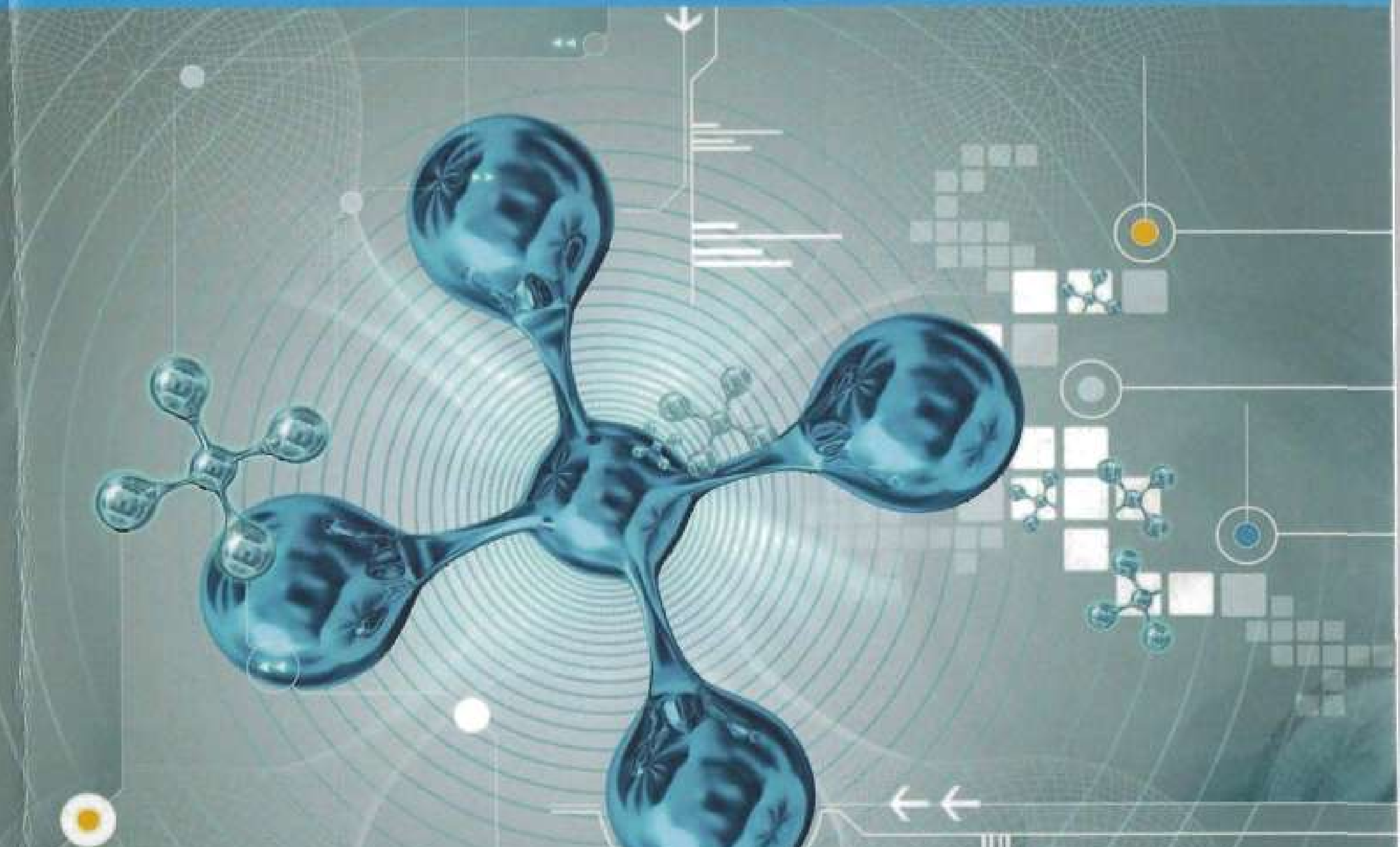
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ЛОДЗЬКИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ (ПОЛЬЩА)

Матеріали III Міжнародної науково-практичної конференції молодих вчених

ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ГАЛУЗІ ПРИРОДНИЧИХ НАУК

6-8 червня 2018 року, Одеса, Україна



Одеса
ТЕС
2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ЛОДЗЬКИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ (ПОЛЬЩА)

Матеріали III Міжнародної науково-практичної конференції молодих вчених

ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ГАЛУЗІ ПРИРОДНИЧИХ НАУК

6-8 червня 2018 року, Одеса, Україна

Одеса
ТЕС
2018

УДК 681.3:502
М 84

Матеріали III Міжнародної науково-практичної конференції молодих вчених:/ «Теоретичні та прикладні аспекти застосування інформаційних технологій в галузі природничих наук»; ОДЕКУ. Одеса: ТЕС, 2018. – 190 с. ISBN 978-617-7337-77-4

У збірнику представлені матеріали науково-практичної конференції молодих вчених, які висвітлюють актуальні проблеми застосування інформаційних технологій в галузі природничих наук, а також засоби їх вирішення.

