~~BD(curs)~~ → nr. 12 17. XII. 2007.

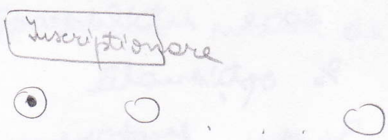
{ 3,10 XII. - BD
{ IRA de joi

Dezvoltare aplicații BD

- manipulare BD (cereri → SQL)
- MENU: este similar cu tipul de meniu al mediului de BD
- FORMS: interfață cu utilizatorul
- REPORTS: rapoarte asociate = inform. extrase din BD prin interogări și afișare/tipărire;

FORMS: dpdv compoziție, cuprinde:

- text area = zonă în care se introduce sau vizualizează date
- text: dif. între textarea și text este:
  - text - inform. limitată, care nu conține simboluri de tip carriage/return și are o singură linie
  - textarea = box cu mai multe linii
- butoane:
  - a) push button
  - b) radio button



- lista de selecții: inform. sunt predef. în aceste zone și se alege una dintre ele.
- check box: pot fi nici unul/unul/toate active;

- grid: organizare similară cu o tabelă; rezult. unei tabele este pus într-un grid.

Dpdv al dezv. aplicației, în Oracle avem:

- structura aplicației Oracle: client-server: ∃ server către care se fac interogări și ∃ mai multe mașini client. Aplicația server este serverul de BD.

SQL = limbaj de interogare, NU de programare

dezv. de aplicații se face în PL/SQL; include toate TOOLS ORACLE.
↓
programming language.

PL/SQL include pe lângă un limbaj de programare specific, și facilități de interogare, însă dpdv al execuției, partea de SQL e transmisă spre execuție serverului, dar restul de componente sunt

executate pe mașina locală.

d.p.d.v. structural, Oracle utilizează ca unitate de execuție-blocuri. Un bloc este văzut izolat în mediul Oracle.

Un bloc poate implementa:

- funcție
- procedură
- trigger referitor la BD sau la progr. de aplicație

La execuția unui bloc, anumite componente se transmit către server pt a fi executate, altele se execută local.

Structura bloc:

local:

DECLARE % componentă opțională
   declarații de variabile utilizate în bloc
   (tip de dată, format)

server:

BEGIN % obligatoriu, este partea executabilă din bloc
   - cereri SQL (interogare BD); această parte de cereri e neprocedurală;

local:

   - instr. PL/SQL, care utilizează rezultatele cererilor;
EXCEPTION % opțională
   - instr. PL/SQL pt. tratare excepții
END;

Blocul se transmite o singură dată la server, serverul execută din bloc, numai cererile SQL (se face degrevarea de alte sarcini). După executarea cererilor, serverul returnează rezult. cererilor.

Aplicația se poate descrie în limbaj nativ al mediului de BD (PL/SQL → procedural) SAU aplicația se poate descrie în limbaj gazdă.

Securitatea datelor în BD

Accesul la date: nu toată inform. stocată în BD e disp. pt. toţi utilizat. Prin drepturile de acces ale utilizat., li se asigură acestora accesul numai la anumite inform.

- acces la date <=> READ

⊖ modificare date: operaţii de tip UPDATE, DELETE, INSERT (APPEND).

- modif date <=> WRITE

OBS: Op de tip ALTER... sunt specifice doar admin. BD.

O schemă de securitate la o BD se împarte în:

a) schemă de securit. la nivel SO: acordarea de drepturi pe volume logice, directoare, fişiere ⟶ drepturi referitoare la stocare.

b) securitate intrinsecă: care se referă la accesul şi modif. BD. Securit. intrinsecă repres al doilea nivel de securizare peste SO.

D.p.d.v. al securit. intrinseci, obiectul poate fi împărţit în:

- tabele
- view - idee de crearea a unui nivel de securit. (care poate sau nu să fie executat)
- cereri: la execuţia unei cereri se verifică drepturile pt tabelele utilizate.

OBS: Un view poate fi văzut ca o cerere predefinită.

<u>drepturi la tabele</u>:

- ∃ 4 tipuri de drepturi

a) CAN-READ: dreptul prin care unui utilizator i se permite accesul în mod citire la o tabelă

b) CAN-WRITE: este un drept de tip UPDATE, sensul op. de WRITE la o tabelă e cel de scriere a inform în tabelă dar fără să modif. nr. de înregistrări (modif înreg. existente!)

c) CAN-DELETE: utilizat. poate şterge înreg din tabelă

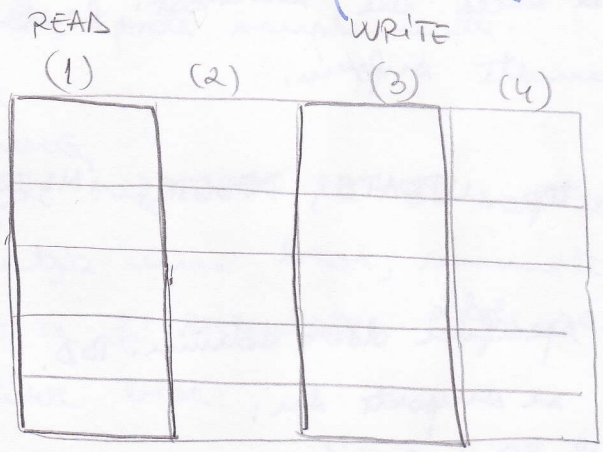
d) CAN-APPEND: dreptul de a adăuga înreg la tabelă.

