο: Οι εγγραφές σε blocks
 - στο αρχείο ευρετήριο: Η τιμή του πεδίου ευρετηρίασης και ένας δείκτης στο block της εγγραφής
- Κέρδος; Σε μία αναζήτηση με τιμή στο πεδίο ευρετηρίασης...
 - Δε φέρνουμε στη μνήμη όλα τα περιεχόμενα του κανονικού αρχείου - Φέρνουμε μόνο τα ζεύγη <τιμή,δείκτης> από το αρχείο ευρετήριο
 - Με μία ανάγνωση έχουμε στη μνήμη περισσότερες πλειάδες για σύγκριση – Εντοπίζουμε πιο γρήγορα τα σωστά block
 - Ανακτούμε μόνο τα σωστά block από το κανονικό αρχείο

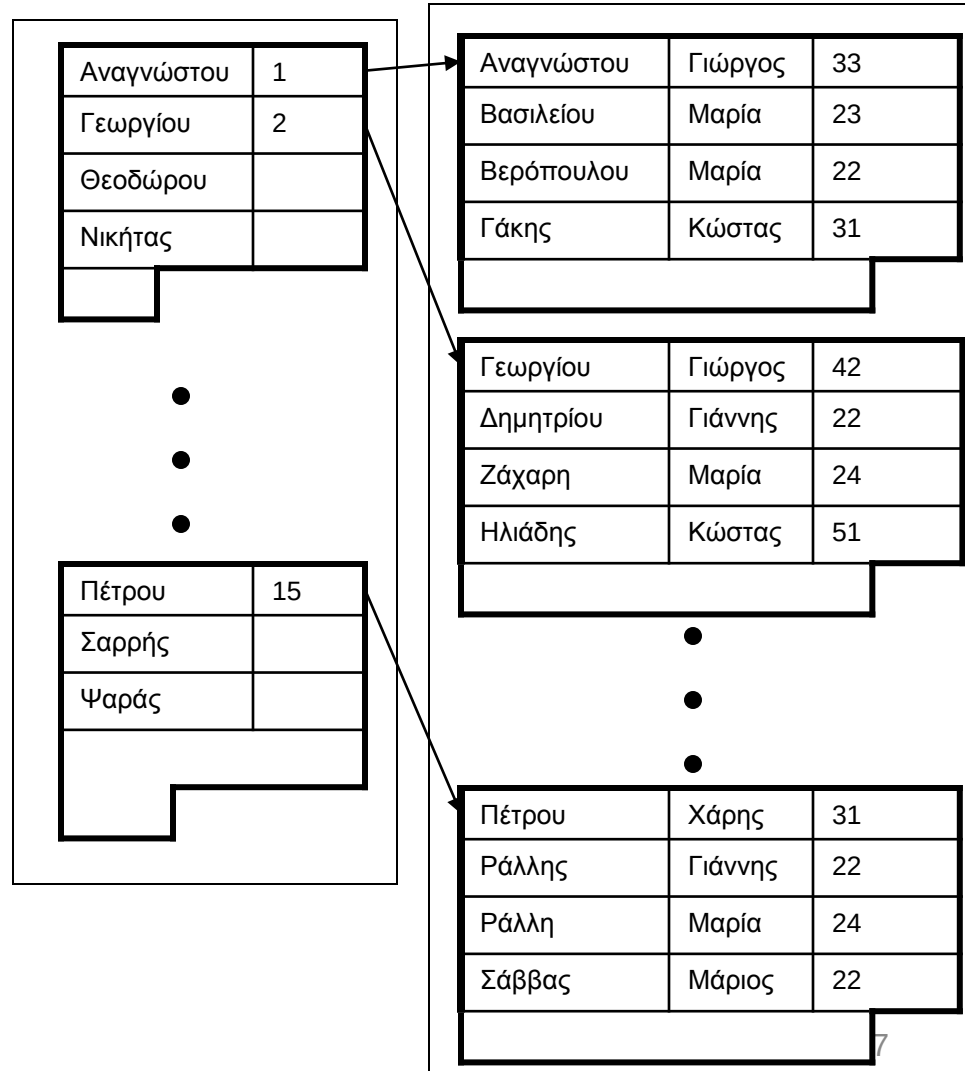


Εγγραφές ευρετηρίου

- Ανάλογα με τον τύπο του πεδίου ευρετηρίασης (κλειδί, πεδίο σε διάταξη, ύπαρξη διπλότυπων) έχουν και διαφορετικό τύπο
 - Πυκνό ευρετήριο: μία καταχώρηση για κάθε εγγραφή του αρχείου
 - Δευτερεύον ευρετήριο
 - Αραιό ευρετήριο: μια καταχώρηση για κάθε block του αρχείου δεδομένων
 - Πρωτεύον ευρετήριο
 - Ευρετήριο συστάδων
 - Κάνοντας *δυναμική αναζήτηση* στο ευρετήριο (γιατί το ευρετήριο είναι διατεταγμένο αρχείο!) βρίσκουμε τον δείκτη στο block όπου αποθηκεύεται η εγγραφή που θέλουμε
 - Εφόσον βρεθεί το block ακολουθείται σειριακή αναζήτηση

1. Πρωτεύον ευρετήριο

