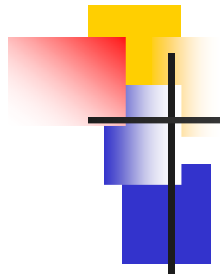




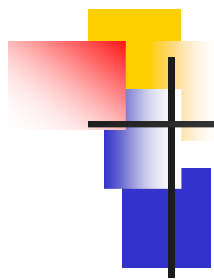
CURS 12

Baze de date distribuite



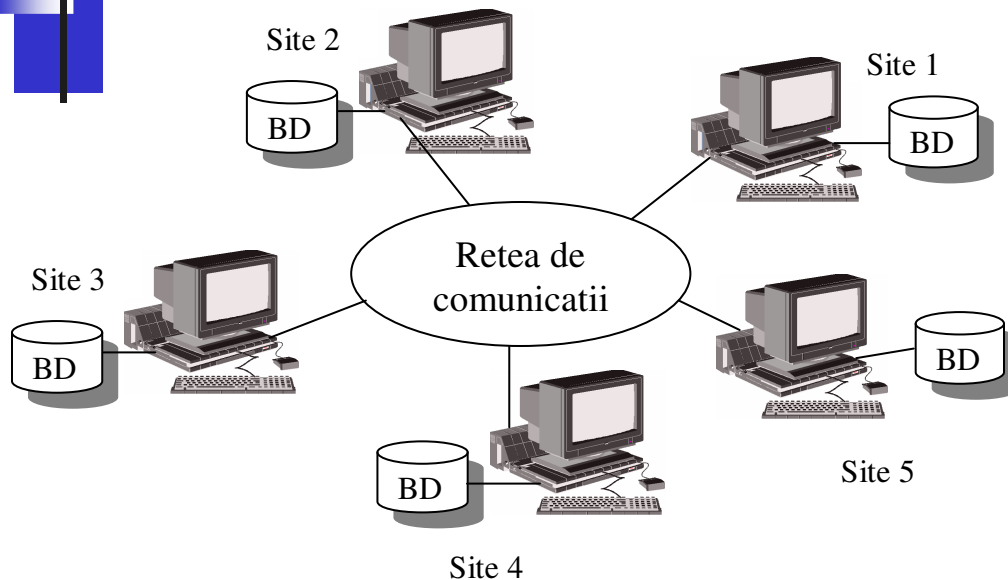
Introducere

- Un sistem de gestiune a bazelor de date distribuite (SGBDD) permite aplicațiilor să acceseze atât date din baza de date locală cât și din baze de date la distanță.
- Sunt 2 tipuri de SGBDD
 - Omogene – toate bazele de date sunt de același tip
 - Eterogene – bazele de date pot fi de diverse tipuri
- Bazele de date distribuite folosesc arhitecturi **client/server**



Definiție, caracteristici

- *Baza de date distribuită* - o colecție de date partajate, intercorelate logic, distribuite din punct de vedere fizic într-o rețea de calculatoare.
- *sistem de gestiune a bazelor de date distribuit (SGBDD)* - sistemul de programe care permite gestionarea bazei de date distribuite și care face ca distribuirea să fie transparentă pentru utilizatori
 - gestionează o singură bază de date logică, împărțită într-un număr de fragmente, fiecare din acestea fiind stocat pe unul sau mai multe calculatoare sub controlul unui SGBD separat



Fiecare site poate fi considerat complet din perspectiva unui sistem de baze de date, deoarece are:

- propriile sale baze de date locale,
- propriul sau SGBD,
- proprii săi utilizatori,
- un soft propriu de gestiune a tranzacțiilor
- propriul său manager local de comunicație a datelor.

⇒ un utilizator poate opera asupra datelor din site-ul său local ca și cum acesta nu ar fi componentă a sistemului distribuit

- fiecare site este caracterizat de un anumit grad de autonomie

Aplicațiile pot fi clasificate în

- *aplicații locale*, care nu necesită date din alte site-uri și
- *aplicații globale*, atunci când acestea gestionează date din mai multe site-uri.
- este necesar ca un sistem SGBDD să conțină cel puțin o aplicație globală.



