

# پروتکل قفل دو مرحله ای (2PL):

پروتکل قفل دو مرحله ای تضمین می کند که زمانبندی، سریال پذیرد  
برخورد باشد و مشکلات همروندی را نیز حل کند.

مراحل:

1) مرحله رشد (growing) : در این مرحله تراکنش فقط می تواند فقط قفل بگیرد اما نمی تواند قفل را آزاد کند.

2) مرحله نقصان یا عقب نشینی (shrinking) : در این مرحله تراکنش فقط می تواند قفل را آزاد کند ولی نمی تواند روی هیچ داده ای قفل بگیرد.

# پروتکل قفل دو مرحله ای (2PL): (ادامه)

- این پروتکل تراکنش های سریالی را کنترل می کند.
- قفل دو مرحله ای خروج از بن بست را تضمین نمی کند.
- عقبگرد آبشاری توسط قفل دو مرحله ای امکان پذیر است.
- قفل دو مرحله ای بسیار دقیق است.

# پروتکل قفل دو مرحله ای پایه (B2PL):

در این پروتکل، تراکنش ها شروع به گرفتن قفل های مورد نیاز و اجرای دستورات خود می کنند. به محض اینکه یکی از تراکنش ها قفلی را آزاد کرد، وارد مرحله دوم می شود که از این پس، هیچ قفلی را نمی تواند بگیرد.

نکته:

- این پروتکل تضمین می کند که تراکنش ها به ترتیب نقطه قفل سریال پذیر شوند.
- در این پروتکل امکان بروز بن بست وجود دارد.

# پروتکل قفل دو مرحله ای محافظه کارانه (C2PL):

در این پروتکل، هر تراکنش قبل از این که شروع به اجرای اولین دستورش بکند، باید تمام قفل های مورد نیازش را گرفته باشد و اگر موفق نشد، دوباره در صف قرار می گیرد.

این پروتکل تضمین می کند که هیچ وقت بن بست رخ ندهد. اما سطح همروندی تراکنش ها را پایین می آورد.

# پروتکل های قفل دومرحله ای محض (S2PL) :

برای تضمین عدم وقوع سقوط های آبخاری، بازکردن قفل های انحصاری تا بعد از اتمام تراکنش به تعویق می افتد. قفل های اشتراکی می توانند کمی زودتر باز شوند. در بقیه موارد شبیه B2PL کار می کند.

مزیت:

- تضمین سریال پذیری و ترمیم پذیری
- کم کردن پیام ها در بانک های اطلاعاتی نامتمرکز

## پروتکل SC2PL:

از ترکیب دو پروتکل C2PL و S2PL این پروتکل به وجود می آید. در این پروتکل قفل کردن داده ها طبق C2PL و آزاد کردن قفلها طبق S2PL می باشد.

# پروتکل های مبتنی بر گراف:

در این پروتکل ها مجموعه تمام داده های مورد نظر  $D=\{d_1, d_2, \dots, d_n\}$  به صورت یک گراف جهت دار فاقد حلقه در می آیند. به این گراف، گراف بانک اطلاعات گفته می شود. در این گراف اگر بین هر دو گره  $d_i$  و  $d_j$  لبه وجود داشته باشد  $(d_i \rightarrow d_j)$ ، آنوقت هر تراکنش که می خواهد از داده  $d_j$  استفاده کند باید قبل از  $d_j$  داده  $d_i$  را نیز قفل کند.

برای هر تراکنش، در شروع گرفتن قفل، داشتن قفل برای پدر ضروری نیست.

## پروتکل های مبتنی بر گراف : (ادامه)

- در پروتکل های درختی، سریال پذیری در برخورد و فاقد بن بست بودن را تضمین می کنند.
- افزایش سربار اضافی
- کاهش سطح همروندی
- مشکل سقوط های آبخاری وجود دارد.



# پروتکل های مبتنی بر مهر زمانی:

در این پروتکل ها به هر تراکنش به محض ورود، یک مهر زمانی تصاعدی داده می شود.

مهر زمانی تراکنش  $T_i$  با  $TS(T_i)$  نشان داده می شود.

پروتکل های مبتنی بر مهر زمانی تراکنش ها را به صورت سریال پذیر اجرا می کند.

## انتخاب مهر زمانی :

در سیستم های متمرکز، مهر زمانی با ساعت سیستم تولید می شود.

در سیستم های نامتمرکز، از یک سایت برای تولید مهر زمانی استفاده می شود و یا از زوج مرتب مهر زمانی محلی سایت و شناسه خود سایت.