- primary index
- Στο πρωτεύον κλειδί (πεδίο κλειδιού διάταξης) ενός διατεταγμένου αρχείου
- Το ευρετήριο είναι επίσης διατεταγμένο αρχείο με εγγραφές σταθερού μήκους (κλειδί, δείκτης σε block)
- Στο ευρετήριο υπάρχουν τόσες εγγραφές όσα είναι τα block στα οποία αποθηκεύεται το κανονικό αρχείο



Παράδειγμα

• Έστω ένα αρχείο με:

- $r_A = 10000$ διατεταγμένες εγγραφές,
- σταθερού μεγέθους $S_A = 250$ bytes,
- μέγεθος block $S_B = 1024$ bytes με μη εκτεινόμενη καταχώρηση (ολόκληρες πλειάδες σε κάθε block)

– Σε κάθε block χωράνε $\left\lfloor \frac{S_B}{S_A} \right\rfloor = 4$ πλειάδες

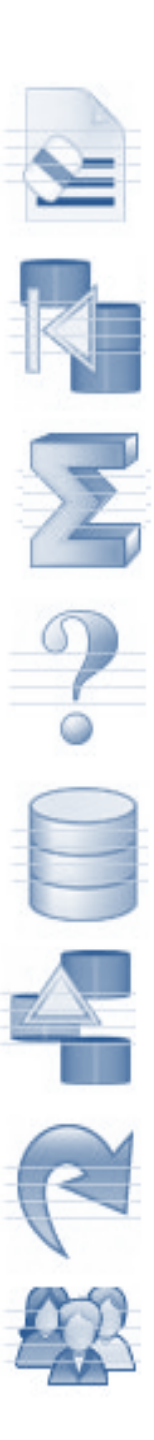
– Μέγεθος αρχείου δεδομένων $S_F = 10000/4 = \mathbf{2500 \text{ blocks}}$

• Κατασκευάζουμε πρωτεύον ευρετήριο (μη εκτεινόμενη καταχώρηση):

- Έστω ότι το μέγεθος του πεδίου κλειδιού είναι $S_K = 10$ bytes
- Και το μέγεθος δείκτη block $P = 6$ bytes

– Σε κάθε block ευρετηρίου $\left\lfloor S_B / (S_K + P) \right\rfloor = \left\lfloor 1024 / 16 \right\rfloor = 64$ εγγραφές

• Μέγεθος αρχείου ευρετηρίου $S_I = \left\lceil 2500 / 64 \right\rceil = \mathbf{40 \text{ blocks}}$



