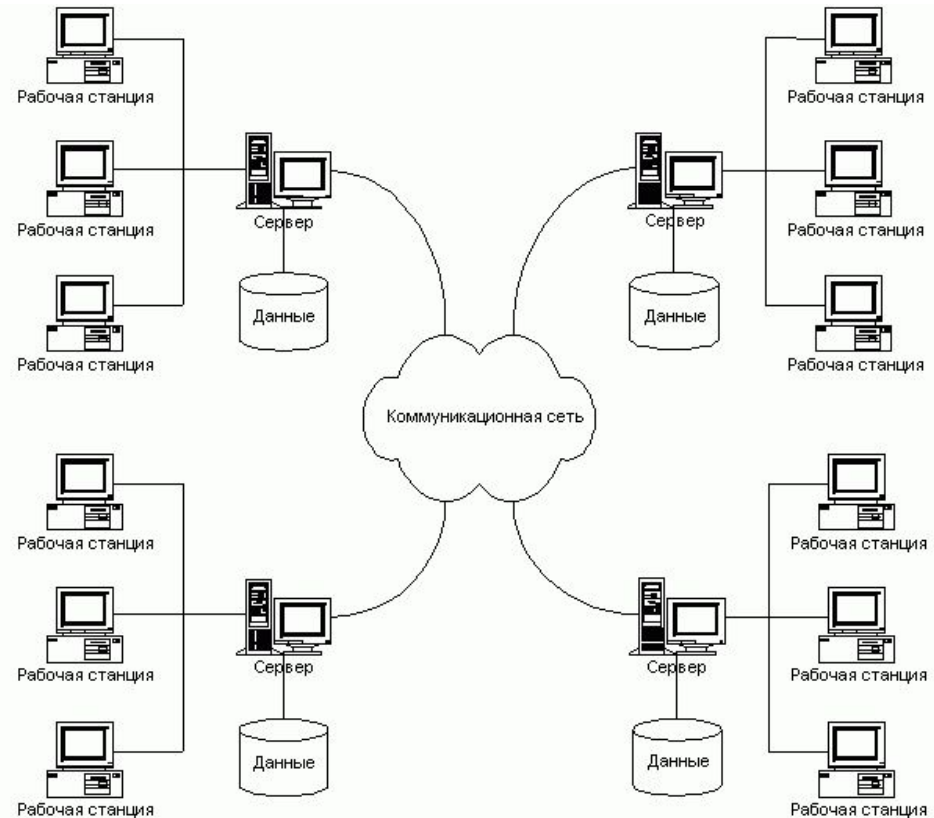


PAYLANMIŞ SİSTEMLƏR

(distributed systems)

- Tətbiqi proseslər və bu proseslərdə istifadə olunan verilənlər bazaları toplusu verilənlərin emalının paylanmış sistemləri və ya, sadəcə olaraq, paylanmış sistemlər (PS) adlanırlar. Kompüter şəbəkəsinin qovşaqları isə paylanmış sistemin qovşaqları adlanır.



Paylanmış sistemin modeli

- Paylanmış sistem $S = \{S_1, S_2, \dots, S_N\}$ qovşaqları və istənilən qovşaqlar cütünü birləşdirən verilənlərin ötürülməsi sistemi toplusudur.
- PS-in qovşaqlarında verilənlər bazaları (VB) şəklində təşkil olunmuş informasiya saxlanılır və tətbiqi proseslərlə emal olunur.
- Tətbiqi proseslər bir-biri ilə informasiyanın məzmunlu emalına təkan verən və ya ancaq idarəedici funksiyaları yerinə yetirən məlumatların mübadiləsi yolu ilə qarşılıqlı əlaqədədirlər. Emal edən məlumatlar: M_{ij} ($i, j = 1, \dots, N$), idarəedici məlumatlar: m_{ij} ($i, j = 1, \dots, N$),
- Verilənlərin (məlumatların) ötürülməsi sistemi etibarlıdır, yəni məlumatlar itmir və öz-özünə yaranmır, bütün qovşaqlar məlumatları qəbul etməyə hazırdırlar.
- Paylanmış sistemlərin qovşaqlarında fəaliyyət göstərən prosessorlar etibarlıdırlar.

Paylanmış sistemlərin idarə edilməsi

■ mərkəzləşdirilmiş

■ mərkəzləşdirilməmiş

- Mərkəzləşdirilmiş paylanmış sistemdə bir və ya bir neçə qovşaq, digərlərinə nisbətən, daha geniş idarə edici funksiyalara malikdir. Mərkəzləşdirilmiş paylanmış sistem mərkəzləşdirilmiş idarə etməni, izafiliyin aradan qaldırılmasını, verilənlərə ümumi baxışı təmin edir. Lakin bu cür təşkil olunmuş sistemlərdə mərkəzi qovşaq və ya əlaqə kanalları işdən çıxdıqda müəyyən çətinliklər yaranır.
- Mərkəzləşdirilməmiş sistemlərdə şəbəkələrin qovşaqlarında yerləşmiş lokal sistemlər ümumi məsələnin həlli zamanı, funksionallıq nöqtəyi-nəzərdən, bərabər hüquqludurlar. Mərkəzləşdirilmənin olmaması verilənləri istifadəçilərə yaxınlaşdırmağa imkan verir, emalı tezləşdirir, verilənlərin təkrarlanması hesabına etibarlılığı artırır.



