

## ДӘРІС

**Тақырыбы:** “Құрылымдар, ұсыну деңгейлері және деректерге қол жеткізу әдістері”

**"АЖ деректер базасы" пәні**

**"Ақпараттық жүйелер" білім беру бағдарламасы**

**Авторлары:** АЕЖ каф. аға оқытушысы Б.М.Саданова  
АЕЖ каф. аға оқытушысы Г.Б.Абилдаева

# Дәріс мақсаты:

Мәліметтерді ұйымдастыру тәсілдері туралы түсінік берудің мақсатында компьютерлердің жадында деректерге қол жеткізудің негізгі әдістері, сонымен қатар мәліметтер базасында негізгі операцияларды орындау уақытын едәуір қысқартуға мүмкіндік беретін индекстерді қолдану.

# Дәріс жоспары:

1. Деректер құрылымы, анықтамалары және жіктелуі
2. Сызықтық деректер құрылымы
3. Сызықтық емес мәліметтер құрылымы
4. Деректерді ұсыну деңгейлері
5. Деректерге қол жеткізу әдістері
6. Индекстер, олардың мақсаты
7. Индекс түрлері

# Деректер құрылымы

- **Деректер құрылымы** - бұл белгілі бір операцияларды түсіндіруге және орындауға болатындай етіп ұйымдастырылған мәліметтердің ұйымдық схемасы.
- Деректер құрылымы болуы мүмкін:

***біртекті***

- барлық деректер элементтері бір типтегі жазбалармен ұсынылған

***біртекті емес***

- бір құрылымның элементтері әртүрлі типтегі жазбалар болуы мүмкін

# Деректер құрылымы

- Әр түрлі деректер құрылымдары өз элементтеріне әртүрлі қол жетімділікті қамтамасыз етеді: кейбір құрылымдарда *кез - келген элементке*, басқаларында *қатаң анықталған элементтерге* қол жеткізуге болады.
- Компьютер жадында деректер *тізбекті* немесе *байланысты* көрініске ие болуы мүмкін. Тиісінше, деректерді *тізбекті ұсыну* және *байланысты ұсыну* арқылы сақтау құрылымдарын ажыратады.

# Деректер құрылымы

## Деректерді тізбекті ұсыну :

- компьютер жадындағы деректер іргелес тізбектелген ұяшықтарға орналастырылады
- жазбаларды ұстанудың физикалық тәртібі логикалық құрылыммен анықталған логикалық тәртіпке толық сәйкес келеді
- бірізді жад ұяшықтарында орналастырылған жазбалар жиынтығы *тізбекті тізімдер* деп аталады.

## Деректерді байланысты ұсыну:

әрбір жазбада көрсеткіш (сілтеме) орналастырылатын қосымша жол көзделеді.

жазбалар кез келген бос ұяшықтарда орналасады және осы жазудан кейін қисынды түрде жазбаның орналасқан жерін көрсететін көрсеткіштермен байланысады.

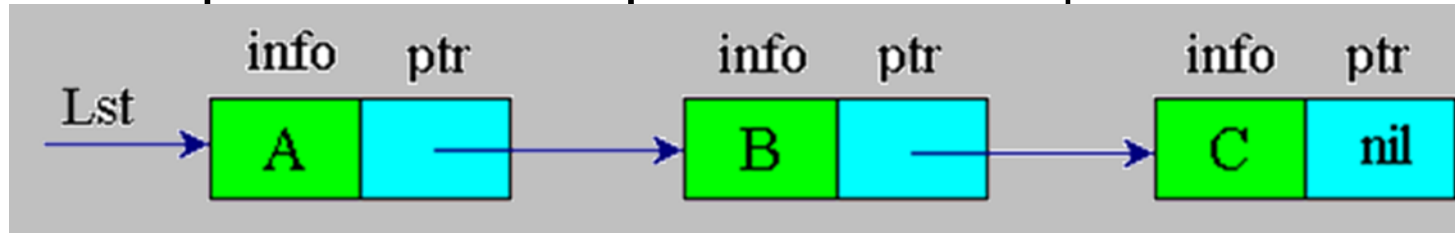
байланысты деректерді ұсынуға негізделген сақтау құрылымдары *байланысты тізімдер* деп аталады.