Κόστος αναζήτησης

- Αν θεωρήσουμε ότι το αρχείο δεδομένων είναι διατεταγμένο ως προς το πεδίο αναζήτησης μπορούμε να εφαρμόσουμε δυαδική αναζήτηση
 - Χωρίς το ευρετήριο: Θα πρέπει να φορτώσουμε $\log_2(S_F)$ blocks = $\log_2(2500)=11,28 \rightarrow 12$ blocks αρχείου και να συγκρίνουμε τις πλειάδες
 - Αν υπάρχει ευρετήριο: Θα πρέπει να φορτώσουμε $\log_2(S_I)$ blocks = $\log_2(40)=5,32 \rightarrow 6$ blocks ευρετηρίου και 1 block αρχείου (αυτό που θα μας υποδείξει η αναζήτηση στο ευρετήριο)

Χωρίς ευρετήριο: $\log_2(2500)=12$

Με ευρετήριο: $\log_2(40)+1=7$

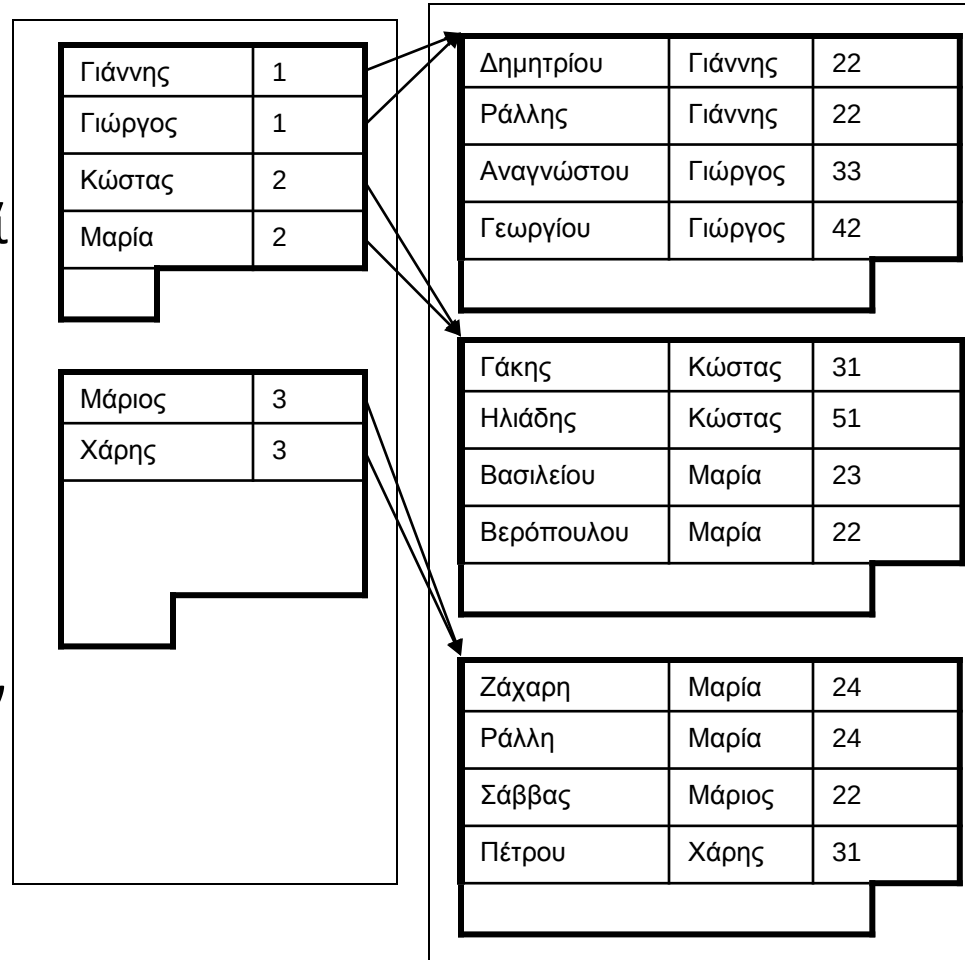


Κόστος ενημέρωσης

- Εισαγωγή εγγραφής
 - Συνεπάγεται αλλαγές στο αρχείο άρα και στο ευρετήριο
 - Οι αλλαγές μπορούν να αποθηκεύονται
 - σε ξεχωριστό αρχείο υπερχείλισης (χωρίς διάταξη)
 - ή σε συνδεδεμένη λίστα εγγραφών υπερχείλισης
 - και το ευρετήριο να ενημερώνεται περιοδικά
- Διαγραφή εγγραφής
 - Συνεπάγεται αλλαγές στο ευρετήριο
 - Συνήθως αντιμετωπίζεται με σημάδια διαγραφής