Müasir PS-lərə qoyulan tələblər

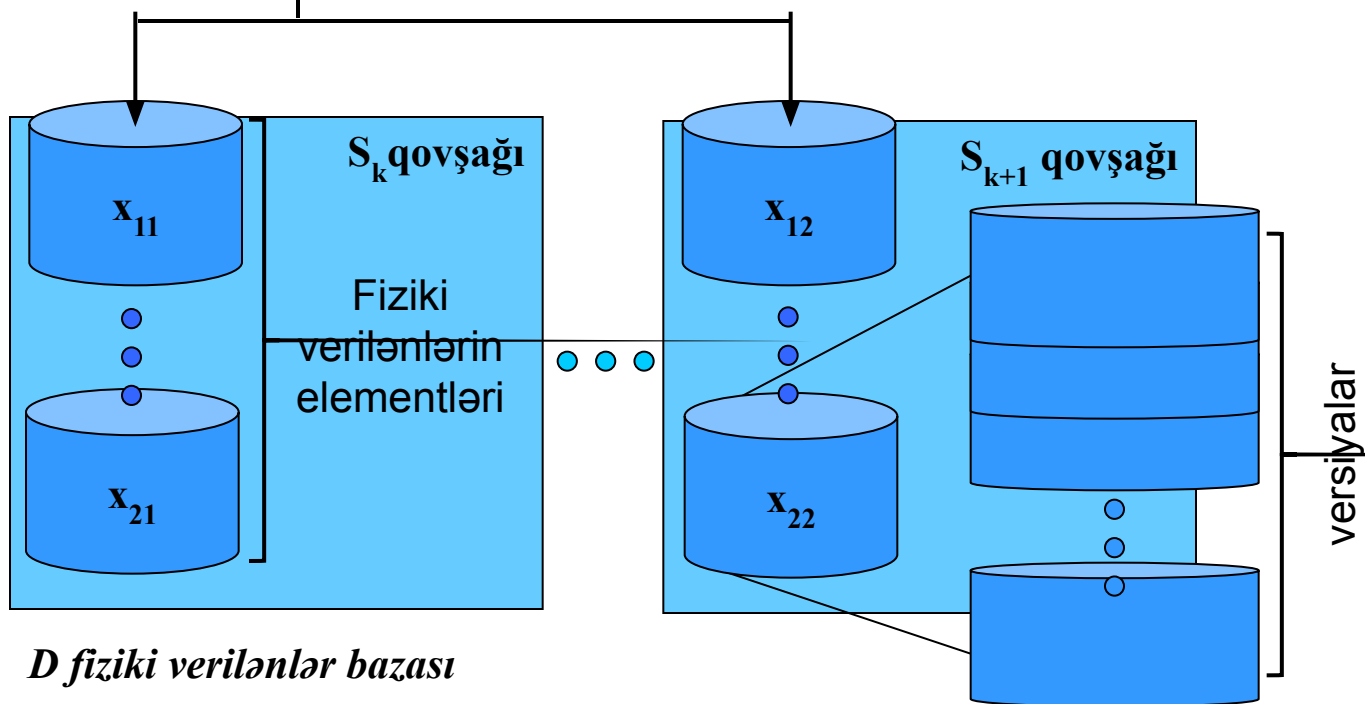
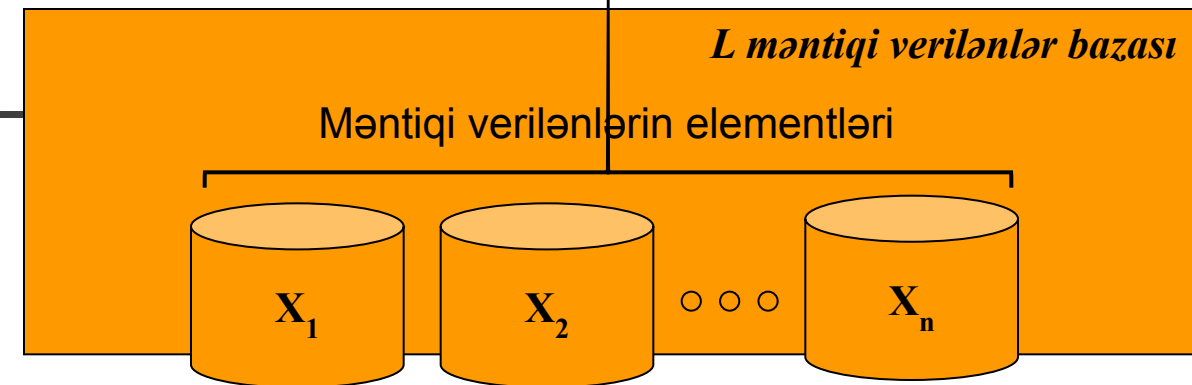
- yüksək etibarlılıq;
- informasiya resurslarına kollektiv müraciət;
- icazəsiz müraciətdən verilənlərin qorunması;
- sistemin həm texniki, həm də informasiya-hesablama resurslarının maksimal dərəcədə istifadəsi;
- sistemə məsafədən və cəld müdaxilə;
- istifadəçilərin sistemlə asan ünsiyyəti;
- arxitekturanın açıq olması (istənilən zaman genişlənmə imkanı)

Paylanmış verilənlər bazası sistemləri



- PS-in qovşaqlarında verilənlər bazalarının lokal idarə etmə sistemləri (VBİS) tərəfindən nəzarət olunan, verilənlər bazaları şəklində təşkil olunmuş informasiya saxlanılır və yenilənir.
- PS-də qovşaqlar, VBİS-in xidmətlərini istifadəçiyə təqdim edərək, bir-birilə qarşılıqlı əlaqə yarada bilirlər.
- Bu halda, PS paylanmış verilənlər bazası (PVB) sistemi adlanır.

PVB-nin modeli





PVB sisteminin modeli

- PVB sisteminin modeli üçlüklə təyin edilir:
- Tranzaksiyalar çoxluğu
- Korrektlik kriterisi olub korrekt yerinə yetirilən tranzaksiyalar çoxluğudur

$$(PVB, \sigma, K(\sigma))$$

$$\sigma = (T_1, T_2, \dots, T_m)$$

$$K(\sigma)$$

PVB – verilənlər çoxluğu olub, hər bir verilən elementinin adı və qiyməti vardır.

Istifadəçilər tərəfindən görünən elementlər məntiqi verilən elementləri olub **məntiqi verilənlər bazasını** təşkil edir.

$$L = (X_1, X_2, \dots, X_n)$$

- Hər bir X_i məntiqi elementinə fiziki elementlər çoxluğu uyğundur
- Fiziki elementlər çoxluğunun sürətləri PVB-nin qovşaqlarında yerləşir
- PVB-nin bütün fiziki elementlər çoxluğu **fiziki verilənlər bazasını** təşkil edir.

$$x_{i1}, x_{i2}, \dots, x_{ic_i}$$

$$S = (S_1, \dots, S_N)$$

$$F = (x_{11}, x_{12}, \dots, x_{1c_1}, \dots, x_{i1}, x_{i2}, \dots, x_{ic_i}, \dots, x_{n1}, x_{n2}, \dots, x_{nc_n})$$