Avantaje si ...

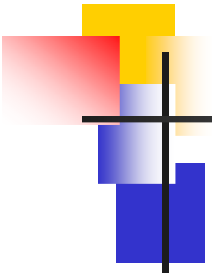
- cel mai semnificativ avantaj al sistemelor distribuite este faptul că tratarea distribuită a bazelor de date este o reflectare a structurii organizaționale a multor companii, care sunt distribuite logic în departamente, proiecte, etc. și fizic în fabrici, uzine sau puncte de lucru, unde fiecare unitate își întreține propriile sale date operaționale.
 - În aceste condiții sistemul informațional al organizației este divizat în așa numitele *insule de informații*, care pot fi interconectate prin facilitățile oferite de SGBDD. (C. J. Date, 1995)
- disponibilitatea și fiabilitatea crescută, obținute din faptul că într-un sistem distribuit până unui site sau a unei linii de comunicație nu împiedică sistemul să funcționeze deoarece acesta are capacitatea de a redirecționa spre un alt site cererile nodului defect.
- consultarea datelor este mult mai eficientă decât în cazul unei baze de date centralizată aflată la distanță, deoarece acestea sunt localizate în site-ul în care sunt cel mai frecvent solicitate,
- scalabilitate crescută. Atunci când este necesar, se pot adăuga în rețea situri noi, fără a afecta funcționarea celorlalte site-uri.



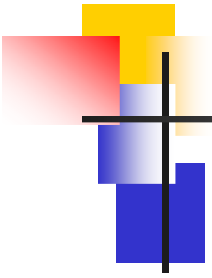
... dezavantaje

- complexitatea crescută a acestor sisteme derivată din necesitatea ca modulele software să trateze în mod corespunzător problemele legate de reproducerea datelor, și nu numai.
- prețuri mai mari legate de procurarea și întreținerea unui sistem distribuit, în comparație cu cele aferente unui sistem centralizat.
- este necesară efectuarea unor cheltuieli suplimentare cu elementele hardware necesare realizării legăturilor dintre site-uri.

Sistem omogen de baze de date distribuite

- 
- rețea de două sau mai multe sisteme de baze de date de același fel (Oracle) rezidente pe una sau mai multe mașini
 - o aplicație poate accesa sau modifica simultan date din mai multe baze de date într-un mediu distribuit
 - pentru o aplicație client locația și platforma diferitelor baze de date sunt transparente
 - este posibilă crearea de **sinonime** pentru obiectele aflate la distanță într-un sistem distribuit
 - utilizatorii pot accesa aceste obiecte folosind aceeași sintaxă ca pentru obiectele locale

Baze de date distribuite vs procesare distribuită

- 
- Baza de date distribuită
 - Un set de baze de date într-un sistem distribuit poate apărea unei aplicații ca o singură sursă de date
 - Procesarea distribuită
 - apare când o aplicația își distribuie sarcinile pe mașini diferite într-o rețea
 - În mod obișnuit, o aplicație de baze de date distribuie sarcinile de prezentare front-end pe calculatoarele client și permite unui server de baze de date back-end să gestioneze accesul partajat la baza de date.
 - Un sistem distribuit de procesare a bazelor de date este referit ca un sistem client/sever

Baze de date distribuite vs baze de date replicate

- într-o bază de date distribuită pură (nereplicata), sistemul gestionează o singură copie a datelor sau obiectelor bazei de date.
 - Aplicațiile de baze de date distribuite folosesc tranzacții distribuite pentru a accesa atât date locale cât și la distanță și pentru a modifica în timp real baza de date globală
- **Replicarea** se referă la operația prin care obiecte ale bazei de date sunt copiate și întreținute în mai multe site-uri ale unui sistem distribuit



Avantajele replicării

- Replicarea este utilizată pentru a îmbunătăți performanțele bazei de date locale și a proteja disponibilitatea aplicațiilor deoarece există opțiuni alternative de acces la date.
 - o aplicație poate accesa mai ușor o bază de date locală decât un server la distanță pentru a minimiza traficul pe rețea și a maximiza performanța
 - o aplicație poate continua să funcționeze dacă serverul local eșuează dar rămân accesibile alte servere pe care este replicata baza de date

Sisteme eterogene de baze de date distribuite



Sunt acele sisteme în care SGBD-urile care funcționează în diferite site-uri sunt de tipuri diferite

- și în aceste condiții aplicațiile vad întreaga bază de date ca fiind o bază locală
 - Serverul plasat într-un anumit site trebuie să ascundă distribuția și eterogenitatea bazei de date.