2. Ευρετήριο συστάδων

- Clustering index
- Ορίζεται στο πεδίο διάταξης το οποίο όμως **δεν** είναι κλειδί
- Υπάρχει μια εγγραφή για κάθε διακεκριμένη τιμή του πεδίου διάταξης (συστάδας) του αρχείου που περιέχει:
 - την τιμή αυτή
 - ένα δείκτη προς το πρώτο block του αρχείου δεδομένων που περιέχει μια εγγραφή με την τιμή αυτή στο πεδίο συστάδας



Εναλλακτικά

- Με ξεχωριστά block για κάθε ομάδα εγγραφών

Γιάννης	1
Γιώργος	2
Κώστας	3
Μαρία	4

Μάριος	5
Χάρης	6

Δημητρίου	Γιάννης	22
Ράλλης	Γιάννης	22
		null

Αναγνώστου	Γιώργος	33
Γεωργίου	Γιώργος	42
		null

Γάκης	Κώστας	31
Ηλιάδης	Κώστας	51
		null

Βασιλείου	Μαρία	23
Βερόπουλου	Μαρία	22
Ζάχαρη	Μαρία	24
Ράλλη	Μαρία	24
		null

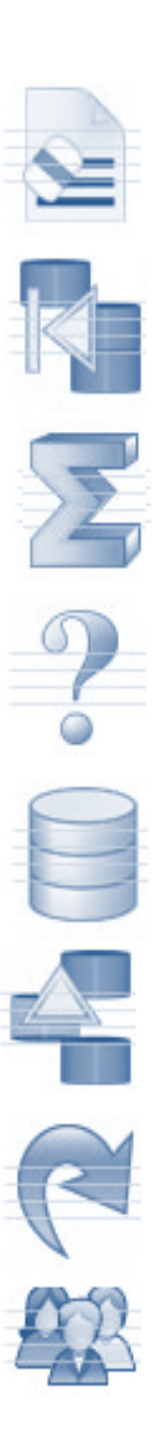
Σάββας	Μάριος	22
		null

Πέτρου	Χάρης	31
		null



Παράδειγμα

- Έστω ένα αρχείο με:
 - $r_A = 10000$ διατεταγμένες εγγραφές,
 - σταθερού μεγέθους $S_A = 250$ bytes,
 - μέγεθος block $S_B = 1024$ bytes με μη εκτεινόμενη καταχώρηση (ολόκληρες πλειάδες σε κάθε block)
 - Σε κάθε block χωράνε $1024/250 = 4$ πλειάδες
 - Μέγεθος αρχείου δεδομένων $S_F = 10000/4 = \mathbf{2500 \text{ blocks}}$
- Κατασκευάζουμε ευρετήριο συστάδων (μη εκτεινόμενη καταχώρηση):
 - Έστω μέγεθος του πεδίου ευρετηρίασης είναι $S_K = 10$ bytes
 - Μέγεθος δείκτη block $P = 6$ bytes
 - Και 500 διαφορετικές τιμές στο πεδίο ευρετηρίασης με ομοιόμορφη κατανομή των πλειάδων σε αυτές
 - Σε κάθε block ευρετηρίου χωράνε όπως πριν $1024/16 = 64$ εγγραφές
- Μέγεθος αρχείου ευρετηρίου $S_I = 500/64 = 7,81 \rightarrow \mathbf{8 \text{ blocks}}$



