УДК 519.8

М.В.КОПЕЙКИН, *канд. техн. наук, доцент, ord@yandex.ru*
В.В.СПИРИДОНОВ, *канд. техн. наук, доцент, vt1rp2@yahoo.com*
Е.О.ШУМОВА, *ст. преподаватель, shumovaeo@yandex.ru*
Национальный минерально-сырьевой университет «Горный», Санкт-Петербург

M.V.KOPEIKIN, *PhD in eng. sc., associate professor, ord@yandex.ru*
V.V.SPIRIDONOV, *PhD in eng. sc., associate professor, vt1rp2@yahoo.com*
E.O.SHUMOVA, *senior lecturer, shumovaeo@yandex.ru*
National Mineral Resources University (Mining University), Saint Petersburg

ВЫБОР СПОСОБА ХРАНЕНИЯ ОТНОШЕНИЙ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

В минерально-сырьевом комплексе проблема хранения и обработки данных имеет определенную специфику, связанную, как правило, со значительным объемом имеющихся данных. Это обусловливает заметный интерес к повышению эффективности поисковых операций, зависящей от организации хранения данных.

В любой реляционной базе данных предполагается, что пользователь воспринимает базу данных как набор нормализованных отношений (таблиц). На физическом уровне отношения (таблицы) могут быть представлены различными способами.

Статья посвящена анализу кортежного (традиционного) и доменного (транспонированного) способа хранения отношений.

***Ключевые слова*:** кортеж, домен, структура данных

THE CHOICE OF DATA RELATIONS STORING IN INFORMATIONAL SYSTEMS

Storing and processing data of mineral resources usually imply large amounts of data. So increasing data search operation speed is the subject of a notable interest.

User of any relational data base, as a rule, imagines it as a set of normalized relations (tables). These tables, at the physical level, can be represented in the different ways.

The paper is dedicated to the analysis and comparison of cortege (traditional) and domain (transposed) ways of storing data relations.

***Key words*:** cortege, domain, data structure.

Одним из наиболее важных факторов функционирования информационных систем в минерально-сырьевом комплексе является грамотно сформированная структура данных, подлежащих обработке в системе. В основном используются доменные или кортежные способы хранения.

Для сравнения кортежного и доменного способа хранения отношений базы данных (БД) на диске рассмотрим характеристики запросов к БД. Пусть в системе хранится только одно отношение R с числом кортежей m и числом полей в каждом кортеже n. Отношение R упорядочено по значению первичного ключа (номер служащего), определенного на одном атрибуте.

Пусть отношение R задано кортежами {С1, О1, П1}, {С2, О2, П2}, {С3, О3, П3}, {С4, О4, П4}, где С1, С2, С3, С4 – номера служащих; О1, О2, О3, О4 – номера отделов; П1, П2, П3, П4 – номера проектов.

Разобьем все запросы к R на три класса: 1-й класс – считывание в оперативную память (ОП) всего файла; 2-й класс – опреде-

ление значений k полей записи, не принадлежащих ключу отношения R; 3-й класс – считывание f выборочных записей по ключу отношения R. Тогда количество запросов первого и второго классов соответственно

$$\lambda^1 = 1;\ \lambda^2 = \sum_{j=2}^{n}\left[c_n^j\left(\sum_{i=1}^{m}c_m^i\right)^{j-1}\right]. \qquad (1)$$

Запросы класса λ^2 учитывают все сочетания значений элементов доменов, входящих в отношение R. Поисковым образом служит $(n-1)$-е поле отношения R со всевозможным сочетанием элементов, входящих в $(n-1)$-й домен по одному из предикатов И, ИЛИ, НЕ или их сочетанию. Формула (1) определена для предиката И.

Упростив (1) с использованием бинома Ньютона, получим

$$\lambda^2 = \frac{(2^m)^n - n2^m + n - 1}{2^m - 1}.$$

В классе 2 для отношения R, упорядоченного по ключу «Номер служащего», примером может служить запрос «Найти номера служащих, работающих в отделе О1 над проектами П1 и П2»:

SELECT № служ FROM R

WHERE № отдела = О1 AND
(№ проекта = П1 AND № проекта = П2).