PVB-də replikasiya

- F fiziki verilənlər bazasının hər hansı x elementinin sürəti başqa bir qovşaqda yoxdursa, o zaman **PVB-də verilənlərin replikasiyası yoxdur**
- F fiziki verilənlər bazasının hər bir x elementinin sürəti hər bir qovşaqda mövcuddursa, o zaman **PVB-də verilənlərin tam replikasiyası vardır**
- F fiziki verilənlər bazasının hər bir x elementinin sürəti bir neçə qovşaqda mövcuddursa, o zaman **PVB-də verilənlərin qismən replikasiyası vardır**



Monovers və multivers PVB

- Hər bir fiziki x elementi adətən bir neçə versiyaya malik olur, versiya yalnız birdirsə, fiziki element **monovers**, əks halda **multivers** adlanır
- PVB-nin hər bir elementi monoversdirsə - **monovers (monoversion) PVB**, əks halda **multivers (multiversion) PVB** adlanır

$$x = \langle x^0, x^1, \dots, x^g \rangle$$



PVB-nin elementləri

- Məntiqi verilənlər bazası

$$L = (X_1, X_2, \dots, X_n)$$

- Fiziki verilənlər bazası

$$F = (x_{11}, x_{12}, \dots, x_{1c_1}, \dots, x_{i1}, x_{i2}, \dots, x_{ic_i}, \dots, x_{n1}, x_{n2}, \dots, x_{nc_n})$$

- Qovşaqlar çoxluğu

$$S = (S_1, \dots, S_N)$$

- Hər bir fiziki elementin hansı qovşaqda olduğunu təyin edən funksiya

$$H: F \rightarrow S$$



PVBİS

- S_k qovşağında yerləşən bütün fiziki elementlər çoxluğu **lokal verilənlər bazasını** təşkil edir
- Hər bir lokal verilənlər bazası lokal verilənlər bazasının avtonom idarəedilməsi sistemi – LVBİS, PVB isə PVBİS ilə idarə edilir.

PVB-nin ardıcılığı bu hallarda pozula bilər:

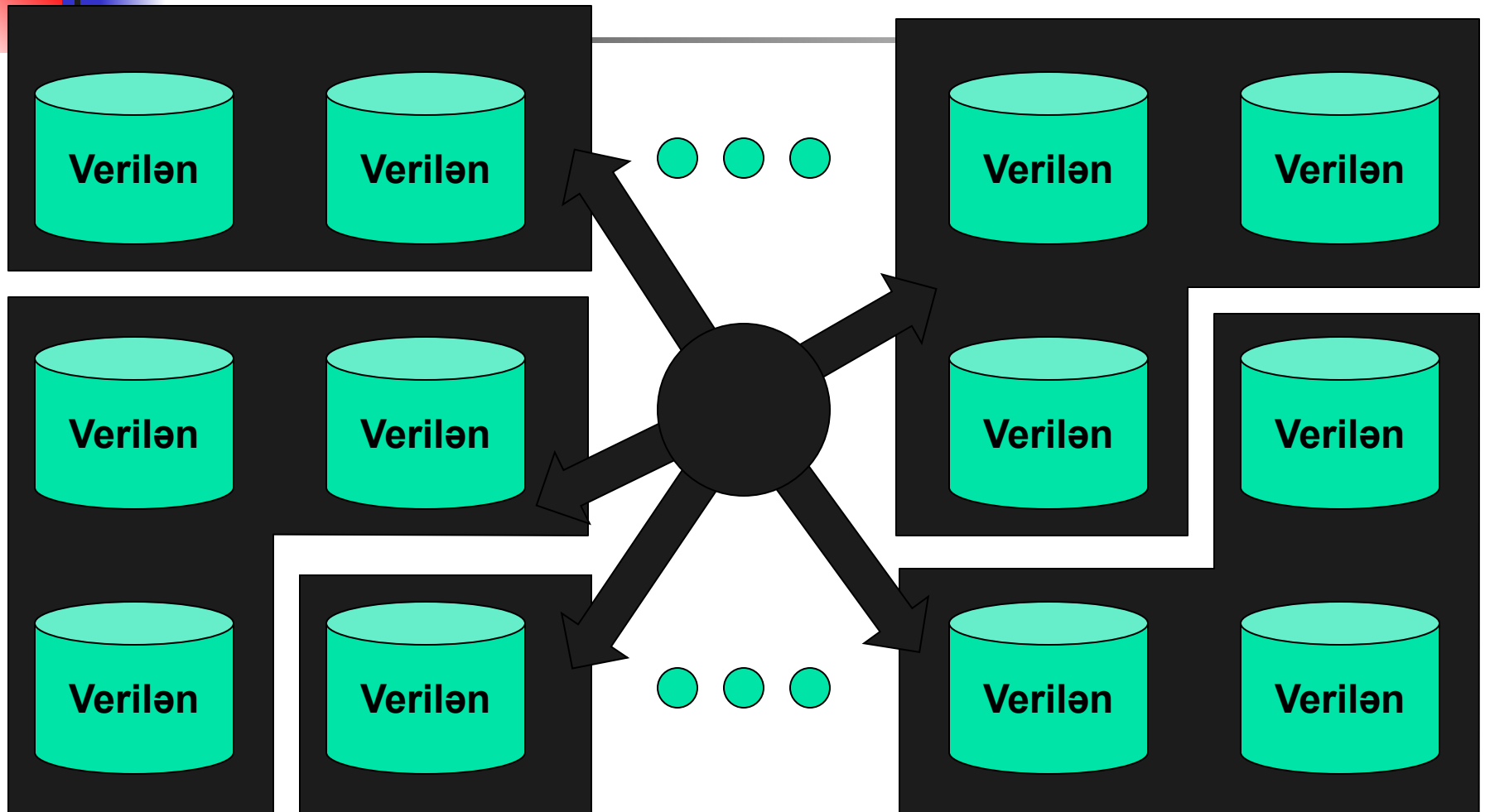
- Verilən elementlərinə müraciət tələb edən proqramın korrekt olmayan paralel idarəsi zamanı
- Sistemin qismən və ya tam olaraq yararsız hala gətirən aparat və proqram təminatı səhvləri zamanı

$$LDB_k = \{x_{ij} : x_{ij} \in F \wedge H(x_{ij}) = S_k\}$$

PVB-nin idarə edilməsində əsas məsələlər

- **Paralelliyyənin idarə edilməsi.** paylanmış verilənlər bazasına paralel müraciətinin korrekt şəkildə idarə edilməsi uyğun alqoritmlərin işlənməsi deməkdir. Paralelliyyənin idarə edilməsi alqoritmləri PVB-də münaqişələrin qarşısının alınması üçün nəzərdə tutulublar. Bu zaman hesablamaların və zaman sərfinin azaldılması, PVB sisteminin məhsuldarlığının artırılması tələb olunur.
- **Sorguların yerinə yetirilməsinin optimallaşdırılması**
- **Etibarlılığın təmin edilməsi.** Tətbiqi proseslərin paralel işləməsi zamanı PVB-nin tərkibində ziddiyyət yarana bilər. Ona görə də, PVB-nin vacib komponenti sayılan tranzaksiyaların idarə edilməsi sistemi tətbiqi proseslərin paralel yerinə yetirilməsini elə idarə etməlidir ki, paylanmış verilənlər bazasının tamlığı pozulmasın.

