DATA PRESENTATION AND TRANSFORMATION IN TRAIN SCHEDULE INFORMATION SYSTEMS

Eugene Kopytov[(a)], Vasilijs Demidovs[(b)], Natalia Petukhova[(c)]

[(a)] Transport and Telecommunication Institute, Riga, Latvia,
[(b) (c)] State Join-Stock Company "Latvian Railway", Riga, Latvia,

[(a)] kopitov@tsi.lv, [(b)] dem@ldz.lv, [(c)] natalia@ldz.lv

ABSTRACT
In the given paper the authors have considered two approaches of the multi-version data modeling in train schedule information systems, based on the different forms of data presentation: point and interval forms correspondingly. The method of the interval data transformation in the point form is proposed. It is realized using SQL tools. The received results are applied in the information system of the inland train traffic schedule on the Latvian Railway.

Keywords: train schedule, database, data presentation, point form, interval form, data transformation

1. INTRODUCTION

The principal issues of the train traffic schedule storage in railway information systems (IS) are connected with the existence of the multitude of its versions conditioned by the seasonal cycles of schedule changes, the days of week, by the systematic and unplanned repair operations, the transfers of the working and festive days, and other factors (Kopytov, Demidovs, and Petukhova 2008).

The efficiency of manipulating with the schedule greatly depends on the organization of temporal data in the database. In practices there exists a variety of the data presentation models determined by a set of different IS using the trains' schedule and characterized by its functionality, the requirements to the resources and the traditions of development.

Research of different models of presenting temporal data is considered in the paper (Jensen, Soo, and Snodgrass 1994). The peculiarities connected with the management of the events' periodicity in the schedule tasks and with the development of a correspondent class of temporal models have been studied in the works (Terenziani 2003, Luca 2005, Terenziani and Luca 2006). The existing models of presenting temporal data have been studied by the authors in more details with the account of the specific passenger trains' schedule (Kopytov, Demidovs, and Petukhova 2008; Kopytov, Petukhova and Demidovs 2010).

In the given paper the authors have considered two approaches of the multi-version data modeling in train schedule IS, based on the different forms of data presentation: point and interval forms correspondingly.

Point (daily) form is convenient for processing operative and analytical queries, but requires huge memory resources and great labour consumption for maintaining trains' traffic schedule in actual state. Though, the main problem of the above approach lies not in the volume of the stored data but in the complexity of providing the reliability of data and the immediacy of the schedule changes. Under the condition of frequent changes, these tasks become urgent.

Taking into account the fact that the schedule is usually made for a lengthy period of time (for example, for a season), it is more convenient to use the *interval form* of presenting data. This form allows substantial reduction of the stored data volume and provides flexibility of the schedule management. The interval form allows to use the data of the schedule's operative temporal database and to account all changes in the schedule as soon as they appear in the system. But along with this, the interval form is inferior to the point form in processing schedule queries and processing analytical queries in particular.

Thus, at different stages of processing the trains' schedule, there is reasonability of using different forms of presenting data, in one case the point form is preferable, in another case priority is given to the interval form. In the suggested research the authors use both models of the trains' schedule in one IS: the interval form – for storing data and the point form – for performing analytical calculations. For the transition of the data presented in the interval form to their point presentation, the authors have developed a special method of data transformation which is considered in the given article.

2. PECULIARITIES OF TRAIN SCHEDULE INFORMATION SYSTEM

The train schedule system considered by authors is the reference system, presenting the information on the trains schedule for any period of time on the web site. Its principal task is the storage of the information on train traffic as at the current moment, so in the future and in the past, as well as provision of the access to this information for the railway customers on the purpose of travel planning. Note, that this information is used by decision-makers on the purpose of carrying out the versatile analysis of the train traffic too.

The system of the train schedule possesses the following peculiarities, determining the rather significant level of the complexity of its implementation considered below.