Для ответа на запрос типа λ^2 файл необходимо считывать поочередно по записям, так как анализу подлежат все m записей. Математическое ожидание числа обрабатываемых полей (числа доменов) в одном запросе

$$\mathrm{MO}_{\lambda^2} = n/2. \qquad (2)$$

Число запросов третьего класса

$$\lambda^3 = \left[\sum_{j=1}^{n-1}c_{n-1}^j \cdot \sum_{i=1}^{m-1}c_m^i\right]$$

или после упрощения

$$\lambda^3 = [(2^{n-1}-1)(2^m-2)] = 2^{m+n-1} - 2^m - 2^n + 2.$$

При этом математическое ожидание числа обращений к жесткому диску $\mathrm{MO}_{\lambda^3\ \text{ж.д}} = m/2 = f$, а математическое ожидание количества полей исследуемых в одном запросе,

$$\mathrm{MO}_{\lambda^3} = \frac{n}{2} + 1.$$

Примером запроса, относящегося к классу 3, для отношения R может служить запрос «Найти номера отделов, в которых работает служащий С1 или (и) С2»:

SELECT DISTINCT № отдела FROM R
WHERE (№ служ = С1 OR № служ = С2).

Нетрудно заметить, что число запросов класса λ^2 превосходит число запросов λ^3 даже по одному предикату И на величину

$$\frac{2^{mn} - n2^m + n - 1}{(2^m-1)(2^{m+n-1} - 2^m - 2^n + 2)} \cong 2^{mn-2m-n+1},$$

что позволяет сделать вывод, что для R запросы типа λ^2 являются определяющими.

Определим среднее время ответа на запрос каждого класса с учетом кортежного и доменного способов хранения отношения R.

Кортежная организация. В связи с оценочной характеристикой среднего времени ответа на запросы, время, связанное с особенностями организаций отношений на диске (прямая, индексная и т.д.) и программами методов доступа, не учитывается. При ответе на запрос класса λ^1 временные затраты

$$t_{\lambda^1}^K = \sum_{i=1}^{m}t_i + \frac{m\,\mathrm{size(rec)}}{t_{\text{пер}}} \qquad (3)$$

где t_i - время поиска m_i-й записи на диске; $t_{\text{пер}}$ – время передачи данных между диском и оперативной памятью; size(rec) – длина в байтах записи (кортежа) отношения R, $\mathrm{size(rec)} = ln$; l – длина i-го поля кортежа; m – количество кортежей в R.

Пусть необходимо определить значение k полей кортежа, не принадлежащих ключу отношения R (запрос класса λ^2). При этом

$$t_{\lambda^2}^k = \sum_{i=1}^{m}t_i + \frac{m\,\mathrm{size(rec)}}{t_{\text{пер}}} + k\frac{5m+3}{V}, \qquad (4)$$

где V – среднее быстродействие ЭВМ; $5m+3$ – число поисковых операций [2].

Для ответа на вопрос класса λ^2 необходимо найти все записи файла (первая составляющая формулы), передать их в оперативную память для обработки (вторая составляющая) и найти в неупорядоченных векторах значения элементов домена, удовлетворяющих поисковому запросу. Последнее слагаемое определяет время для поиска элемента в неупорядоченном векторе.

Пусть необходимо проанализировать f выборочных записей (кортежей) отношения R (класс запросов λ^3). Тогда

$$t^K_{\lambda^3} = \sum_{i=1}^{f} t_i + \frac{f\text{size(rec)}}{t_{\text{пер}}} + f\frac{18\log_2 f + 13}{V}. \quad (5)$$

Последняя составляющая формулы (5) учитывает время поиска в упорядоченном векторе [2].

Доменная организация. Рассмотрим способ хранения отношения R по доменам, когда число записей в файле R^{-1} равно n, а количество полей в записи m. Пусть необходимо считать весь файл в ОП. Тогда

$$t^D_{\lambda^1} = \sum_{i=1}^{n} t_i + \frac{n\text{size(dom)}}{t_{\text{пер}}}, \quad (6)$$

где size(dom) – длина в байтах домена отношения R, $\text{size(dom)} = lm$.

Если необходимо определить значения k полей кортежа отношения R (класс 2), то

$$t^D_{\lambda^2} = \sum_{i=1}^{k} t_i + \frac{k\text{size(dom)}}{t_{\text{пер}}} + k\frac{5m+3}{V}. \quad (7)$$

Заметим, что в отличие от подобной операции при кортежной организации в данном случае передаются в ОП и обрабатываются не все n полей, а только те k из них, которые непосредственно входят в запрос.