# Деректер құрылымы

- байланысты деректерді ұсынуға негізделген сақтау құрылымдары *байланысты тізімдер* деп аталады.
- егер әрбір жазбада тек бір көрсеткіш болса, онда тізім *бір-бірімен байланысты*, көп көрсеткіштермен-тізім *көп байланысты*

# Деректер құрылымы

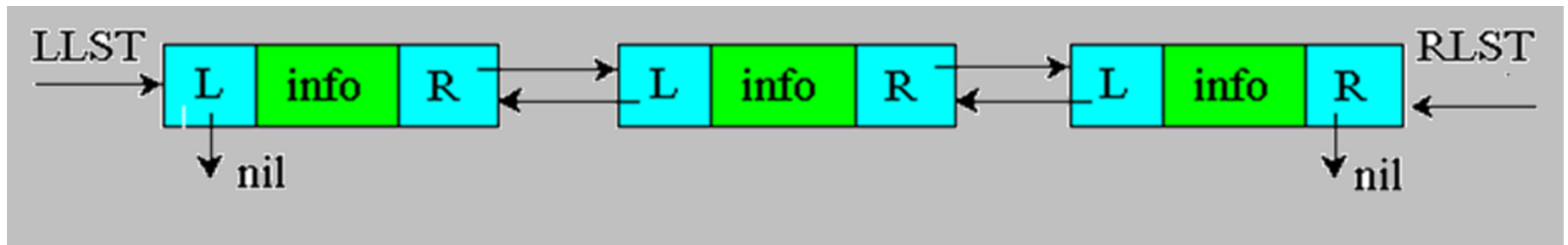
Бір байланыстырылған тізімнің мысалы



- Бір байланыстырылған тізім элементі екі өрістен тұрады: ақпарат өрісі (info) және көрсеткіш өрісі (ptr)
- Көрсеткіш тек тізімнің келесі элементінің мекенжайын береді
- Тізімдегі соңғы элементтің көрсеткіш өрісі бос (NIL)
- Lst - тізімнің басындағы көрсеткіш. Тізім бос болуы мүмкін, онда LST = NIL
- Тізім элементіне кіру тек оның басынан бастап жүзеге асырылады, бұл тізімде кері байланыс жоқ

# Деректер құрылымы

Көп байланысты тізімінің мысалы:



- Екі байланысты тізімде кез-келген элементтің екі көрсеткіші бар
- Біреуі алдыңғы (сол) элементті (L), екіншісі келесі (оң) элементті (R) көрсетеді

# Деректер құрылымы



# Сызықтық (статикалық) мәліметтер құрылымы

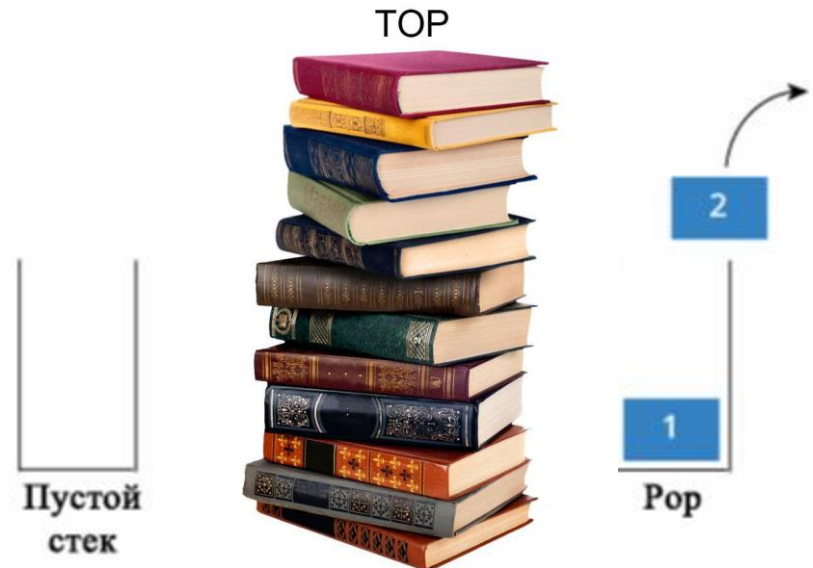
## Массив

- деректердің тізбекті көрсетілімін қолдана отырып жүзеге асырылатын бекітілген көлемдегі деректердің сызықтық құрылымы.
- Массивтің әр элементі бір немесе бірнеше индекстермен анықталады.
- *Индекс* дегеніміз - мәні массивтегі тиісті элементтің орнын анықтайтын және осы элементке қол жеткізу үшін қолданылатын бүтін сан.

# Сызықтық (статикалық) мәліметтер құрылымы

## Стек

- Айнымалы өлшемдегі сызықтық құрылым.
- элементтері қосу немесе шығару мүмкіндік береді, яғни бағдарламаны орындау кезінде деректер дестесін көлемі динамикалық өсуі және қысқартылуы мүмкін
- Ақпарат "соңғы келді, бірінші кетті" (LIFO) қағидаты бойынша өңделеді.