PVB-də tranzaksiyaların paralel idarə edilməsi





TRANZAKSİYALAR (transactions)

- Hamar tranzaksiyalar (*flat transaction*)
- İç-içə tranzaksiyalar (*nested transactions*)
- Paylanmış tranzaksiyalar (*distributed transactions*)

Hamar Tranzaksiyalar

(flat transactions)

- **Hamar Tranzaksiya** verilənlər üzərində aparılan bölünməyən əməliyyatlar ardıcılığıdır. Bunları **ACID-tranzaksiyalar** da adlandırırlar.
- **Xassələri:**
 - **Atomarlıq (Atomicity).** Tranzaksiya atomar əməliyyat kimi yerinə yetirilir – tranzaksiya ya tam şəkildə yerinə yetirilir, ya da heç yerinə yetirilmir.
 - **Uyğunluq (Consistency).** Tranzaksiya verilənlər bazasını bir uyğun (ziddiyyətsiz) vəziyyətdən digər uyğun vəziyyətə keçirir.
 - **İzolyasiya (Isolation).** Müxtəlif istifadəçilərin tranzaksiyaları bir-birinə mane olmamalıdır.
 - **Uzun ömürlülük (Durability).** Tranzaksiya yerinə yetirilibsə, onun fəaliyyətinin nəticələri (növbəti anda sistem işdən çıxsə belə) verilənlər bazasında saxlanılmalıdır.

TRANZAKSİYALAR



Tranzaksiya işə başladıqdan sonra aşağıdakı hadisələrdən biri baş verənə kimi davam edir:

- ***Commit work*** (tranzaksiyanı qeyd etməli) əmri verilib.
- ***Rollback work*** (tranzaksiyanı geri çəkməli) əmri verilib.
- İstifadəçinin VBİS-dən açılması baş verib.
- Sistemin işdən çıxması qeyd olunub.



TRANZAKSİYALAR

- ***Commit work*** əmri cari tranzaksiyanı başa çatdırır və avtomatik olaraq yeni tranzaksiyanı başlayır. Bu zaman başa çatmış tranzaksiyanın nəticələrinin qeyd olunmasına, yəni verilənlər bazasında saxlanılmasına zəmanət verilir.
- ***Rollback work*** əmri cari tranzaksiya tərəfindən yerinə yetirilmiş dəyişikliklərin geri çəkilməsi (yəni sanki, heç bir şey olmayıb) vəziyyətini bildirir.
- **İstifadəçinin VBİS-dən açılması** zamanı tranzaksiyanın avtomatik qeyd olunması baş verir.
- **Sistemin işdən çıxması zamanı** daha mürəkkəb proseslər baş verir. Sistemin növbəti işə düşməsi zamanı sistemin işdən çıxması anına qədər yerinə yetirilən tranzaksiyaların təhlili aparılır. ***Commit work* əmri verilmiş**, lakin işlərin nəticələri qeyd olunmamış tranzaksiyalar yenidən yerinə yetirilir. ***Commit work* əmri verilməyən** tranzaksiyalar geri çəkilirlər.

TRANZAKSİYALAR

Tranzaksiyalar iki sinfə bölünürlər:

- Lokal verilənlər bazasında informasiyanı ancaq oxuyan tranzaksiyalar - R
- Lokal verilənlər bazasında informasiyanı dəyişdirən tranzaksiyalar – W
- Əgər tranzaksiyalar zamana görə bir-biri ilə toqquşur və eyni verilənlərə müraciət edirlərsə, onda onlar **rəqabət aparan tranzaksiyalar** adlanırlar. Verilənlər uğrunda tranzaksiyalar arasında yaranan rəqabət nəticəsində verilənlərə müraciətin münaqişəsi yaranır. **Münaqişələrin** aşağıdakı növləri mövcudur:
 - **W – W (yazı – yazı)**. Birinci tranzaksiya obyektini dəyişdirdi və başa çatmadı. İkinci tranzaksiya eyni obyektini dəyişdirməyə cəhd edir. Nəticədə, yenilənmənin itirilməsi baş verir.
 - **R – W (oxuma – yazı)**. Birinci tranzaksiya obyektini oxudu və başa çatmadı. İkinci tranzaksiya eyni obyektini dəyişdirməyə cəhd edir. Nəticədə, uyğun olmayan təhlil yaranır.
 - **W – R (yazı – oxuma)**. Birinci tranzaksiya obyektini dəyişdirdi və başa çatmadı. İkinci tranzaksiya eyni obyektini oxumağa çalışır. Nəticədə, düzgün olmayan verilənlərin oxunması baş verir.
- Oxuma zamanı verilənlər dəyişmədiyi üçün **R – R** növlü münaqişə olmur.