Матеріали конференції надруковано в авторській редакції. Автори матеріалів несуть відповідальність за достовірність наведених відомостей, точність даних за цитованою літературою і за використання даних, що не підлягають відкритій публікації.

ISBN 978-617-7337-77-4

DEVELOPMENT OF THE PROGRAM FOR AUTOMATED SCHEDULING OF LESSONS USING RELATIONAL ALGEBRA METHODS AND SQL LANGUAGE TOOLS

Petukhin D.,teacher of special disciplines, Velykodniy S. – PhD, Computer Science, Associate Professor

The Communication and Information College of the Odessa National Academy of Communications named after A.S. Popov

Programs for automated scheduling of classes in higher education institutions are being developed for several years and are widely represented on the software market. Commercial programs for creating timetables have a private code. There is no common, generally accepted algorithm for solving the problem of scheduling classes at a higher educational institution; nevertheless, almost all authors admit the need for using DBMS to create and store a database.

This need arises from the large amount of information that is used as input data for this task. However, DBMS in this instance can be used not only as a "data warehouse", the task of scheduling the classes is completely described in terms of relational theory and relational algebra and can be solved by means of relational databases – the SQL language using stored procedures.

Consider the formulation of the problem in terms of relational theory and relational algebra. Let's call "lesson" one training pair in one academic group for one academic discipline. Lessons can vary in types: lectures, seminars (practical classes), laboratory work (classes). Students of higher educational institutions are trained according to educational and professional programs for the specialties. Each year, according to these programs, curricula are developed for each course of each specialty. Each academic group is attached to a curriculum.

Let the relation G describe academic groups; the relation H describes the possible types of lessons; relation D describes the academic disciplines that are studied in a higher education institution; the L relation describes the set of university curricula. The basic types of entities listed are table-directories that contain the identifier of the corresponding relation, attribute-name (group, type of lesson, academic discipline, curriculum), as well as other attributes that characterize the corresponding type of entity.

The training load according to curricula is obtained by connecting the relation D with the relation H through the associative type of the entity LP, which describes the filling of the curriculum with the study disciplines (1):

$$V \leftarrow (L \bowtie (LP \bowtie D)) \bowtie H, \qquad (1)$$

where "$\leftarrow$" is the relational assignment operation;

"$\bowtie$" is the operation of the natural connection of the relations on a common set of key attributes.

The curriculum indicates the load per semester for this discipline for every type of lesson. So relation LP will contain the following set of attributes:

ID_p – is the identifier of the curriculum:

ID_d – the identifier of the academic discipline:

ID_h – lesson type identifier:

k – the number of hours of this type of lessons per discipline per week.

The base type of the entity L contains the attribute "weeks" – the number of study weeks in the semester, with the help of which the total number of hours of different types of classes in the discipline per semester is obtained. The integer attribute k, generally, can take values from 1 to 6, that is, one discipline can have several pairs of one type of classes per week. Let's consider only the classes that take place every week.

Let the associative type of the GP entity describe the attachment of the academic group to the curriculum. It has two attributes: ID_g – group identifier and ID_p – curriculum identifier. The associative entity type L_P describes the classroom load for different types of lessons for each discipline of the curriculum in standard training pairs; unlike the LP relation, instead of the attribute k, the L_P relation has the attribute num – the number of the training pair of this type of lesson for this discipline per week. Then the classes of all academic groups can be described by the associative type of the entity W:

$$W(ID\_s, ID\_g, ID\_p, ID\_d, ID\_h, num) \leftarrow GP \bowtie L\_P, \qquad (2)$$

where ID_s is the lessons identifier.

Let the associative type of the entity TL (ID_t, ID_s) (TeacherLoad) describe the assignment of the classes to the teachers. Then the lessons that need to be put on the schedule are described by the associative type of the entity S:

$$S\ (ID\_s, ID\_g, ID\_t) \leftarrow \pi_{ID\_s,\, ID\_g,\, ID\_t}(W \bowtie TL), \qquad (3)$$

By applying the operation of the Cartesian product to the relations S, P, A, we obtain the relation Z, which describes all possible variants of the schedule of classes (4):

$$Z\ (ID\_s, ID\_g, ID\_t, ID\_p, D\_a) \leftarrow \pi_{ID\_s, ID\_g, ID\_t,\, ID\_p, ID\_a}(S \times A \times P), \qquad (4)$$

where P, A are the basic relations describing training pairs and audiences, respectively;

ID_p is the identifier of the training pair;

ID_a is the identifier of the audience.

There is a large number of classes that can be conducted only in specialized audiences, for example, computer classes, or various laboratories. In addition, audiences are different in terms of the number of seats and equipment.