# Сызықтық (статикалық) мәліметтер құрылымы

## Кезек

- Айнымалы өлшемдегі сызықтық құрылым.
- Элементтерді кезектен алып тастауға бір аяғынан бастап - кезектің басынан бастап рұқсат етіледі.
- Элементтерді қосу тек қарама - қарсы бағытта-кезектің соңында мүмкін болады.
- Деректер "бірінші келді, бірінші кетті" (FIFO) қағидаты бойынша олардың түсу ретімен өңделеді.



# Сызықтық (статикалық) мәліметтер құрылымы

## Кесте

- Деректердің сызықтық құрылымы, оның әр элементі белгілі бір кілт мәнімен сипатталады және оның элементтеріне кілт арқылы қол жеткізіледі
- *Кілт* - жазбаны бірегей анықтайтын өрісі немесе өрістер жиынтығы

| Идентификатор | Тегі    | Аты     | Әкесінің аты |
|---------------|---------|---------|--------------|
| 232           | Иванов  | Иван    | Иванович     |
| 453           | Петров  | Петр    | Петрович     |
| 545           | Сидоров | Савелий | Сергеевич    |

Кілт әр элемент үшін ерекше

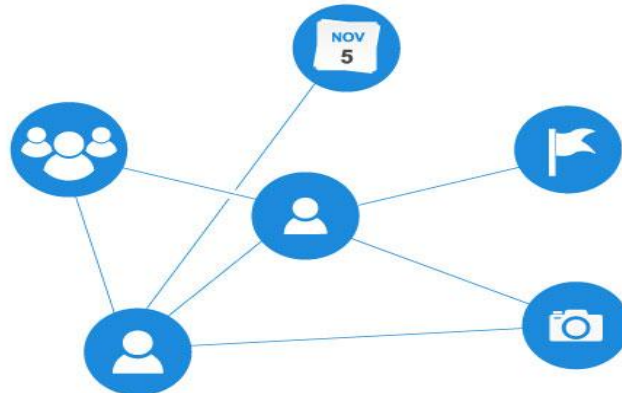
# Сызықтық емес мәліметтер құрылымы

- *"Бір-көпке"* қарым-қатынасы иерархиялық және ағаш тәрізді құрылымдармен көрінеді.
- *«Көп-көпке»* қатынастар әмбебап сипатқа ие және графикалық құрылымдармен көрінеді.

# Сызықтық емес мәліметтер құрылымы

## Граф

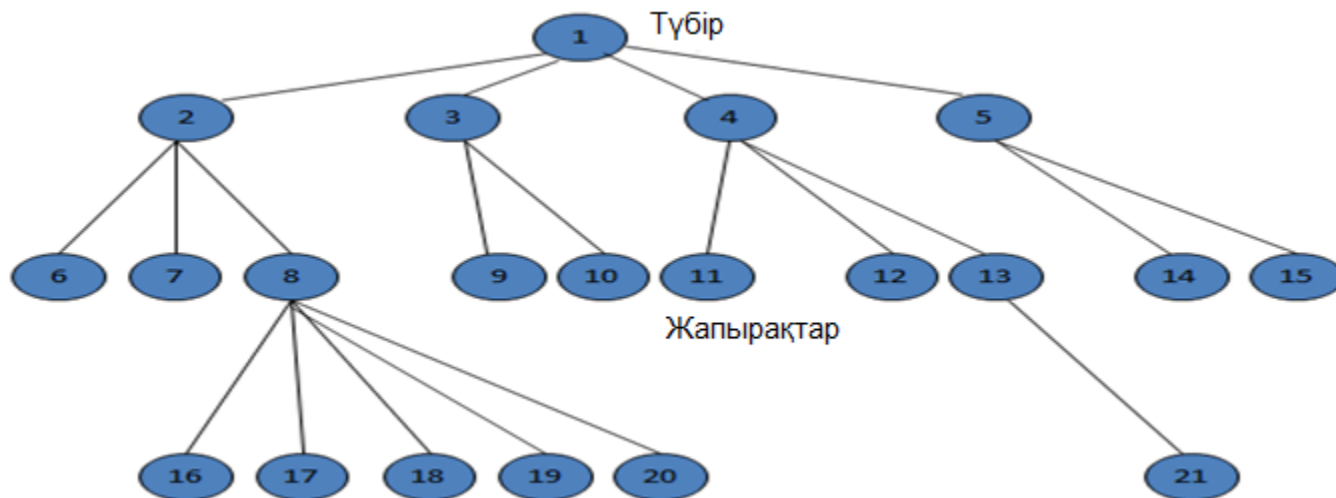
- Жалпы көрініс графы бірнеше шыңдардан (түйіндерден) және шыңдардың жұптарын байланыстыратын жиектерден тұрады. Егер "жиек" және "шың" ұғымдарына белгілі бір семантикалық жүктеме салынса, онда графтар деректерді ұсыну үшін пайдалануға болады.
- Графтың шыңдарын белгілі бір объектілермен салыстыруға болады, содан кейін жиектер объектілер арасындағы қатынастарға сәйкес келеді.
- Бағытталған графтың көрінісі бар деректер моделі *желі* деп аталады.