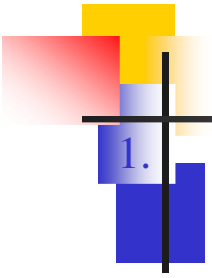
■ Exemplu:

- Un server Oracle poate accesa un sistem non-Oracle prin intermediul Oracle Heterogeneous Services în conjuncție cu un **agent**.
- Dacă se accesează date non-Oracle prin Oracle Transparent Gateway, atunci agentul este o aplicație specifică sistemului accesat
- Alternativ se poate folosi facilitatea de conectivitate generică pentru a accesa date non Oracle dacă aceste sisteme suportă protocoale ODBC sau OLE DB



Principiul fundamental și cele 12 reguli pentru bazele de date distribuite

- pentru utilizator, sistemul distribuit trebuie să arate exact la fel ca unul nedistribuit.
 - Ca urmare toate problemele sistemelor distribuite trebuie să fie interne sau de implementare și nu probleme externe sau de natură să afecteze utilizatorul
- principiu fundamental atrage după sine necesitatea respectării unui număr de 12 reguli

- 
1. autonomia locală
 2. absența unei dependențe de un site central
 3. operare continuă
 4. independența de localizare
 5. independența de fragmentare
 6. independența de reproducere
 7. prelucrarea distribuită a interogărilor
 8. gestionarea distribuită a tranzacțiilor
 9. independența de hardware
 10. independența de sistemul de operare
 11. independența de rețea
 12. independența de sistemul SGBD.

Autonomia locală

presupune ca toate operațiile dintr-un anumit site să fie controlate de acesta.

- nici un site A nu trebuie să depindă de un alt site B pentru succesul operării.
- autonomia locală se referă și la faptul că datele locale sunt deținute și gestionate local
 - toate datele aparțin, în realitate, unei baze de date locale, chiar dacă aceasta este accesibilă și din alte site-uri.
 - problemele legate de integritatea, securitatea și reprezentarea fizică pentru datele locale se rezolvă în site-ul căruia îi aparțin.
- În practică, acest lucru nu poate fi respectat în totalitate - sunt situații în care un site trebuie să acorde un anumit grad de control unui alt site.
- Concluzia - acest obiectiv ar putea fi enunțat mai nuanțat, în sensul că **site-urile trebuie să fie autonome în cea mai mare măsură posibilă.**



Absența dependenței de un site central

- toate site-urile trebuie tratate în mod egal.
 - *nu trebuie să existe nici un fel de dependență față de un site central* (de exemplu prelucrarea centralizată a interogărilor sau gestionarea centralizată a tranzacțiilor).
 - dependența ar fi inoportună deoarece în anumite situații acest site central ar putea reprezenta o strangulare sau sistemul în ansamblu ar fi vulnerabil – dacă site-ul central cade, atunci cade întregul sistem.



Posibilitatea de operare continuă

urmare a faptului că sistemele distribuite pot oferi o mai mare:

- fiabilitate (probabilitate ca sistemul să funcționeze în orice moment)
- disponibilitate (probabilitatea ca sistemul să funcționeze continuu pe parcursul unei perioade specificate).
 - în cazul în care într-un anumit punct din sistem apare o *întrerupere neplanificată* atunci sistemul poate continua să opereze chiar dacă la nivel redus.
- în sistemul distribuit nu trebuie solicitate niciodată ***întreruperi planificate***, pentru scopuri precum adăugarea unui nou site sau pentru modernizarea SGBD-ului dintr-un site existent.