Therefore, you need an associative entity type SA (Study Audiences), which describes the number of audiences in which each lesson can be conducted:

$$Z \leftarrow \pi_{ID\_s, ID\_g, ID\_t, ID\_p, ID\_a} ((S \bowtie SA) \bowtie A) \times P) \quad (5)$$

In addition, not all classes can be conducted on any training pair. Groups can have assigned days for self-preparation of students or on duty. Teachers may be busy on some days of the week or on some training pairs by doing other jobs not related to the classes conducting. Therefore, there is need for associative types of entities, for storing information about timetable positions (training pairs) that are acceptable for conducting classes — for teachers, academic groups and audiences. We denote them by PT, PG and PA, respectively. These relations have two attributes – the identifier of the training pair and the identifier of the object (teacher, group, audience). Finally, we obtain an expression for the relation Z:

$$Z \leftarrow \pi_{ID\_s, ID\_g, ID\_t, ID\_p, ID\_a} ((S \bowtie SA \bowtie A) \bowtie PA) \bowtie (PG \bowtie PT) \quad (6)$$

The relation Z is calculated by the formula (6) and, in fact, is a representation (view), and not a basic relation. But using the Z as a database table that is physically written to disk gives great advantages in calculating the schedule. These advantages exceed this method's drawback – data redundancy with storage in a large-scale table database. This disadvantage is not significant, since the Z is a working table that is filled with data only during program operation. When the schedule is finished, all data from the table is deleted.

The Z relation is used in the algorithm for step-by-step scheduling. At each step of the calculations, the Z relation tuples are ranked according to the increase in the freedom of setting the lesson in the schedule and the decrease in the density of classes in the teachers' schedule, which is determined with the help of the four coefficients K1, K2, K3, K4:

The relation Z is used in the algorithm of step-by-step scheduling. At each step the tuples of the relation Z are ranked according to the increase in the freedom of setting the lesson in the schedule and the decrease in the density of classes in the schedule of teachers, which is determined using the four coefficients K1, K2, K3, K4:

- K1 – the number of training pairs during the week, for which the lesson can be put on the schedule;
- K2 – the number of places in the two-dimensional space (the training pair, the audience) for which the lesson can be put on the schedule;
- K3 = Kpair – Kclass; K4 = Kpair / Kclass;
- Kclass – the number of lessons from the teacher, not yet scheduled;
- Kpair – the number of free teacher's pairs for which any of the teacher's remaining lessons can be put.

To determine the next lesson to be put on the schedule, a four-level sorting of the list of classes is carried out. The list is ranked first by K1, the second by K2, the third by K3, the fourth by K4. All ordering is performed by increasing the corresponding coefficients. The coefficients K1, K2, K3, K4 are easily computed by a select from the Z table using the grouped SELECT command.

The first lesson in the list is to be put on the schedule. When the lesson is putting on the schedule, the training pair is determined for it, on which it should be put to obtain the optimal schedule. The algorithm for adding a lesson to the schedule should include the fulfillment of several conditions, for example:

1) first of all, the lesson is put on the training pair that is the "window" in the schedule;

2) if there are no "windows" in the set of training pairs allowed for a given lesson, then the lesson is placed next to the lesson that is already in the schedule (on the previous or next pair);

3) if conditions 1 and 2 are not fulfilled for a set of training pairs that are valid for a given lesson, then the lesson is put on a day off from work for the minimum possible pair.

4) if conditions 1 – 3 are not fulfilled on the set of training pairs that are acceptable for a given lesson, the lesson is put on a schedule with the creation of a "window".

The information on the lessons already put on the schedule is stored in the relation TT (ID_s, ID_g, ID_t, ID_p, ID_a). For this table the SELECT command selects the occupied (busy) training pairs with grouping by the days of the week, and calculates the minimum and maximum occupied training pair for each day of the week, as well as the number of occupied pairs. "Window" in the schedule of the academic group is on the day when:

$$Pmax \geq Pmin + K, \quad (7)$$

where Pmax is the number of the last busy pair on some day of the week;

Pmin – the number of the first busy pair on this day of the week;

K – the number of occupied training pairs this day.

After adding the records of the lesson that has been put into the schedule to the table TT, all the tuples corresponding to this lesson are deleted from the table Z. When drawing up a schedule of classes, it must be fulfilled that one pair can not be put into several classes by an academic group, teacher or audience. Therefore, all tuples are deleted from the Z relation with the same learning pair ID, for which the teacher ID or group ID or audience ID is the same as the corresponding ID of the scheduled class.

The completed schedule database can be used in programs like "Student's Organizer", "Teacher's Organizer", "University Guide" and the like. These programs can be used as mobile applications.

Наукове видання

МАТЕРІАЛИ III МІЖНАРОДНОЇ НАУКОВО- ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ МОЛОДИХ ВЧЕНИХ

ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ГАЛУЗІ ПРИРОДНИЧИХ НАУК

6-8 червня 2018 року, Одеса, Україна

Відповідальний за випуск: Черепанова Катерина Валеріївна

Надруковано з готового оригінал-макета

Підписано до друку 15.05.2018 р. Формат 60х84/16
Папір офсетний. Ум. друк. арк. 11,16
Наклад 70 прим. Замовлення 497
Видавництво та друкарня «ТЕС»
(Свідоцтво ДК № 771) Одеса, Канатна 81/2
Тел.:(0482)42-90-98, (0482)42-89-72