Κόστος αναζήτησης

- Θεωρούμε ότι το αρχείο δεδομένων είναι διατεταγμένο ως προς το πεδίο αναζήτησης μπορούμε να εφαρμόσουμε δυαδική αναζήτηση
- 500 διαφορετικές τιμές $\rightarrow 10000/500=20$ πλειάδες με την ίδια τιμή πεδίου αναζήτησης $\rightarrow 20/4=5$ block αντιστοιχούν σε κάθε τιμή
 - Χωρίς το ευρετήριο: Θα πρέπει να φορτώσουμε $\log_2(S_F)$ blocks = $\log_2(2500)=11,28 \rightarrow 12$ blocks αρχείου και 5 block για τις πλειάδες που έχουν την τιμή αναζήτησης
 - Αν υπάρχει ευρετήριο: Θα πρέπει να φορτώσουμε $\log_2(S_I)$ blocks = $\log_2(8)=3 \rightarrow 3$ blocks ευρετηρίου και 5 block αρχείου (για τις πλειάδες που έχουν την τιμή αναζήτησης)

Χωρίς ευρετήριο: $\log_2(2500)+5=17$

Με ευρετήριο: $\log_2(8)+5=8$



- 15



Δημιουργία ευρετηρίου

```
CREATE INDEX idxempfname on employee(fname)  
CREATE UNIQUE index idxemp on employee(lname)
```

Το primary key κάθε πίνακα έχει ήδη ευρετήριο
UNIQUE ευρετήρια μόνο σε πεδία που δεν έχουν διπλότυπα

Το ευρετήριο μπορεί να περιλαμβάνει περισσότερα από ένα πεδία, δεν είναι όμως τόσο αποτελεσματικό



Διαχείριση ευρετηρίου

- Ενημέρωση: Μετά από πολλές εισαγωγές και διαγραφές το ευρετήριο μπορεί να πρέπει να ξαναδημιουργηθεί:

`ALTER INDEX όνομαΕυρετηρίου REBUILD;`

- Διαγραφή:

`DROP INDEX όνομαΕυρετηρίου;`



Σκανδάλη - trigger

- Διαδικαστικός (procedural) κώδικας που εκτελείται ως συνέπεια κάποιων γεγονότων στη ΒΔ: INSERT, UPDATE, DELETE
- Ο κώδικας μπορεί να πυροδοτηθεί πριν (BEFORE) ή μετά (AFTER) το γεγονός
- Με τη χρήση triggers μπορούμε να ελέγχουμε σύνθετους περιορισμούς πριν από μια εισαγωγή ή ενημέρωση και να κάνουμε τις κατάλληλες διορθώσεις
- Χρησιμοποιούνται επίσης όταν η σχεδίαση της ΒΔ δεν καλύπτει όλες τις συναρτησιακές εξαρτήσεις

Παράδειγμα trigger

CREATE OR REPLACE TRIGGER *firstEmployee*

ΓΕΓΟΝΟΣ

BEFORE INSERT ON *employee*

ΠΥΡΟΔΟΤΗΣΗΣ

FOR EACH ROW

ΣΥΝΘΗΚΗ

BEGIN

ΠΥΡΟΔΟΤΗΣΗΣ

DBMS_OUTPUT.PUT_LINE('INSERTING: ' || :NEW.LNAME);

IF (:NEW.dno > 5) THEN

insert into department(dname,dnumber,mgrssn) values ('neo',:new.dno,123456789);

ΑΠΟΤΕΛΕΣΜΑ

END IF;

END;

ΑΝΑΦΟΡΑ ΣΤΗ ΝΕΑ
ΠΛΕΙΑΔΑ

Set serveroutput on



Εναλλακτικά



```
CREATE OR REPLACE TRIGGER firstEmployee
BEFORE INSERT ON employee
FOR EACH ROW
declare found boolean;
BEGIN
found:=false;
  DBMS_OUTPUT.PUT_LINE('INSERTING PERSON: ' || :NEW.LNAME);
  for dnum in (select dnumber from department where dnumber=:new.dno)
  loop
    found:=true;
  end loop;
  IF (found) THEN
    insert into department(dname, dnumber, mgrssn) values
    ('neo',:new.dno,123456789);
  END IF;
END;
```





Εισαγωγή στην PL/SQL

Η δομή μιας εντολής PL/SQL

DECLARE

/* Δηλώσεις (προαιρετικά):

Μεταβλητές, τύποι και τοπικά υποπρογράμματα.