Independența de localizare

■ *Se mai numește transparența de localizare*

- pornește de la ideea că utilizatorii nu trebuie să știe unde sunt stocate fizic datele, ci ei trebuie să se poată comporta ca și cum toate datele ar fi stocate pe site-ul lor local.
- permite datelor să migreze de la un site la altul fără a invalida un program aplicație sau vreuna din activități.
 - făcând o analogie cu proprietățile bazelor de date centralizate se poate afirma că independența de localizare este o extensie, în cazul sistemelor distribuite a conceptului de independență fizică a datelor.



Independența (transparența) de fragmentare

- presupune ca utilizatorul să se comporte, cel puțin din punct de vedere logic, ca și cum datele nu ar fi fragmentate deloc.
 - simplifică programele de aplicație și activitatea utilizatorilor finali.
- implică faptul că utilizatorilor le va fi prezentată o vedere a datelor în care diferitele fragmente pot fi recombinate logic prin uniuni sau reuniuni adecvate.
 - în acest caz optimizatorul are sarcina de a determina ce fragmente trebuie accesate fizic pentru a satisface orice cerere a utilizatorilor.



Independenta (transparența) de reproducere

- presupune ca utilizatorii să se poată comporta, cel puțin din punct de vedere logic ca și cum datele nu ar fi de fapt reproduse
 - simplifică programele de aplicație și activitatea utilizatorilor finali.
- permite crearea și distrugerea, în orice moment, a reproducerilor, fără a afecta vreun program sau alt gen de activitate.
 - În acest context este responsabilitatea optimizatorului să determine ce reproduceri trebuie accesate fizic pentru a satisface o anumită cerere a utilizatorului.



Independența de hardware

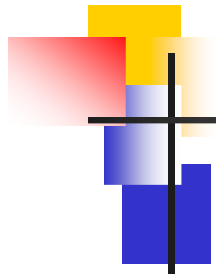
In lumea reală se utilizează o mare diversitate de calculatoare (IBM, Fujitsu, HP sau PC-uri și stații de lucru de diferite tipuri). Este necesar ca să existe posibilitatea de a integra datele de pe toate aceste sisteme pentru a se prezenta utilizatorului imaginea unui sistem unitar.

- *Independența de hardware* presupune și posibilitatea de a rula același SGBD pe diferite platforme hardware permițând diferitelor mașini să participe ca parteneri egali în sistemul distribuit.



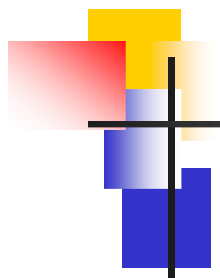
Independență față de sistemul de operare si independență față de rețea

- completare a independenței de hardware
- să admită ca un același SGBD să poată rula pe platforme cu diferite sisteme de operare. Poate exista situația în care sisteme de operare precum OS/400, UNIX și WINDOWS să participe în același sistem distribuit.
- Dacă un sistem distribuit admite mașini diferite pe care rulează sisteme de operare diverse atunci se pune, problema posibilității ca între site-uri comunicarea să poată fi făcută folosind diferite rețele de comunicație, cu alte cuvinte, să existe *independență față de rețea*.