Tranzaksiyaların emalı

- $S = \{S_1, S_2, \dots, S_N\}$ qovşaqlar çoxluğunda hər zaman 2 altçoxluq fəaliyyət göstərir:
 - TM_i (*Transaction meneger*) - tranzaksiyaları inisiyasiya edən (başladan) qovşaqlar altçoxluğu.
 - DM^k (*Data meneger*) – tranzaksiyaları yerinə yetirən (emal edən) qovşaqlar altçoxluğu

Qovşaqlar çoxluğunun bu cür bölünməsi *client-server* rejimində kompüterlərin işlərinin bölünməsinə uyğun gəlir. Axırncı sətir bir qovşağın eyni vaxtda tranzaksiyanı həm başlada, həm də emal edə biləcəyini göstərir.

$$\begin{aligned} \{TM_i\} &\subseteq S \quad \|\{TM_i\}\| = n_{TM} \\ \{DM^k\} &\subseteq S \quad \|\{DM^k\}\| = n_{DM} \\ \{TM_i\} \cap \{DM^k\} &\neq \emptyset. \end{aligned}$$



Tranzaksiyaların emalı

- TM_i qovşağı ilə başlanmış tranzaksiyalar T_i kimi işarə edilir. Ümumi halda T_i tranzaksiyası bir neçə DM^k qovşağında yerinə yetirilir.
- Bir DM^k qovşağında yerinə yetirilən T_i tranzaksiyasının bir hissəsinə alttranzaksiya deyilir və T_i^k kimi işarə edilir.

$$T_i = \left\{ T_i^1, \dots, T_i^{n_i} \right\}$$

Tranzaksiyaların emalı

- Hər bir T_i^k alttranzaksiyası işini başlamazdan əvvəl DM^k qovşağında lokal verilənlər bazasını zəbt etməlidir. Lokal verilənlər bazasını verilmiş qovşağın informasiya resursu adlandırırlar.
- Əgər kəsişmə boş çoxluqdursa, onda T_i və T_j tranzaksiyaları **münaqişə etmirlər**
- Əgər kəsişmə boş çoxluqdan fərqlidirsə, onda tranzaksiyaları DM^k qovşağının resursları uğrunda **münaqişə edən** tranzaksiyalar adlandırırlar

$$\{DM_i^k\} \cap \{DM_j^k\} = \emptyset,$$

$$\{DM_i^k\} \cap \{DM_j^k\} \neq \emptyset,$$



Paylanmış plan

- TM_i qovşaqlarında tranzaksiyaların başlanması prosesinin təsadüfi və müstəqil olduğu üçün DM^k qovşaqlarında emal olunmanı gözləyən alttranzaksiyaların növbələri yarana bilər. Bütün növbələr toplusu (onlardan bəziləri boş ola bilər) alttranzaksiyaların yerinə yetirilməsinin **paylanmış planı** (*Distributed Subtransaction Ordering, DSO*) adlanır.
- Paylanmış sistem çərçivəsində tranzaksiyaların paralel emalının əlaqələndirilməsi, TM_i və DM^k qovşaqlarının hər birində öz agentləri olan, tranzaksiyaların idarə edilməsi sistemi (TİS) ilə yerinə yetirilir. Tətbiq olunan VBİS-dən asılı olmayaraq TİS bir proqramdır.



TİS-ə qoyulan tələblər

- **Atomarlıq.** Tranzaksiya (yəni onun alttranzaksiyalar çoxluğu) tam şəkildə yerinə yetirilməlidir, əgər bu mümkün deyilsə, onda PVB-də onun emalına aid heç bir iz qalmasın deyə müəyyən tədbirlər görülməlidir.
- **Statistik ədalət.** Başlanmış bütün tranzaksiyalara qarşı statistik ədalət prinsipi təmin olunmalıdır. Yəni hər hansı bir TM_i qovşağı ilə başlanmış tranzaksiya münasib vaxt ərzində yerinə yetirilməlidir. Qeyd etmək lazımdır ki, rəqabət aparan tranzaksiyaların heç biri hər hansı bir üstünlüyə malik olmamalıdır.
- **Seriallanma.** Seriallanma prinsipi eyni zamanda müraciət vaxtı korrektlik meyarı kimi təqdim oluna bilər. Bu prinsipə görə tranzaksiyalar çoxluğunun eyni zamanda (paralel) yerinə yetirilməsindən alınan effekti, onların ardıcıl yerinə yetirilməsi effektinə bərabər olmalıdır.
- **Məhsuldarlıq.** Müxtəlif DM^k qovşaqlarında alttranzaksiyaların yüksək dərəcədə paralel yerinə yetirilməsi hesabına paylanmış sistemlərin məhsuldarlığı təmin olunmalıdır.
- **Nasazlığa dayanıqlılıq.** Nasazlığa dayanıqlı sistem dedikdə, kompüterlərin müxtəlif nasazlıqları zamanı sistemin fəaliyyətini davam etdirə bilən sistem başa düşülür.

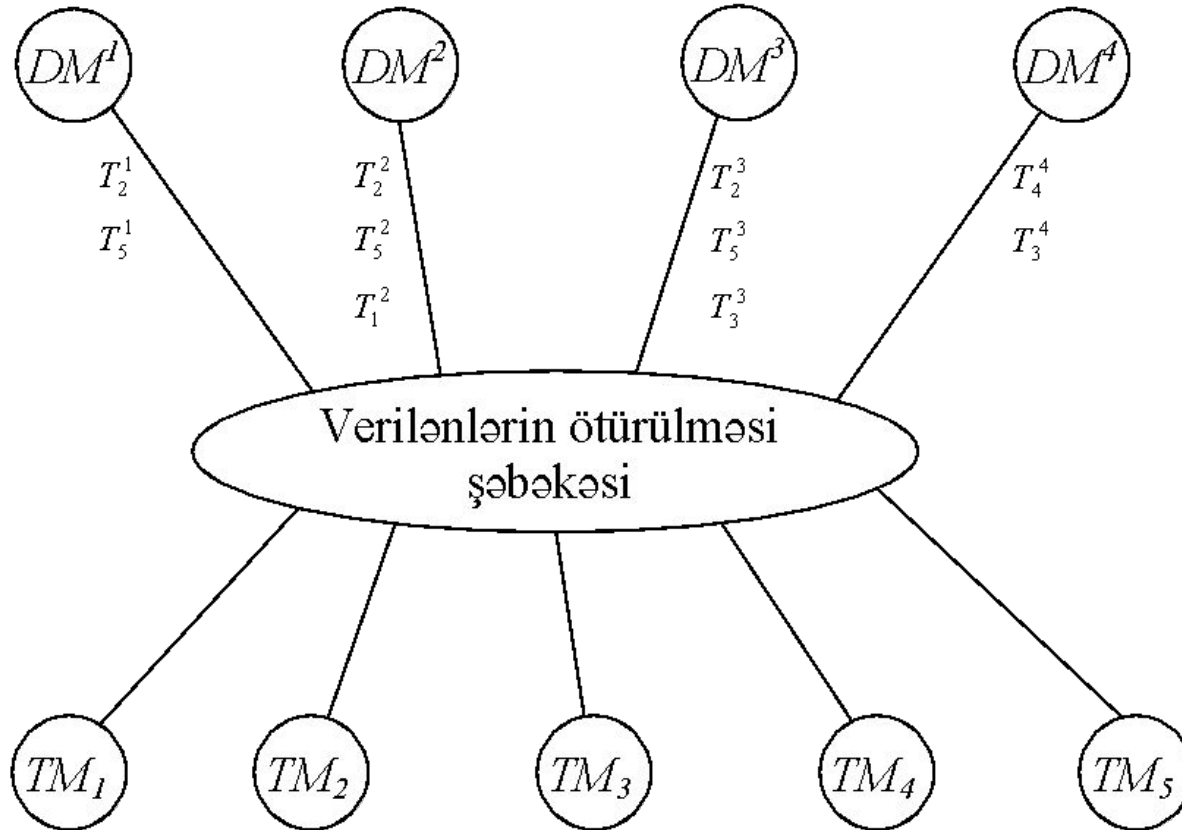
Tranzaksiyaların paralel yerinə yetirilməsi