*/

BEGIN

/* Εκτελέσιμο (υποχρεωτικό):

διαδικασίες και οι εντολές PL/SQL. */

EXCEPTION

/* Χειρισμός εξαιρέσεων (προαιρετικό):

εντολές χειρισμού λαθών*/

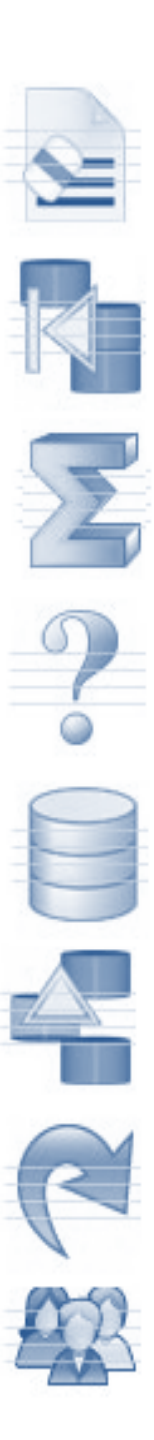
END;





Εκτέλεση

- Για να εκτελέσουμε ένα πρόγραμμα PL/SQL γράφουμε μετά τον κώδικα:
 - Μια γραμμή που περιέχει μόνο μία τελεία(".").
 - Μια γραμμή με την λέξη run;
- Η εκτέλεση γίνεται:
 - Γράφοντας τις εντολές στην γραμμή εντολών της sqlplus.
 - Γράφοντας τον κώδικα σε ένα αρχείο και καλώντας το από την sqlplus (@codefile.sql).



Μεταβλητές και τύποι

- Τύποι μεταβλητών:
 - Ακέραιοι: PLS_INTEGER, NUMBER(*length*)
 - Αλφαριθμητικά: CHAR(*length*), VARCHAR2(*max length*)
 - Πραγματικοί: NUMBER(*precision*, *scale*), REAL, FLOAT(*binary precision*)
 - Ίδιος με τον τύπο μίας στήλης ενός πίνακα.
 - Ίδιος με τον τύπο μίας εγγραφής με πολλά πεδία (tuple).
- Η αρχική τιμή μεταβλητών είναι NULL.
- Οι μεταβλητές δεν πρέπει να έχουν ίδιο όνομα με στήλες πινάκων.
- Στις μεταβλητές ανατίθεται τιμή με τον τελεστή “:=”.



Τμήμα δηλώσεων

DECLARE

name [CONSTANT] datatype [(constraint)] [:= value];

π.χ.

v_empno	Number(4) ;
xyz	VARCHAR2(200) := 'Hello World';
myvariable	INTEGER DEFAULT 1;
does_dept_exist	Boolean ; --true, false, null
pi	constant Number(6,4) := 3.1421 ;
next_checkup	Date Not Null := '2-Dec-92';
yourSalary	employee.salary%TYPE;
temp_emp	employee%Rowtype ;

Τμήμα εντολών

BEGIN

/ Εκτελέσιμο (υποχρεωτικό):
διαδικασίες και οι εντολές PL/SQL. */*

DBMS_OUTPUT.PUT_LINE(myvariable);

DBMS_OUTPUT.PUT_LINE(xyz);

myvariable := 15;

DBMS_OUTPUT.PUT_LINE(myvariable);

EXCEPTION

 WHEN *exc_name*
THEN
 εντολές χειρισμού;

...

[WHEN OTHERS
 εντολές χειρισμού;]

END;

/

- Συχνές ερωτήσεις:

- Θέλουμε να εκτελούμε μια ερώτηση που να εμφανίζει όλους τους υπαλλήλους

CREATE OR REPLACE PROCEDURE p1 () AS

```
e_fname VARCHAR2(50);
```

BEGIN

```
SELECT fname INTO e_fname FROM employee;
```

END;

CALL p1 ();

- Παραμετρικές ερωτήσεις

- Θέλουμε να εκτελούμε μια ερώτηση που να εμφανίζει όλους τους υπαλλήλους ενός τμήματος

CREATE OR REPLACE PROCEDURE p2 (IN k INT)

```
SELECT * FROM employee WHERE dno=k;
```

CALL p2(8);