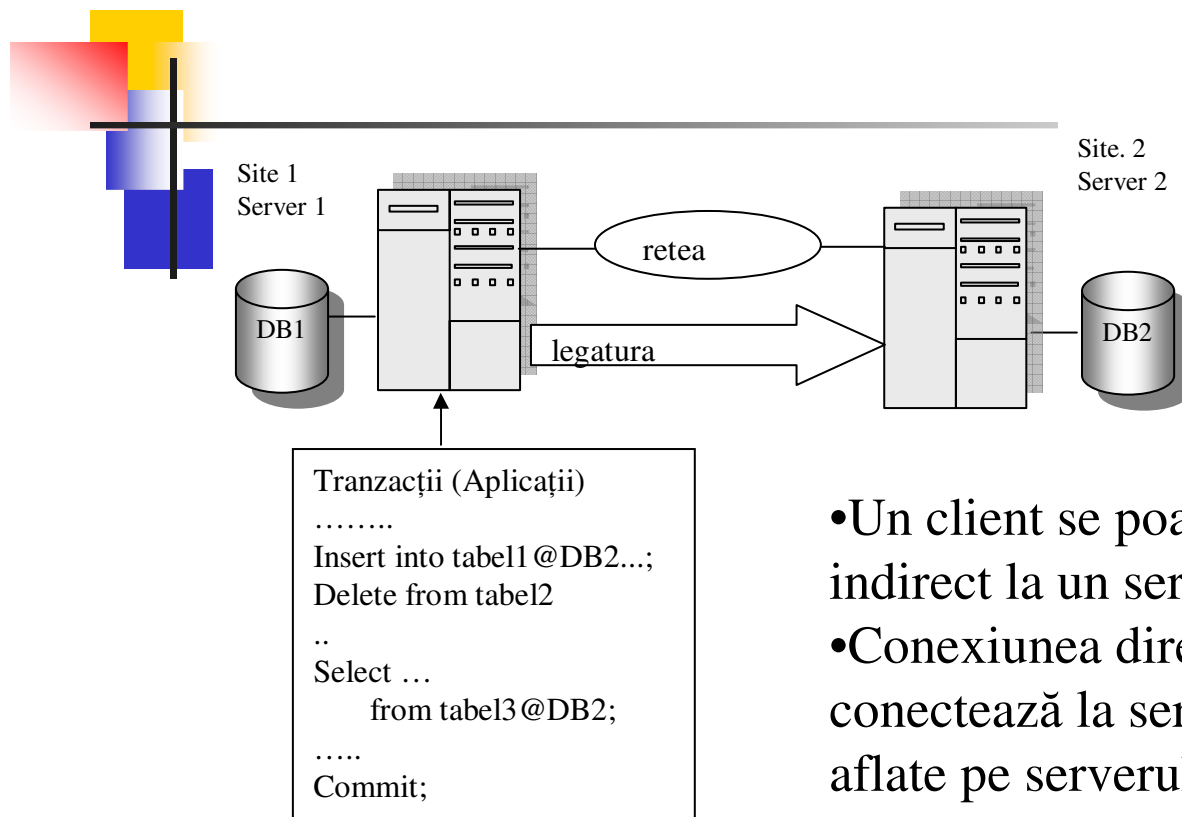
Independență de SGBD

- sistemele de baze de date distribuite pot fi omogene, caz în care toate site-urile au instalat același SGBD.
- În realitate însă, sistemele distribuite nu numai că au în componență site-uri cu mașini și sisteme de operare diverse, dar ele gestionează datele sub diferite SGBD-uri, deci ipoteza omogenității acestora este prea strictă.
- În fapt, singurul aspect necesar este ca diversele sisteme SGBD din site-uri să accepte aceeași interfață, deci un sistem distribuit ideal trebuie să ofere *independență de SGBD*.



Arhitectura Client/Server

- Intr-o arhitectura client/server serverul este SGBD-ul iar clientul este aplicația care solicită date de la server.
- Fiecare calculator dintr-o rețea este un nod care poate găzdui una sau mai multe baze de date.
- Fiecare nod al unei baze de date distribuite poate acționa ca un server, ca un client sau poate juca ambele roluri.

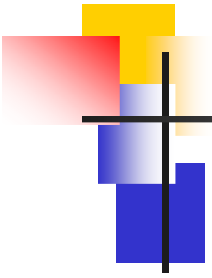


- Un client se poate conecta direct sau indirect la un server de baze de date
- Conexiunea directă apare când clientul se conectează la server și accesează datele aflate pe serverul respectiv.
- Conexiunea indirectă apare când un client se conectează la un server și apoi accesează datele conținute în baza de date aflată pe alt server