# СЫЗЫҚТЫҚ емес мәліметтер құрылымы

## Ағаш

- Бұл кейбір шектеулері бар граф, яғни циклдары жоқ бағдарланған граф.
- Ағаш түйіндері деңгейлер бойынша ұйымдастырылған және белгілі бір иерархияға бағынады.
- Ағаштың кез - келген түйіні жоғары деңгейдегі жалғыз түйінмен – тудырған - және жақын деңгейдегі  $m$  түйіндерімен - туындаған - байланысты



# СЫЗЫҚТЫҚ емес мәліметтер құрылымы

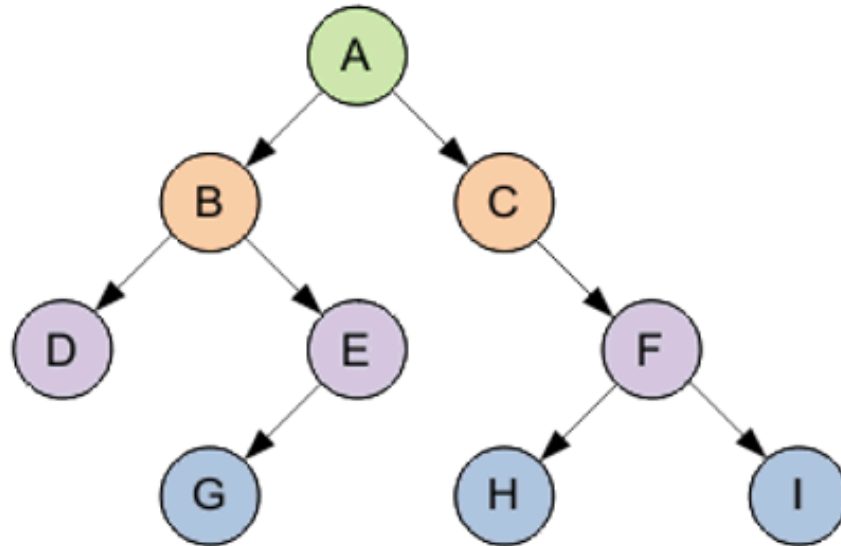
## Ағаш

- Жоғарғы деңгейде ағаштың басында *түбір* аталатын жалғыз түйін бар.
- Ағаштың әр тармағының соңында орналасқан және өсірілмеген түйіндер *жапырақтары* деп аталады.
- Ағаштарда бағыт міндетті түрде тудырған туындыға дейін болады.
- Тамырдан белгілі бір түйінге дейінгі жолдың ұзындығы осы түйіннің *деңгейіне* тең.
- Ағаш деңгейлерінің саны ағаштың *биіктігін* анықтайды.
- Егер ағашта жалпы бастапқы болып табылатын туындаған түйіндер арасында олардың тәртібі маңызды деп саналса, онда ағаш *реттелген* деп аталады.

# Сызықтық емес мәліметтер құрылымы

## Ағаш

- *Бинарлық (екілік) ағаш* - бұл әр шыңның екіден көп емес ұрпағы бар ағаш болып табылатын динамикалық мәліметтер құрылымы.