Определим среднее время ответа на запросы 3-го класса при доменной организации отношения R:

$$t^D_{\lambda^3} = \sum_{i=1}^{n} t_i + \frac{n\text{size(dom)}}{t_{\text{пер}}} + n\frac{5m+3}{V}. \quad (8)$$

Далее попарно сравним времена из (3)-(8). Нетрудно видеть что $t^k_{\lambda^1} = t^D_{\lambda^1}$. Учитывая (2), (4) и (7) запишем

$$\Delta_{\lambda^2} = t^D_{\lambda^2} - t^K_{\lambda^2} = \sum_{i=1}^{\frac{n}{2}} t_i + \frac{\frac{n}{2}\text{size(dom)}}{t_{\text{пер}}} + \frac{n}{2}\left(\frac{5m+3}{V}\right) -$$

$$- \sum_{i=1}^{m} t_i - \frac{m\text{size(rec)}}{t_{\text{пер}}} - \frac{n}{2}\left(\frac{5m+3}{V}\right). \quad (9)$$

Для реальных файлов $m >> n$, поэтому

$$\sum_{i=1}^{\frac{n}{2}} t_i < \sum_{i=1}^{m} t_i \text{ и } \frac{\frac{n}{2}lm}{t_{\text{пер}}} < \frac{mln}{t_{\text{пер}}}; \quad (10)$$

из (9) получим

$$\Delta_{\lambda^2} = \sum_{i=1}^{\frac{n}{2}} t_i - \sum_{i=1}^{m} t_i < 0. \quad (11)$$

Следовательно, доменная организация предпочтительнее кортежной.

Аналогичным образом оценим разность $\Delta_{\lambda^3} = t^D_{\lambda^3} - t^K_{\lambda^3}$. При этом, учитывая соотношения (5) и (8), найдем

$$\Delta_{\lambda^3} = \sum_{i=1}^{\frac{n}{2}+1} t_i + \frac{\left(\frac{n}{2}+1\right)\text{size(dom)}}{t_{\text{пер}}} + \left(\frac{n}{2}+1\right)\frac{5m+3}{V} -$$

$$- \sum_{i=1}^{\frac{m}{2}} t_i - \frac{\frac{m}{2}\text{size(rec)}}{t_{\text{пер}}} - \frac{m}{2}\left(\frac{18\log_2\frac{m}{2}+13}{V}\right).$$

Знак величины Δ_{λ^3} определяется значениями

$$\sum_{i=1}^{\frac{n}{2}+1} t_i \text{ и } \sum_{i=1}^{\frac{m}{2}} t_i.$$

Так как вторые составляющие в (5) и (8) равноценны, а третьи составляющие на порядок меньше, чем первые, то приближенно можно записать

$$\Delta_{\lambda^3} = \sum_{i=1}^{\frac{n}{2}+1} t_i - \sum_{i=1}^{\frac{m}{2}} t_i.$$

Тогда

$$\Delta = \Delta_{\lambda^2} - \Delta_{\lambda^3} = \sum_{i=1}^{\frac{n}{2}} t_i - \sum_{i=1}^{m} t_i - \sum_{i=1}^{\frac{n}{2}+1} t_i + \sum_{i=1}^{\frac{m}{2}} t_i ,$$

учитывая, что $m >> n$ запишем

$$\Delta = -\sum_{i=1}^{\frac{m}{2}} t_i, \text{ откуда } \Delta < 0 .$$

Отрицательный знак Δ указывает, что среднее время ответа на запрос при доменной организации меньше, чем при кортежной организации отношения R.

Учитывая изложенное, можно сделать вывод, что доменная организация при условии, что $m >> n$, более эффективна. Однако при малых значениях f и использовании методов вторичного индексирования вопрос о предпочтительности доменной организации остается открытым. Полученные уравнения для времени поиска записей при доменной и кортежной организации файла сведем в таблицу, учитывая, что формулы, записанные для $A_{\text{I},1}$, $A_{\text{I},2}$ и $A_{\text{I},3}$, эквивалентны формулам в ячейке $A_{\text{II},1}$, $A_{\text{II},2}$, и $A_{\text{II},3}$ соответственно, т.е. выполняются эквивалентные операции, и рассмотрим влияние способа доступа и организации файла на временные характеристики. Здесь индексы I и II – соответствуют кортежной и доменной организациям, индексы 1, 2 и 3 – последовательной, индексной и прямой организациям.

Заметим, что время t_i для каждой из формул зависит от способа доступа и способа организации файла. Получим время ответа на запросы (для однозначности считаем, что задан индексно-последовательный файл) с числом записей m; для удобства примем, что поля записи (всего полей n) имеют равную длину (в байтах).