- Tutaq ki, dörd DM^1 - DM^4 qovşağında beş TM_1 - TM_5 qovşağının hərəsi bir tranzaksiyanı inisiasiya edib (başlayıb)
- Müvəqqəti olaraq qəbul edək ki, DM^k qovşaqlarında alttranzaksiyaların növbələrə düzülməsi *FIFO* (*first-in, first-out*, gəlmə qaydasına görə, birinci gələn birinci emal olunur) rejimində baş verir.

T_2, T_4, T_5, T_3, T_1

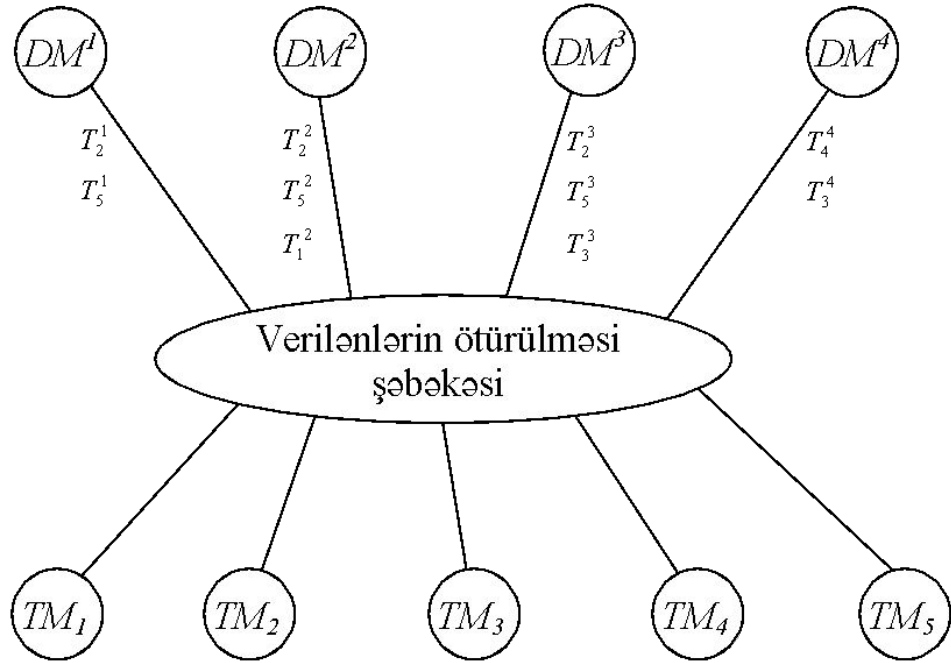
$$T_1 = \{T_1^2\},$$
$$T_2 = \{T_2^1, T_2^2, T_2^3\},$$
$$T_3 = \{T_3^3, T_3^4\},$$
$$T_4 = \{T_4^4\},$$
$$T_5 = \{T_5^1, T_5^2, T_5^3\}$$