# Сызықтық емес мәліметтер құрылымы

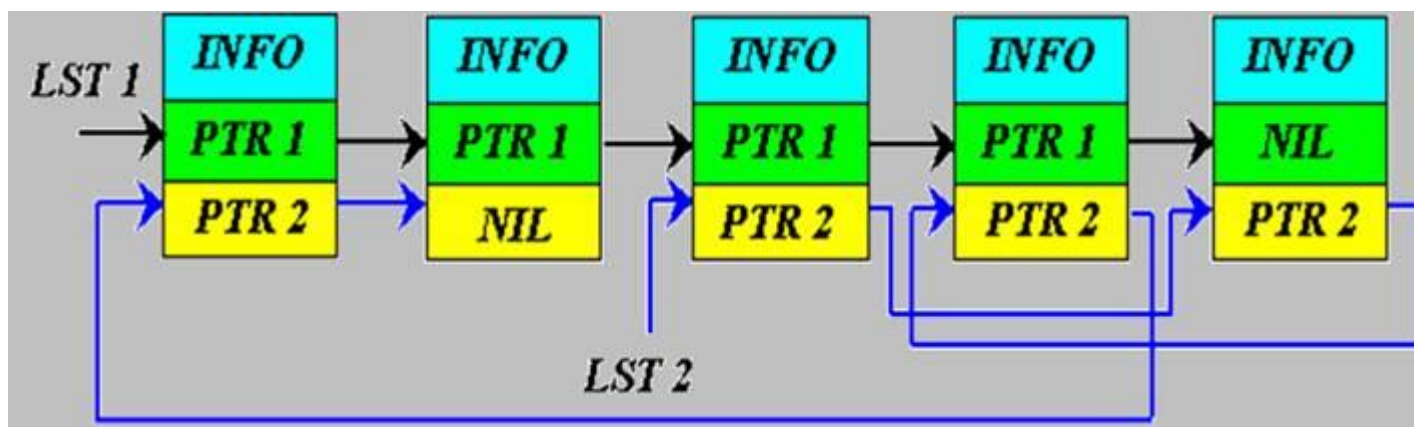
## Тізімдер

- Кез-келген элементтің өзі тізім бола алатын мәліметтер құрылымы *тізім құрылымы* деп аталады.
- Тізім құрылымы граф түрінде кескіндеу болады. Графтың бастапқы шыңы тізімге кіруге сәйкес келеді.
- Графтың кез-келген басқа шыңы *атомдық* немесе *құрылымдық* болып табылады.
- *Атом шыңдары* тізім құрылымының атом элементтеріне сәйкес келеді және объектілер туралы ақпаратты қамтиды.
- Нысандар туралы ақпарат құрылымдық шыңдарында жоқ, олар қосалқы элементтер болып табылатын элементтерге сәйкес келеді.

# Сызықтық емес мәліметтер құрылымы

## Тізімдер

- Компьютер жадында көрініс беру үшін тізім құрылымдары байланысты көріністі қажет етеді.
- Жазбалары негізгі тізімнің элементтеріне айналатын нысандар ерекшеленеді. Қалған объектілер белгілі бір логикалық байланыста болатын негізгі тізімнің элементтерінен тармақталатын екінші деңгейдегі қосалқы жазбаларға топтастырылады.



# Деректерді ұсыну деңгейлері

## 1. Логикалық деңгей:

- деректер базасының объектілері (кестелері) мен олардың сипаттамалары арасындағы нақты қатынастарды көрсететін логикалық деректер құрылымдарымен жұмыс істейді.
- ақпаратты сақтау бірлігі-логикалық жазба.
- жүйенің техникалық және математикалық қамтамасыз етілуі ескерілмейді

## 2. Сақтау деңгейі:

- сақтау құрылымдарымен, яғни компьютердің жадындағы деректердің логикалық құрылымының көріністерімен жұмыс істейді.
- ақпаратты сақтау бірлігі - логикалық жазба.
- бірдей логикалық деректер құрылымын компьютердің жадында әртүрлі сақтау құрылымдары жүзеге асыра алады (мысалы, массив, жазу, кесте).

# Деректерді ұсыну деңгейлері

## 3. Физикалық деңгей:

- физикалық деректер құрылымымен жұмыс істейді.
- компьютердің жадында сақтау құрылымын іске асыру мәселесі шешіледі.
- ақпаратты сақтау бірлігі - физикалық жазба, яғни бір немесе бірнеше логикалық жазбалар орналасқан жеткізгіш бөлімі

# Деректерді ұсыну деңгейлері

## Деректерден физикалық тәуелсіздік:

- дегеніміз, деректердің физикалық орналасуындағы немесе дерекқордың техникалық қамтамасыз етуіндегі кез-келген өзгерістер логикалық құрылымдар мен қолданбалы бағдарламаларда көрінбеуі керек, яғни оларда өзгерістер тудырмауы керек.

## Деректерден логикалық тәуелсіздік :

- бұл сақтау құрылымдарындағы өзгерістер логикалық деректер құрылымында және қолданбалы бағдарламаларда өзгеріс тудырмауы керек дегенді білдіреді.