Рассмотрим три типа запросов к файлу, причем каждый тип запроса может быть оценен соответствующей формулой из таблицы. Так как индексно-последовательный файл позволяет использовать как последовательный доступ, так и прямой, то для выполнения соответствующего типа запроса используется тот способ доступа, который наиболее отвечает природе запроса.

Действие всех запросов ограничено на некотором интервале времени. Определим суммарное время T для ответа на запрос для файла с кортежной организацией

$$T = \lambda^1 A_{1,1} + \lambda^2 A_{1,2} + \lambda^3 A_{1,3},$$

где λ^i – количество запросов i-го типа.

Определим время ответа на запросы для каждого типа запросов. При последовательном способе доступа [3]

$$t_{\text{посл}} = 1{,}25(N_{\text{ц}} t_{\text{п}} + m t_{\text{об}}),$$

где $N_{\text{ц}}$ – число цилиндров, занимаемых файлом, $N_{\text{ц}} = N_{\text{д}} / d$; d – количество дорожек в цилиндре; $N_{\text{д}}$ – число дорожек, занимаемых файлом,

$$N_{\text{д}} = \frac{m}{1 + \dfrac{l_{\text{д.т}} - l_{\text{з}}}{l_0 + k l_{\text{з}}}};$$

$l_{\text{д.т}}$ – длина трека; $l_{\text{з}}$ – длина логической записи (блока); l_0 – длина служебной информации; k – коэффициент, учитывающий расположение записей на дорожке; $t_{\text{п}}$ – время перемещения механизма доступа на один цилиндр; m – число записей (блоков) в файле; $t_{\text{об}}$ – время полного оборота диска.

Время поиска записей при доменном и кортежном способах хранения файла

Способ хранения	Организация файла		
	последовательная	индексная	прямая
Кортежный	$t = \sum_{i=1}^{m} t_i + \dfrac{m\text{card}(m)}{t_{\text{пер}}}$	$t = \sum_{i=1}^{m} t_i + \dfrac{m\text{card}(m)}{t_{\text{пер}}} + \sum_{j=1}^{k} \dfrac{5\text{card}(n_j)+3}{V}$	$t = \sum_{i=1}^{F} t_i + \dfrac{F\text{card}(m)}{t_{\text{пер}}} + \sum_{j=1}^{F} \dfrac{18\log_2 \text{card}(m_j)+13}{V}$
Доменный	$t = \sum_{i=1}^{m_1} t_i + \dfrac{n\text{card}(\alpha)}{t_{\text{пер}}}$	$t = \sum_{i=1}^{k_1} t_i + \dfrac{k\text{card}(\alpha)}{t_{\text{пер}}} + k\dfrac{5m+3}{V}$	$t = \sum_{i=1}^{m_1} t_i + \dfrac{n\text{card}(\alpha)}{t_{\text{пер}}} + n\dfrac{5m+3}{V}$

Здесь выражение $1+\frac{l_{д.т}-l_з}{l_0+kl_з}$ – учитывает количество записей (блоков), помещающейся на одной дорожке; сумма $N_ц t_п + m t_{об}$ – время для считываний последовательного набора данных; коэффициент 1,25 – последовательную обработку индексного файла. Данная формула учитывает также и время передачи между оперативной памятью и магнитным диском (МД).

Время считывания одной записи при произвольной обработке индексно-последовательного файла складывается из времени двух перемещений считывающего механизма к индексу цилиндров и к первичному цилиндру, и времени на три считывания: индекса цилиндра, индекса дорожки и требуемой записи данных.

Время на считывание записи данных равно полному обороту диска [1], а время на поиск и чтение индекса дорожки цилиндра – половине оборота, т.е. должно быть учтено время на передачу одной записи между МД и ОП, поэтому

$$T_{пр} = 2(t_п^c + t_{об})m_i, \qquad (23)$$

где $t_п^c$ – среднее время перемещения механизма доступа; $t_{об}$ – время полного оборота диска; m_i – количество считываемых записей.