DM^k qovşaqlarında yaranmış növbələr

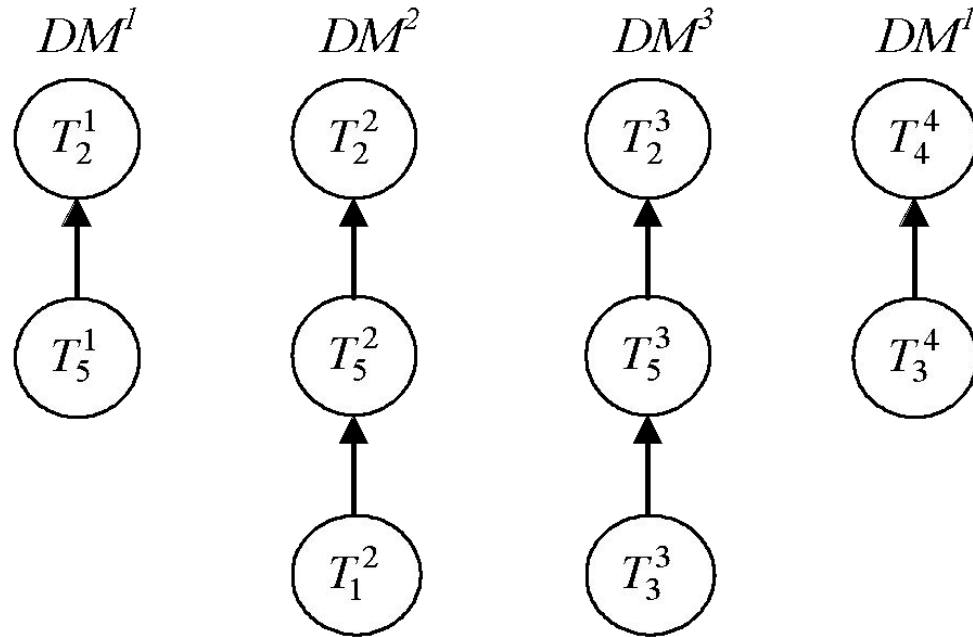


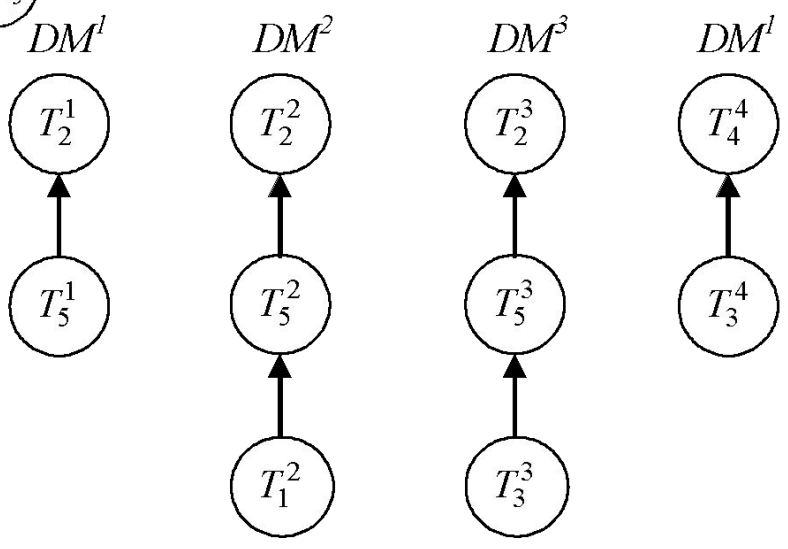
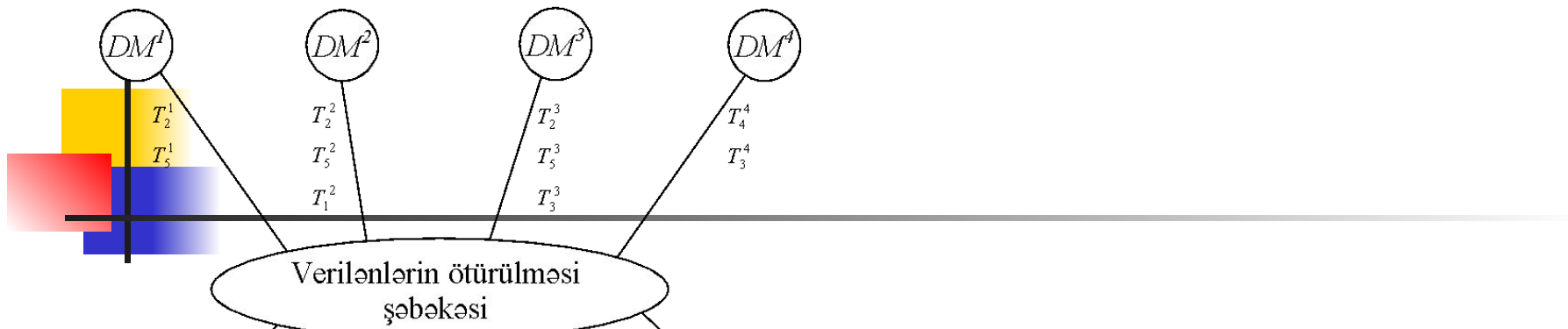
DM^k qovşaqlarında yaranmış növbələr

$$T_1 = \{T_1^2\},$$
$$T_2 = \{T_2^1, T_2^2, T_2^3\},$$
$$T_3 = \{T_3^3, T_3^4\},$$
$$T_4 = \{T_4^4\},$$
$$T_5 = \{T_5^1, T_5^2, T_5^3\}$$



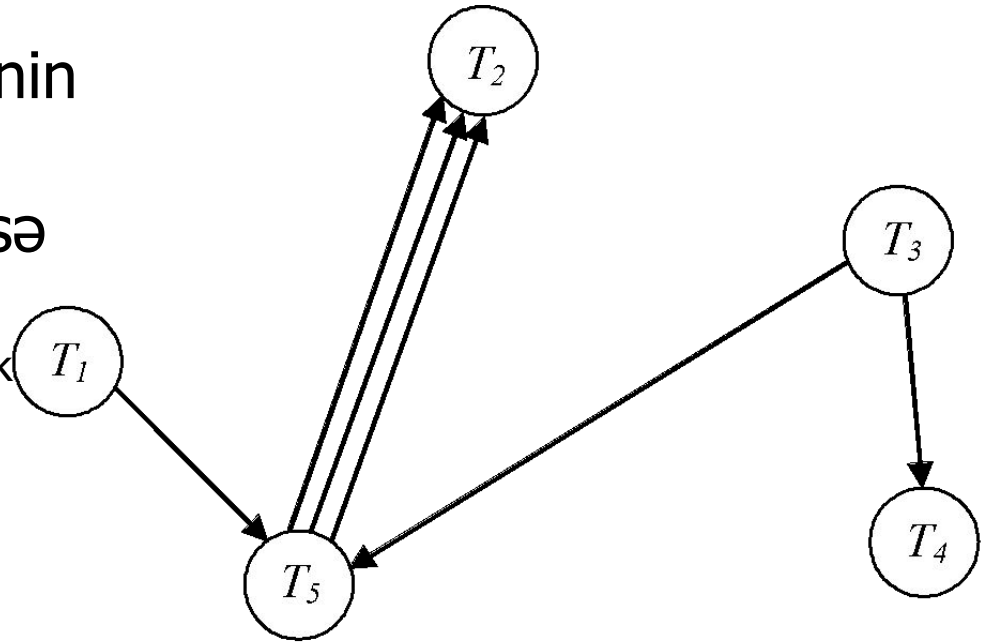
DM^k qovşaqlarında yaranmış növbələrə uyğun Q – qrafı (gözləmə qrafı)