# Деректерді ұсыну деңгейлері

## Виртуалдық мәліметтер:

- тек логикалық деңгейде бар.
- пайдаланушы үшін бұл деректер шынымен бар болып көрінеді және қажет болған жағдайда олармен жұмыс істейді.
- осы мәліметтерге жүгінген сайын, жүйе белгілі бір жолмен оларды жүйеде физикалық бар басқа мәліметтер негізінде жасайды (мысалы, көрініс View).

## Мөлдір мәліметтер :

- олар логикалық деңгейде жоқ болып көрінеді.
- логикалық деректер құрылымын физикалық (мысалы, индекстер) түрлендіруде қолданылатын көптеген күрделі механизмдерді пайдаланушыдан жасыруға мүмкіндік береді.

# Деректерге қол жеткізу әдістері

- Деректерді өңдеу операцияларына мыналар жатады:
  - Іріктеу
  - Өзгерту
  - деректерді қосу
  - деректерді жою
- Бұл операциялар қол жеткізу операциясына негізделген.
- Іздеу тапсырмаларында барлық деректер белгілі бір сәйкестендірумен жадта сақталады деп болжанады.

# Деректерге қол жеткізу әдістері

## Тапсырма

- әрқайсысы бірегей кілт өрісінің мәні бар бірдей жазбалар жиынтығы бар файлға кіруді ұйымдастыру керек

## Ш

Кілттер файлдағы тиісті жазбаларға сілтемелермен бірге сұрыптау және іздеу әрекеттерін жылдам орындауға мүмкіндік беретін басқа құрылымға көшіріледі.

## Ше ба

Деректерге қол жеткізген кезде алдымен осы құрылымда тиісті кілт мәнін табады, содан кейін файлдан жазба онымен бірге сақталған көрсеткіш бойынша алынады.

# Деректерге қол жеткізу әдістері

Кілт бойынша деректерге қол жеткізуді іске асыратын әдістер:

ағаш арқылы іздеу  
әдістері

хэштеу әдістері

Барлық көптеген кілттер алдын-ала белгілі болған кезде қолданылады және өңдеу кезінде оларды жедел жадқа орналастыруға болады. Хэш функциясы ("to hash"-кесу, ұсақтау) жасалады, ол көптеген сілтемелерде көптеген кілттерді көрсетеді. Осындай функцияны қолдана отырып, файлдағы жазу мекенжайын берілген іздеу кілті арқылы есептеуге болады.

# Деректерге қол жеткізу әдістері

## Екілік ағаш арқылы іздеу алгоритмі :

Массив сұрыпталған болуы тиіс. Іздеу кезінде массивті екіге бөлу қолданылады

Алдымен іздеу аргументі тамырдағы кілтпен салыстырылады.

Егер дәлел кілтпен сәйкес келсе, іздеу аяқталады, егер ол сәйкес келмесе, онда дәлел кілттен аз болған жағдайда іздеу **сол жақ** ішкі торда, ал кілт үлкен болған жағдайда **оң жақ** ішкі торда жалғасады.

Деңгейді 1-ге көбейтіп, ағымдағы түйінді тамыр деп санап, салыстыруды қайталаңыз.

# Деректерге қол жеткізу әдістері

## Екілік ағаш арқылы іздеудің мысалы :

- студенттердің аты-жөндері мен үлгерімінің орташа балы көрсетілген
- барлық жазбалардың белгіленген ұзындығы бар, сондықтан жазба нөмірін көрсеткіш ретінде пайдаланаға болады.
- Бұл жағдайда файлдағы жазбаның жылжуы ( $[ \text{жазбаның нөмірі} ] - 1$ ) \*  $[ \text{жазбаның ұзындығы} ]$  ретінде есептеледі.

| Студент   | Балл |
|-----------|------|
| Васильев  | 4,2  |
| Иванов    | 3,4  |
| Кузнецов  | 3,5  |
| Петров    | 3,2  |
| Сидоров   | 4,6  |
| Тихомиров | 3,8  |



# Деректерге қол жеткізу әдістері

## Бинарлық ағаштар

- Бинарлық ағаштар әсіресе көптеген кілттер алдын-ала белгісіз болған кезде немесе бұл жиын қарқынды өзгерген кезде тиімді. Айнымалы кілттердің жиынтығымен теңдестірілген ағаштың болуы жақсы.
- Бинарлық ағаш *теңдестірілген* деп аталады, егер әр түйіннің сол жақ бағанының биіктігі оң жақ бағананың биіктігінен 1-ден аспаса.
- Сыртқы жадта деректерді іздеу кезінде деректердің сыртқы жадтан жедел жадқа ауысу санын азайту мәселесі өте маңызды.