OBS: considerăm că cele 4 drepturi sunt globale (se referă la tabelă în ansamblu).

Drepturi la câmpuri

a) CAN-READ: utilizatorul are acces la datele din acel câmp

b) CAN-WRITE: drept de a scrie un câmp.



OBS: utilizator va vedea doar câmpurile 1 și 3 și numai odată pe 2 și 4; pe 1 - poate numai să editeze, iar pe 3 doar să scrie (WRITE implică și READ).

- o metodă modalitate prin care să se facă acces selectiv la înregistrări.

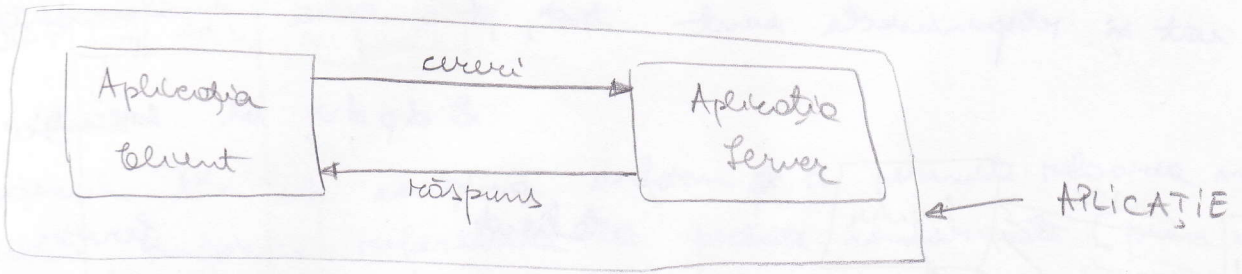
Drepturi la înregistrări: se poate face numai prin view; utilizator nu are acces la tabelă, ci numai la view, care afișează doar câmpurile pt care există drepturi.

OBS: - acces la bază prin ID_user
- securitatea: password → nu întotdeauna sigure

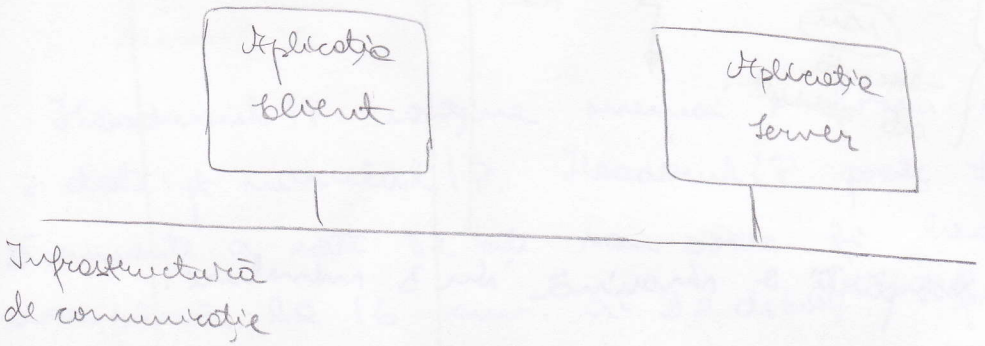
Pt mai multă siguranță, se pot implementa pt. securitate; și alte metode de securitate:

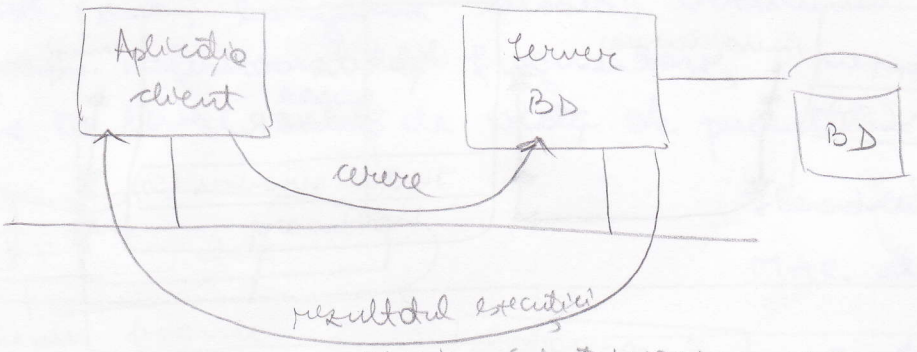
- smart card
- caracteristici biometrice < fingerprint (amprenta digitală)
  iris (retină)