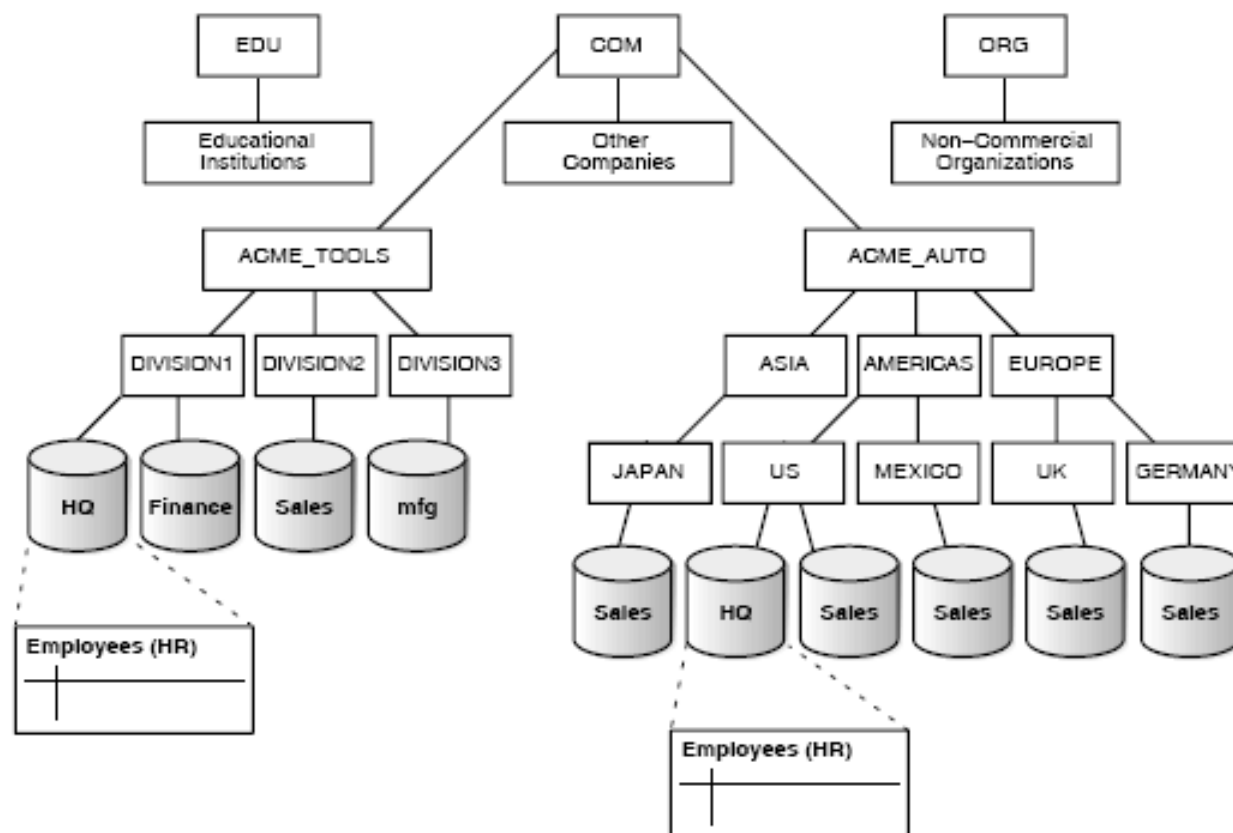


Legăturile bazei de date (Database Links)

- Link- pointer care definește o cale de comunicare unidirecțională de la un server de baze de date Oracle către un alt server de baze de date
- este memorată în baza de date
- Pentru a accesa link-ul un user trebuie să fie conectat la baza de date locală care conține intrarea corespunzătoare acestui link în dicționar
- Legătura este unidirecțională deoarece un client conectat la o bază de date A poate utiliza legăturastocată aici pentru a accesa datele dintr-o bază de date la distanță B, dar un user conectat în punctul B nu poate folosi același link pentru a accesa datele din site-ul A

- 
- Pentru a stabili o legătură fiecare bază de date dintr-un sistem distribuit trebuie să aibă un nume global unic (global database name) în domeniul rețelei
 - Acest nume identifică în mod unic baza de date în rețea
 - Cum se stabilește?
 - Se prefixează domeniul în rețea al bazei de date (care se specifică prin valoarea parametrului DB_DOMAIN la crearea bazei de date) cu numele individual al bazei de date (specificat în parametrul de inițializare DB_NAME)