Тогда время для файла с кортежной организацией

$$T^I = \lambda^1 \sum_{i=1}^{m}\left[1{,}25\left(N_ц t_п + m_i t_{об}\right)\right] +$$

$$+ \sum_{h=1}^{\lambda^3}\lambda_h^2\left(\sum_{i=1}^{m}\left[1{,}25\left(N_ц t_п + m_i t_{об}\right)\right] +\right.$$

$$\left. + (n-1)2t_{исп}m + (n-1)\left(\frac{5\mathrm{card}(n)+3}{V}\right)\right) +$$

$$+ \sum_{h=1}^{\lambda^3}\lambda_h^3\left(\sum_{i=1}^{\frac{m}{2}} 2(t_п^c + t_{об})m_i + \frac{m_i}{2}t_{исп}\left(\frac{n}{2}+1\right)\right).$$

При этом учитывается тот факт, что в ОП элементы массива запоминаются поочередно один за другим.

Тип запроса λ^2, кортежная организация файла. Массив R задан кортежами С1, О1, П1; С2, О2, П2; С3, О3, П3; С4, О4, П4. Тогда время поиска местоположения поля записи

$$t_з = t_з^1 + t_з^2,$$

где $t_з^1$ – поиск местоположения записи, $t_з^1 = t_{исп}^i m^i$; $t_з^2$ – поиск местоположения поля записи, $t_з^2 = t_{исп}^i m$; $t_{исп}^i$ – время для формирования адреса записи m_i.

Адрес записи m_i

$$A_i = A_1 + (i-1)l_з^1$$

где A_1 – базовый адрес массива, равный адресу 1-й записи; $l_з^1$ – длина записи, т.е. количество ячеек ОП для хранения записи $l_з$.

Время $t_{исп}$ зависит от характеристик ЭВМ: от времени цикла обращения к ОП, времени выборки (разрядности процессора) и времени выполнения операций, формирующих исполнительный адрес.

Тогда общее время $t_з = 2t_{исп}m$.

Время для ответа на запросы типа λ^2

$$\sum_{h=1}^{n}\lambda_h^2\left[\sum_{i=1}^{m}\left[1{,}25\left(\mathrm{N}_ц t_п + m_i t_{об}\right)\right] + (n-1)2t_{исп}m + \right.$$

$$\left. + (n-1)\left(\frac{5\mathrm{card}(n)+3}{V}\right)\right].$$

Тип запроса λ^3, кортежная организация файла. Для ответа на запросы типа λ^3 необходимо считывание в среднем $m/2$ записей, среднее число полей, подлежащих анализу $n/2+1$. После считывания прямым методом доступа m_i, записи (всего их $m/2$) производится анализ $(n/2+1)$ полей, т.е. время для поиска $(n/2+1)$ полей $t_з = t_{исп}(n/2+1)$. Тогда время для ответа на запросы типа λ^3

$$\sum_{h=1}^{\lambda^3}\lambda_h^3\left[\sum_{i=1}^{\frac{m}{2}} 2\left(t_п^c + t_{об}\right)m_i + \frac{m_i}{2}t_{исп}\left(\frac{n}{2}+1\right)\right];$$

$$\lambda^3 = \sum_{i=1}^{n-1} C_{n-1}^{i} = 2^{n-1} - 1.$$

Время для ответа на запросы для файла с доменной организацией

$$T^{\text{II}} = \lambda^1 \sum_{i=1}^{n} \left[1{,}25\left(N_{\text{ц}} t_{\text{п}} + n_i t_{\text{об}}\right)\right] +$$

$$+ \sum_{h=1}^{\lambda^2} \lambda_h^2 \sum_{i=1}^{n-1} \left[1{,}25\left(N_{\text{ц}} t_{\text{п}} + n_i t_{\text{об}}\right)\right] +$$

$$+ n_i t_{\text{исп}} + n_i \left(\frac{5\text{card}(n) + 3}{V}\right) +$$

$$+ \sum_{h=1}^{\lambda^3} \lambda_h^3 \left[2\gamma\left(t_{\text{п}}^{\text{c}} + t_{\text{об}}\right)\right] + \frac{m}{2} \frac{18\log_2 m + 13}{V} +$$

$$+ \sum_{i=1}^{\frac{n}{2}} \left[2\gamma(t_{\text{п}}^{\text{c}} + t_{\text{об}}) n_i + t_{\text{исп}} \frac{mn}{4}\right],$$

где γ – коэффициент, учитывающий переходящие записи.

Заметим, что, используя индексную организацию для файла с доменным способом хранения, необходимо применять средства, позволяющие использовать кластерные файлы или, например, прицепы, или другие способы списковой организации данных. Сделанное замечание не влияет на расчетные формулы.