Arhitectura client - server



- Client / Server Intrasistem: ambele aplicații rulează pe aceeași mașină, dar din punctul de vedere al SO - ex să avem multitasking.
- Client / Server Intersistem:





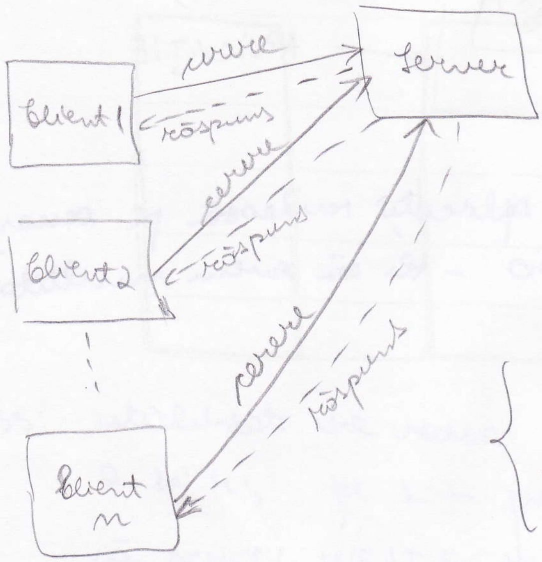
cerere (interogarea SQL e făcută de server).

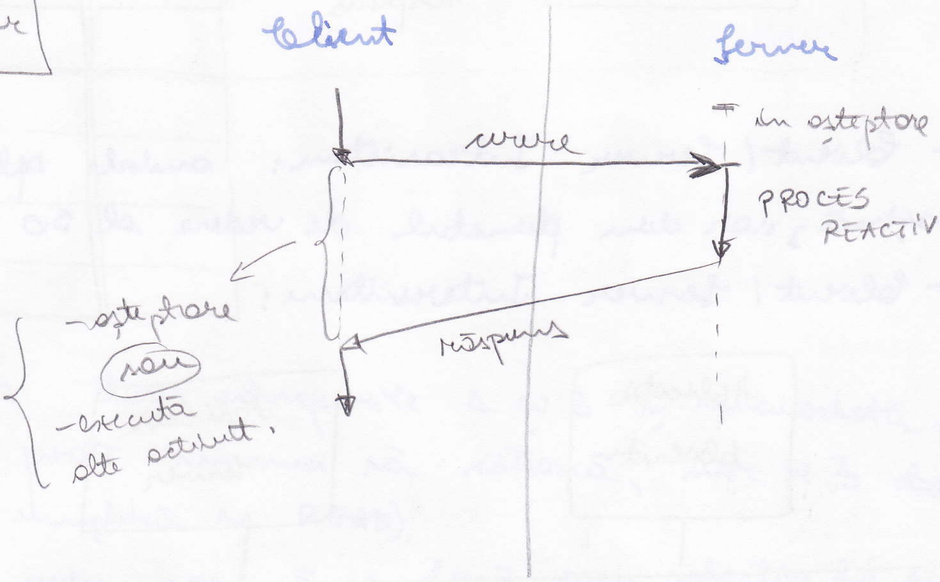
OBS: prin această structură, se face o diminuare a volumului de date manipulate.

Client	Server
- proactivă (rulează mereu)	- reactivă (intră în execuție numai când se formulează o cerere)
- generează mesaje către server	- așteaptă mesaje de la clienți - execută cerere expediată în mesaj - generează răspuns la client
- gestionează mesajele de la server - interfață utilizator, gestionare tastatură, mouse, validare locală date, gestionare vizată mesaje	- gestionare vizată mesaje, analiză cerere, execuție cereri clienți