Περιπτώσεις χρήσης

- Συνθήκες

```
CREATE OR REPLACE PROCEDURE p3 (IN k INT)
DECLARE v1 INT;
BEGIN
SET v1 = k % 2;
IF v1 = 0 THEN
    INSERT INTO month VALUES ('mono');
ELSE
    INSERT INTO month VALUES ('zygo');
END IF;
END;
CALL p3 (30) ;
```



Ένα πρόγραμμα περιέχει

- Μεμονωμένες εντολές ή σύνολα εντολών (blocks)
 - **Ανώνυμα blocks**
 - Επώνυμα blocks
 - Procedures
 - Functions



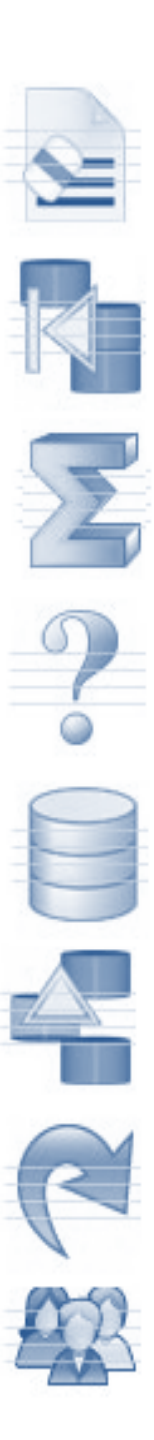
Αποθηκευμένη διαδικασία

- Stored procedure
- Συνήθως:
 - Κάνουμε μία ή περισσότερες ερωτήσεις στη ΒΔ
 - Ανακτούμε τις πλειάδες και τις μεταφέρουμε σε μια εφαρμογή
 - Επεξεργαζόμαστε τις απαντήσεις και κάνουμε νέες ερωτήσεις
- Με τις αποθηκευμένες διαδικασίες
 - Εκτελούμε ερωτήσεις στη ΒΔ
 - Ανακτούμε πλειάδες και τις επεξεργαζόμαστε στο ΣΔΒΔ
 - Μεταφέρουν μέρος των υπολογισμών (της λογικής) από την εφαρμογή στο ΣΔΒΔ
- Μια διαδικασία έχει όνομα, λίστα ορισμάτων και μια ή περισσότερες εντολές SQL



Σε ένα PL/SQL block

- Συνθήκες και έλεγχος ακολουθίας ροής
 - IF THEN, IF THEN ELSE
 - GOTO
 - CASE δηλώσεις
- Βρόχοι (loops)
 - FOR (αριθμητικά και με χρήση δείκτη),
 - WHILE
 - απλά LOOP
- Τελεστές ανάθεσης, σύγκρισης κλπ.



Δείκτες - Cursors

- Μας επιτρέπουν να διατρέχουμε τις επιλεγμένες πλειάδες και να τις χειριζόμαστε προγραμματιστικά με αποθηκευμένες διαδικασίες
 - Δηλώνουμε το δείκτη
`DECLARE cursor-name CURSOR FOR SELECT ...;`
 - Τον δημιουργούμε
`OPEN cursor-name;`
 - Ανακτούμε τις πλειάδες
`FETCH cursor-name INTO variable [, variable];`
 - Κλείνουμε το δείκτη
`CLOSE cursor-name;`



Παράδειγμα

```
CREATE OR REPLACE PROCEDURE findtotal (return_sum OUT NUMBER) IS
misthos employee.salary%TYPE;
total employee.salary%TYPE;
CURSOR curmisth IS SELECT salary FROM employee;
BEGIN
    total:=0;
    OPEN curmisth;
    LOOP
        FETCH curmisth INTO misthos;
        EXIT WHEN curmisth%NOTFOUND;
        IF misthos is not NULL THEN
            total:=total+misthos;
        END IF;
    END LOOP;
    CLOSE curmisth;
    return_sum:=total;
END;
```

```
DECLARE
total employee.salary%TYPE;
BEGIN
total:=0;
findtotal(total);
DBMS_OUTPUT.PUT_LINE('Total salary:' || total);
END;
```