# Деректерге қол жеткізу әдістері

## Бинарлық ағаштар

- *n-ретті B-ағаш*  $2n+1$  дәрежелі қатты тармақталған ағаш деп аталады, ол келесі қасиеттерге ие:
  - ✓ Түбірден басқа әрбір түйін кемінде  $n$  және  $2n$ -ден аспайтын кілттерден тұрады.
  - ✓ Түбірде кемінде бір және  $2n$ -ден аспайтын кілт болады.
  - ✓ Барлық жапырақтары бірдей деңгейде орналасқан.
  - ✓ Әрбір парақ түйінінде екі тізім бар: мәндердің өсуіне қарай реттелген кілттер тізімі және оған сәйкес көрсеткіштер тізімі (Парақ түйіндері үшін көрсеткіштер тізімі жоқ).

# Деректерге қол жеткізу әдістері

## Бинарлық ағаштар

- *В ағашы үшін:*
  - ✓ салыстырмалы түрде қарапайым, дәйекті қол жетімділікті ұйымдастыруға болады, өйткені барлық жапырақтары бірдей деңгейде орналасқан;
  - ✓ кілттерді қосу және өзгерту кезінде барлық өзгерістер бір түйінмен шектеледі.
- Кеңістіктік (көп өлшемді) деректерге қол жеткізу үшін R-ағаштар (R-Tree) қолданылады.
- *R-ағаш* - бұл А. Гуттман ұсынған кеңістіктік деректерге қол жеткізудің индекс құрылымы. R-ағаш мезгіл-мезгіл қайта индекстеусіз деректерді қосу, жою және іздеу әрекеттерін ерікті түрде орындауға мүмкіндік береді.

# Деректерге қол жеткізу әдістері

## Хештеу

- жазу мекенжайын анықтау үшін пайдаланылатын негізгі деректер *хэш-кестесі* деп аталатын кесте түрінде ұйымдастырылады.
- Хэш функциясы жазу кілтінің немесе оның кодының мәнін анықтайтын сандар немесе биттер тізбегін хэш етеді, нәтижесінде жазбалар орналастырылып, ізделетін хэш мекен-жайы пайда болады.
- Кілтті жинақтау әдісі бойынша түрлендіруді жүзеге асыратын жиі қолданылатын хэш функциясы кілтті бірнеше бөлікке бөледі, содан кейін олар қажетті диапазонда Сан пайда болатындай етіп жинақталады.

# Деректерге қол жеткізу әдістері

## Хештеу мысалы

- Сегіз таңбалы  $K=97434658$  және  $K=31269857$  кілттері болсын. Оларды үш таңбалы мекенжайларға түрлендіру керек.
- Сіз келесі операцияларды жасай аласыз:
- $h(97434658) = 974 + 346 + 58 = 378$
- $h(31269857) = 312 + 698 + 57 = 067$
- $h$  белгісі кілтті өңдеуді хэш-функциясы жүзеге асыратынын білдіреді.

# Деректерге қол жеткізу әдістері

## Хеш-функция

- жазу кілтін оның мекен-жайына бір мәнді түрлендіруді қамтамасыз етуі керек
- мекенжайлар деректерді сақтау үшін бөлінген жад аймағына біркелкі бөлінуі керек.
- бұл өте күрделі болмауы керек, өйткені түрлендіруге қажет уақыт жүргізу, іздеу немесе өңдеу операцияларының орындалу уақытына қосылады.
- бірдей мекенжайларды алу мүмкіндігін толығымен жоймайды. *Қақтығыстар (коллизия)* бар, яғни әртүрлі жазбалар бірдей мекен-жай алатын жағдайлар.

# Деректерге қол жеткізу әдістері

## Қақтығыстарды шешу әдістері:

- Хэш-мекен-жайы қайта рехештеу болады, яғни белгілі бір мекен-жай түрлендірулері. Мысалы, оны тұрақтыға көбейтуге болады, нәтижесінде жаңа мекен-жай пайда болады.
- Мұндай түрлендірулер бос жад ұяшығының адресі пайда болғанша орындалуы мүмкін. Жазбаны іздеген кезде түрлендірулер тізбегі бірдей болады.
- Тағы бір әдіс-жасалған мекенжайды дәйекті қарау үшін бастапқы нүкте ретінде пайдалану. Осы мекен-жайдан жадта бос орынды іздеу басталады.