Exemplu



Numele global al unei baze de date se stabilește pornind de la frunze și urmând o cale până la rădăcina arborelui ierarhic

Ex. `hq.us.americas.acme_auto.com`



Denumirea unui link

- Un link are aceeași denumire ca și numele global al bazei de date pe care o referă
- Dacă parametrul GLOBAL_NAMES este setat ca TRUE, Oracle asociază legăturii numele bazei de date globale referite
 - Pentru stabilirea unei legături Oracle verifică partea de domeniu al numelui global al bazei de date care este memorat în dicționarul bazei de date și nu pe cel atribuit parametrului DB_DOMAIN
 - Avantaj: legătura este transparentă pentru utilizator
- Dacă parametrul GLOBAL_NAMES este setat ca FALSE atunci se poate atribui legăturii orice nume este dorit de cel ce o creează



Tipuri posibile de link-uri

- D.p.d.v. al userilor care le pot folosi o legătură pot fi:
 - Privată – poate fi accesată numai de userul care a creat-o
 - Publică – poate fi folosită de toți userii bazei de date
- după modul în care se poate stabili legătura la o bază de date la distanță
 - Legătură pentru userul conectat (connected user link) – userii se conectează cu propriile lor credențiale
 - Acești useri trebuie să aibă pe serverul la distanță un cont cu același nume precum contul din baza de dae locală
 - Legătură pentru user fixat – userul se conectează cu userul și parola referită în legătură și are toate privilegiile pe care acest cont le are în baza de date țintă
 - Legătură pentru userul curent – un user se conectează ca un user global
 - Un user local se poate conecta ca un user global în contextul unei proceduri stocate fără a mamra parola userului global în definiția linkului



Crearea unui link

- **CREATE [SHARED] [PUBLIC] DATABASE LINK** nume_dblink [**CONNECT TO** { **CURRENT_USER** | user **IDENTIFIED BY** password [authenticated_clause] } | authenticated_clause] [**USING** 'connect_string'];
- Se definesc:
 - numele linkului
 - șirul de conectare pentru accesul la baza de date la distanță
 - numele și parola pentru conectarea la baza de date la distanță



Relația dintre obiectele schemă și link-ul la o baza de date

- După crearea unui link pot fi executate instrucțiuni SQL care să acceseze date aflate la distanță

```
SELECT * FROM emp@foo;
```

- este accesat obiectul *emp* aflat la distanță utilizând legătura *foo*
- Pentru anumite obiecte aflate la distanță poate fi nevoie de autorizare în baza de date la distanță
- Manipularea datelor în baze de date distribuite este posibilă prin construirea corectă a numelor obiectelor:

schema.obiect_schema@nume_global_baza_date



Sinonime pentru obiectele schemă

- Oracle permite crearea de sinonime care ascund utilizatorilor numele unui link
 - Este permis accesul la un tabel aflat la distanță folosind aceeași sintaxă care realizează accesul la un tabel al bazei de date locale

**CREATE [OR REPLACE] [PUBLIC]
SYNONYM [schema .] synonym FOR
[schema .] object [@ dblink];**



Exemplu

- Fie următoarea interogare:

SELECT * FROM emp@hq.acme.com

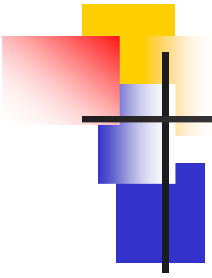
- Dacă se crează un sinonim pentru *emp@hq.acme.com* astfel:

**CREATE SYNONYM emp FOR
emp@hq.acme.com**

- Atunci interogarea se poate scrie:

SELECT * FROM emp

Restricții impuse de legăturile între bazele de date

- 
- Câteva exemple de operații ce NU pot fi realizate prin intermediul link-urilor
 - Acordarea de privilegii asupra obiectelor la distanță
 - Execuția comenzii DESCRIBE asupra obiectelor la distanță
 - Analiza acestor obiecte
 - Definirea sau forțarea constrângerilor de integritate
 - Acordarea de roluri utilizatorilor într-o baza de date la distanță