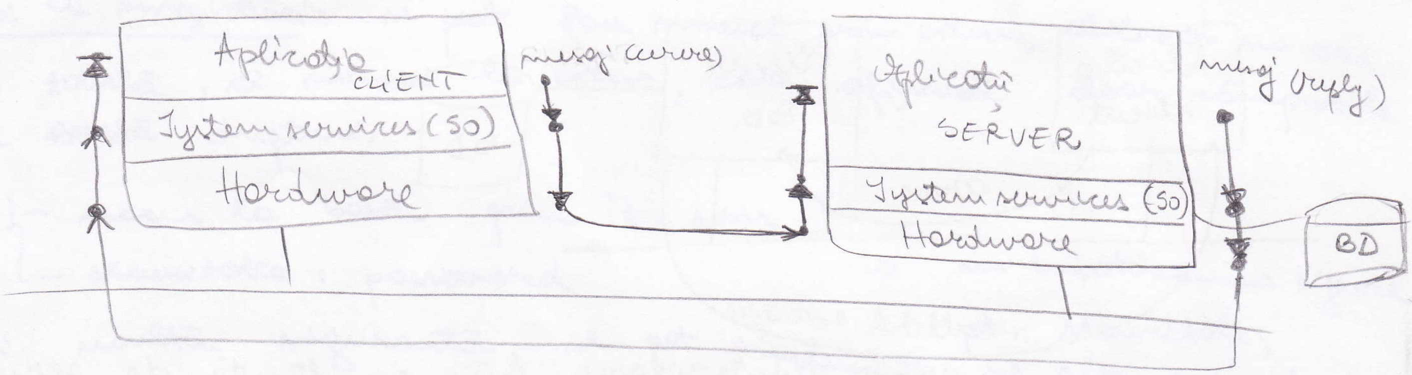
Pe client și pe server, I sist de operare diferite. ... transmise între client/server prin text (format ASCI). Atât cererile cât și răspunsurile sunt text, pt a avea independență față de SO.

D.p.d.v. al execuției:





D.p.d.v. logic, putem împărți o mașină în 3 niveluri:

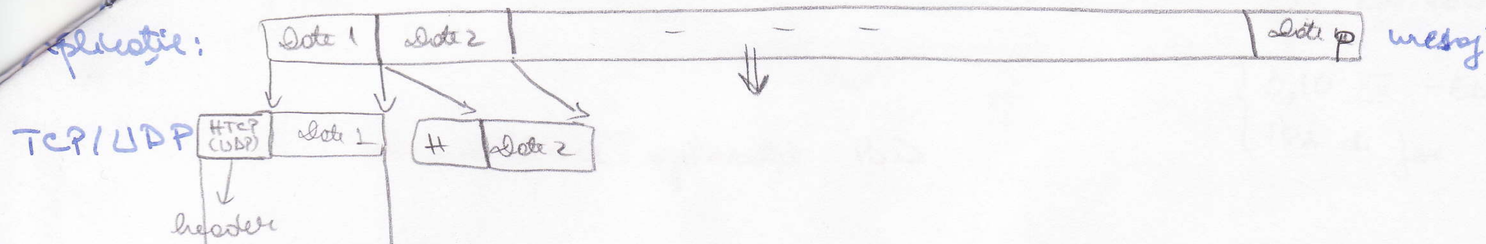


OBS. - pt client: avem structura de bloc
- pt cazul intrasistem: mesajele nu mai ajung la nivel hardware, ci doar la system services.

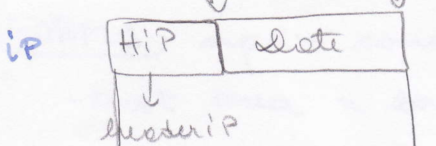
Protocolul TCP/IP: se găsește între aplicație și nivelul hardware.

Se poate transmite:
- TCP: Transport control Protocol
- UDP: User data Protocol

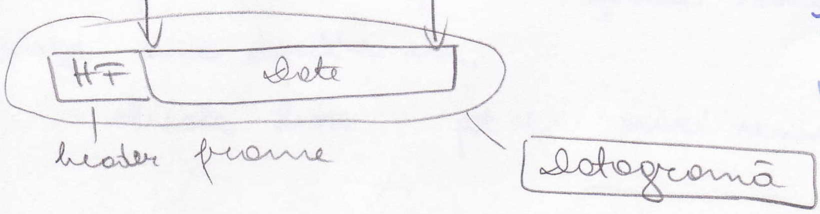
dif între cele 2: TCP e protocol de comunicație sigură, adică protocol ce necesită la recepție - confirmare și poate necesita retransmisă. UDP este nesigur, deoarece vor fi utilizate la receptor doar acele pachete ce au ajuns corect.



Headerul tb să conțină inform pt a permite refacerea mesajului din pachete, inform. referitoare la pachete confirmate (prin nr. de secvență de confirmare), portul și adresa ip a mașinii între care se stabilește conexiunea, o sumă de control a pachetului.



Headerul IP conține numai inform. de transport; HTCP+date = = date pt nivelul IP. Headerul IP poate fi minimal format din 5 cuvinte a câte 32 biți sau poate fi header extins ce poate avea până la 16 cuv. a 32 de biți. Inform. în HIP: adrese IP sursă/dest, lungime header, versiunea protocolului fizic, inform. referitoare la fragmentare, sumă de control header, time to live (~~comp~~ timp de viață al pachetului în rețea).



Header frame = cele 2 adrese MAC ale sursei și dest;
— asta e pachetul transportat în rețea.

OBS: În aplicații client-server NU lucrez cu UDP ci cu TCP. UDP e utilizat la aplicațiile video-conferință.