# Деректерге қол жеткізу әдістері

## Қақтығыстарды шешу әдістері:

- Функциямен жасалған мекен-жай белгілі бір жазбаны емес, жад парағын сақтау мекен-жайы болып саналады, оның ішінде осы мекенжайды алған барлық жазбалар орналастырылған. Беттің шегінде жазбалар түсу ретімен немесе қандай да бір белгіленген тәртіпке сәйкес орналастырылуы мүмкін.
- Егер уақыт өте келе парақ толтырылса, жадта алдыңғы көрсеткішпен байланысқан жаңа бет бөлінеді.

# Индекстер

- *Индекс* - бұл ДҚБЖ-ге файлдағы жеке жазбаларды тезірек анықтауға және пайдаланушылардың сұраныстарының орындалу уақытын қысқартуға көмектесетін мәліметтер құрылымы.
- Индекстердің мақсаты-қарым-қатынас кортежіне (жазбаға) кілт бойынша тікелей қол жеткізуді қамтамасыз ету. Әдетте индекс тек бір қатынас үшін анықталады.
- *Индекс* - бұл деректер кестесіндегі әр жазбаның негізгі мәндерін қамтитын және пайдаланушы талап ететін ретпен жазылған кесте.
- Кілтті мәндер кестенің бір немесе бірнеше өрістеріне негізделген. Сонымен қатар, индексте кестедегі тиісті жазбаларға ерекше сілтемелер бар.

# Индекс түрлері

- *Бастапқы индекс.* Деректер файлы тапсырыс кілтінің өрісіне дәйекті түрде реттеледі, ал тапсырыс кілтінің өрісіне сүйене отырып, әр жазбада ерекше мәнге ие болуға кепілдік беретін индекстеу өрісі жасалады.
- *Екіншілік индекс.* Деректер файлы кілтсіз өріске сәйкес келеді және осы кілтсіз өрістің негізінде индекстеу өрісі қалыптасады, сондықтан файлда осы индекстеу өрісінің мәніне сәйкес келетін бірнеше жазбалар болуы мүмкін.
- Егер индекстеу файлында индекстелген өрістің әр мәніне сәйкес келетін бір ғана жазба болса, онда мұндай индекс *бірегей* деп аталады.

# Индекс түрлері

- Индекстер арқылы жазбаға жүгіну екі кезеңде жүзеге асырылады:
  1. индекс құрылымында талап етілетін атрибут мәні және тиісті жазба мекенжайы бар;
  2. бұл мекен-жайда сыртқы сақтау құрылғысына (ССҚ) сілтеме жасалады. Индекс тұтастай ЖЖ-қа жүктеледі (немесе ДБ-мен жұмыс істеу кезінде онда тұрақты сақталады).
- Әр мәліметтер базасы үшін бір уақытта бірнеше бастапқы және екінші индекстерді қолдауға болады.
- **Жеке** индекстер мен **құрама** индекстер бар. Құрама индекс бір кестенің екі немесе одан да көп бағандарын қамтиды.

# Индекс түрлері

- Индексстерге динамикалық қолдау көрсетіледі, яғни дерекқорды жаңартқаннан кейін – жазбаларды қосу немесе жою, сондай-ақ кілт индексіне кіретін жазу өрістерін өзгерту кезінде мәліметтер базасының жаңартылған нұсқасына сәйкес келеді.
- Индексті жаңарту біраз уақытты алады, сондықтан көптеген индексстердің болуы дерекқорды баяулатуы мүмкін.
- Логикалық жазбалары бар файл *деректер файлы* деп аталады, ал индексстік жазбалары бар файл *индекс файлы* деп аталады.

# Бақылау сұрақтар:

1. Деректер құрылымы ұғымын анықтаңыз және олардың жіктелуін беріңіз
2. Әр түрлі сызықтық деректер құрылымдарының салыстырмалы сипаттамасын жасаңыз
3. Әр түрлі сызықтық емес деректер құрылымдарының салыстырмалы сипаттамасын жасаңыз
4. Деректерді ұсыну деңгейлерін және олардың әрқайсысының міндеттерін сипаттаңыз
5. Кілт деректеріне қол жеткізуді жүзеге асыратын әдістердің сипаттамасын беріңіз
6. Индекс ұғымын, индекс түрлерін анықтаңыз

# Әдебиет тізімі:

1. Т. Конноли. Базы данных. Проектирование, реализация и сопровождение. Теория и практика.: Пер.с англ. – М.: Изд.дом «Вильямс», 2012. – 1440 с.
2. К.Дейт. Введение в системы БД. изд.6-е. :Пер.с англ. - М.: Изд.дом "Вильямс", 2014. – 1200с.
3. Д.Ульман. Введение в системы баз данных. – М.: Издательство «Лори», 2015. – 853с.



Назарларыңызға рахмет!