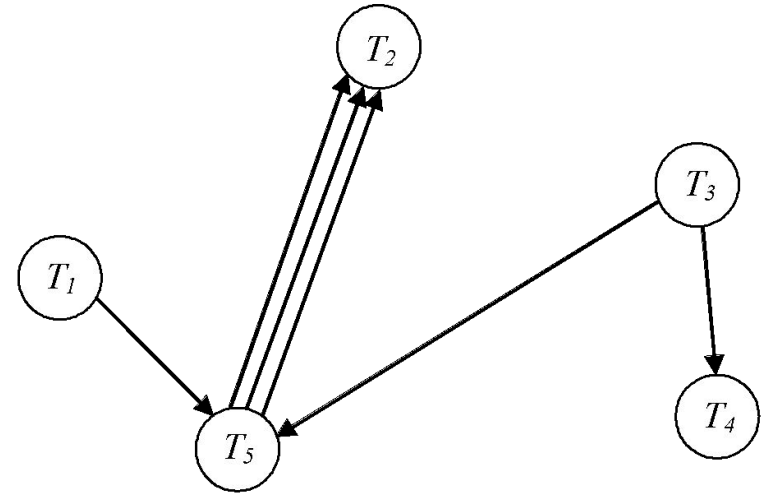
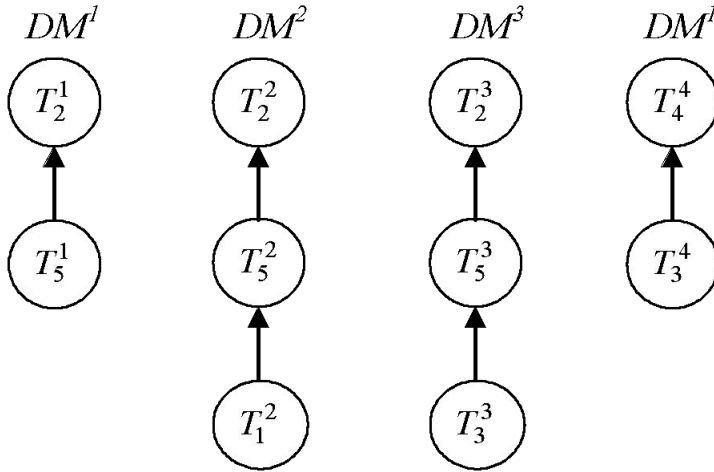
DM^k qovşaqlarında yaranmış növbələrə uyğun D – qrafı (asıllıqlar qrafı)

- D -qrafın təpə nöqtələrinin sayı TM_i qovşaqlarının sayına, qövsələrin sayı isə aşağıdakı kimi təyin olunur, burada l^k – DM^k qovşağındakı növbənin uzunluğudur

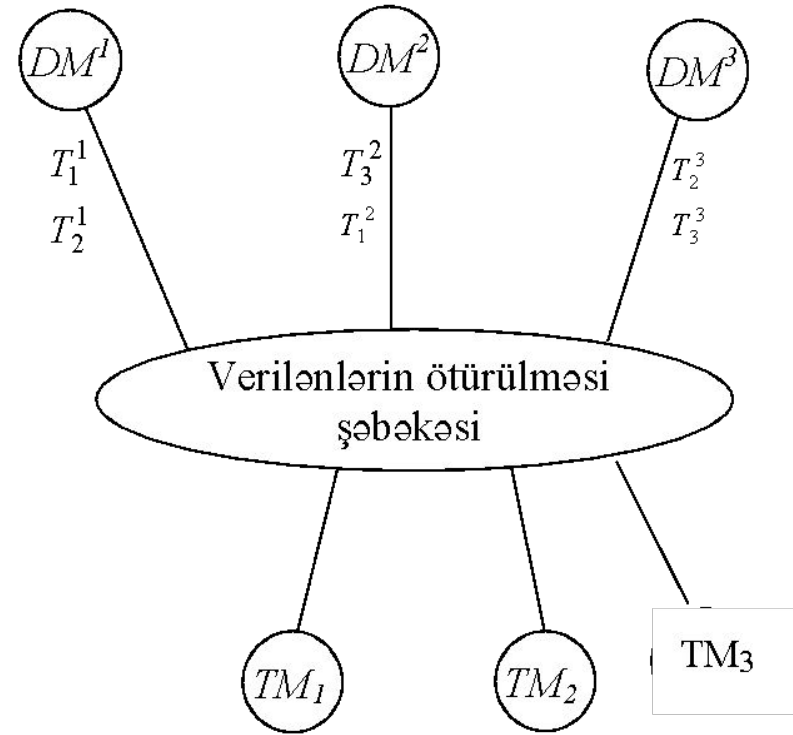


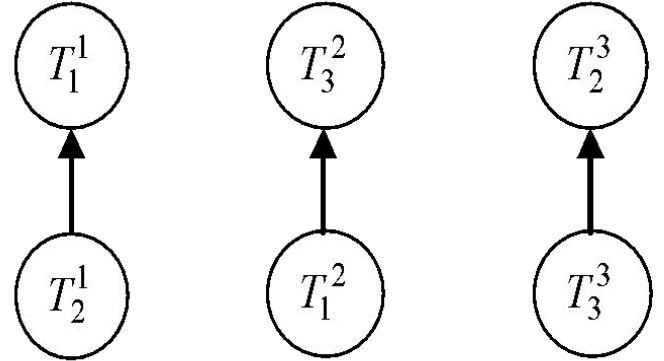
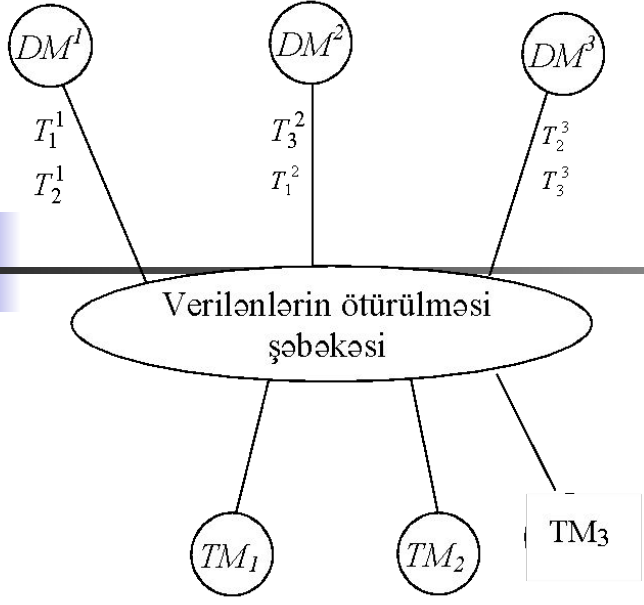
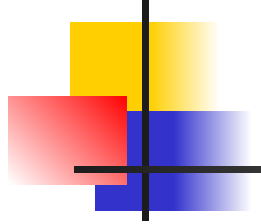
$$\sum_{k=1}^{n_{DM}} (l^k - 1)$$

Dövrü olmayan D-qrafı



$$T_1 = \{T_1^1, T_1^2\},$$
$$T_2 = \{T_2^1, T_2^3\},$$
$$T_3 = \{T_3^2, T_3^3\}.$$





Dövrü olan D-qraf