Тип запроса λ^2, доменная организация файла. В ОП массив R располагается в форме С1, С2, С3, С4; О1, О2, О3, О4; П1, П2, П3, П4. В этом случае $t_{\text{з}} = t^{i}_{\text{исп}}$, а время поиска $(n-1)$ полей записи отношения R получим в виде $t_{\text{з}} = (m-1)t_{\text{исп}}$. Время для ответа на запросы λ^2 при доменной организации

$$T^{\text{II}} = \sum_{h=1}^{\lambda^2} \lambda_h^2 \left[\sum_{i=1}^{n\lambda_h^2} \left(\left[1{,}25\left(N_{\text{ц}} t_{\text{п}} + n_i t_{\text{об}}\right)\right] + \right.\right.$$

$$\left.\left. + n_i t_{\text{исп}} + n_i \frac{5\text{card}(n) + 3}{V} \right) \right],$$

где $n\lambda_h^2$ учитывает число необходимых доменов при выполнении запросов λ^2.

Так как среднее число $\tilde{n}\lambda_h^2 = n - 1$, то для подсчета среднего времени $T_{\lambda^2}^{\text{II}}$ принимаем, что файл считывается последовательно. Необходимо заметить, что файл может обслуживаться и прямым методом доступа при условии $T_{\lambda^2}^{\text{II}} = T_{\lambda^2}^{\text{I}}$.

Тип запроса λ^3, доменная организация файла. Запрос типа λ^3 требует считывания $(n/2 + 1)$ записей (полей) R, причем будем считать, что одна запись содержит ключи отношения R и упорядочена. При ответе на запросы λ^3 требуется в среднем исследовать $m/2$ полей. Для ответа на запрос типа λ^3 необходимо считать в ОП запись n_i, содержащую значение ключей R. В считанной записи необходимо определить местоположение поисковых полей $(m/2)$, т.е. произвести поиск в упорядоченной таблице. Время для поиска одного поля $(18\log_2 m + 13)/V$, а для поиска $m/2$ полей $(m/2)(18\log_2 m + 13)/V$.

Затем в оставшихся $(n/2 + 1 - 1)$ записях необходимо определить местоположение $m/2$ полей, считывая записи прямым методом доступа, т.е.

$$t_{\text{исп}} \frac{m}{2}\left(\frac{n}{2} + 1 - 1\right) = \left(t_{\text{исп}} \frac{m}{2}\right)\frac{n}{2}.$$

Тогда время для ответа на запросы λ^3

$$T^{\text{II}} = \sum_{h=1}^{\lambda^3} \lambda_h^3 \left[2\left(t_{\text{п}}^{\text{c}} + t_{\text{об}}\right) + \frac{m}{2} \frac{18\log_2 m + 13}{V} + \right.$$

$$\left. + \sum_{i=1}^{n\lambda_h^3} (2t_{\text{п}}^{\text{c}} + t_{\text{об}}) n_i + t_{\text{исп}} \frac{m}{2}\frac{n}{2} \right].$$

Величина $n\lambda_h^3$ учитывает число обращений при выполнении запросов типа λ^3:

$$n\lambda_h^3 = \frac{n}{2} + 1 - 1 = \frac{n}{2}.$$

Составляющие в формулах $(18\log_2 x + 13)/V$ и $(5\,\text{card}\,(x) + 3)/V$ определяют мак-

симальное время для соответствующих операций. В реальности составляющие могут быть уменьшены как за счет применения наиболее эффективных алгоритмов [1, 2], так и из-за того, что для этих формул в среднем $x = x/2$.

ЛИТЕРАТУРА

1. Алгоритмы: Построение и анализ: Пер. с англ. / Томас Х.Кормен, Чарльз И.Лейзерсон, Рональд Л.Ривест, Клиффорд Штайн. М., 2007.

2. Кнут Д. Искусство программирования для ЭВМ. Сортировка и поиск. М., 2000. Т.3.

3. *Конноли Т.* Базы данных: проектирование, реализация и сопровождение. Теория и практика / Т.Конноли, К.Бегг, А.Страчан. М., 2000.

REFERENCES

1. *Cormen Thomas H.*, *Leiserson Charles E.*, *Rivest Donald L.*, *Stein Clifford*: Introduction to Algorithms, Moscow, 2007.

2. *Knuth Donald E.* The Art of Computer Programming. Sorting and Searching. Moscow, 2000. Vol.3.

3. *Connolly T.*, *Begg C.*, *Strachan A.* Database systems: a practical approach to design, implementation and management. Moscow, 2000.