- The active schedule of the trains traffic is arranged on the basis of core fundamental (seasonal) timetable and the number of other schedules, occurring as a result of such procedures as the trains removal, changes in the traffic schedule due to the accidents, pre-planned or unplanned repair operations, public events (the City Days, concerts, etc.), short-run including / excluding the stations in / from the route. Further these schedules are called as *the schedule versions*.

- In the course of time the various versions rotate. One and the same version can come into operation repeatedly: the fundamental timetable exchanges the altered one, and vice versa.

- One and the same train might have the different schedules on the different week days.

- On holidays and other special days (for example, holding the NATO summit in Riga in November 2006) all the trains run in accordance with the day off schedule. There is an exception for the certain trains when the special timetable is assigned for these days. But from practical side it is known that only 1% of all trains are assigned with the special schedule.

- The trains run on the days of week or month the schedule is assigned for (for example, on Fridays only, or twice a week on Tuesdays and on Thursdays, on the working days, on the week-ends, on the even days of the month, daily).

- There is a break in the local commuter traffic in Latvia at night. In this connection the following peculiar features can be mentioned: the passengers and the system percept the opening and the closing of the day in different ways: the system day spans from 00:00 till 23:59, but the passengers' day can span for more than 24 hours, and for convenience are determined to span from 04:00 of one day and till 03:59 of the following day. All the trains, departing the night of the following day, the passengers refer to the current day. There is a special term – the night trains. For the passengers' convenience the system of the schedule delivery should take into account this peculiarity of understanding the day notion.

- The system of scheduling the long-distance trains must take into consideration the time zones, changing in the course of the train movement along the route. There are also peculiarities connected with the schedule data management (schedule registration).

- First of all the system obtains the core fundamental schedule, then with the time running the operative changes are contributed to this schedule. Consequently, several versions of the schedule appear. Quite often the period of operation of such altered schedule is so prolonged, that it also becomes subjected to certain corrections. The operational time of these or those corrections can be contracted or prolonged.

- The data on the train schedule version comprise the concise information about the order, in accordance with which the new schedule is introduced, the date of this schedule version coming into operation, the date of going this schedule version out of operation, the type and the number of carriages in the train, the set of stops at the stations with information on the time of arrival or departure, the type of the transport at the stations (in case of certain repair works the train between the definite stations can be exchanged by the bus).

- The system of schedule is utilized by the customers by means of the Internet very intensively. As the statistics states, the number of addresses to the system reaches 50 times during one minute period. The search for the corresponding train takes place under different conditions and criteria consideration: the station of departure and arrival, the date and the time range of the departure, the possibility of travel in transit, and so on. As all the queries to the system occur in the on-line mode, it becomes very important to provide the velocity of the result delivery. This task is complicated by the great volume of the processed data. In the relation database one version of the timetable for one train takes about 1 KB. The minimum of data is taken into account, without considering the reference information (such as the stations names, trains types, etc.). The integrated volume of all data of the passenger trains commuters schedule of the Latvian Railway in interval form takes more than 100 MB (about 250 thousand records).

3. PERIODICITY IN THE TRAIN SCHEDULE

One of the most important properties of the schedule is its periodicity. It is the property that makes the basis of the effective schedule presentation in the databases. Periodicity means that the continuity of the object version's lifespan is broken cyclically and is valid only at the moments, which match the specified periodicity.

The periodicity of event is mathematically represents by the pair:

$$\langle [t_s, t_e], p \rangle ,$$

where $[t_s, t_e]$ is time interval; t_s and t_e are time of event beginning and time of event termination accordingly;

p is periodicity; the examples of p are the following: "on Tuesday and Wednesday", "on even days", "at weekends", "every first Monday of the month".

Recently, some commercial databases started to support user-defined periodicity in queries in order to provide "a human-friendly way of time handling" (see, e.g., TimeSeries in Oracle 8). On the other hand, only few relational data models support user-defined periodicity in the data, mostly using "mathematical" expressions to represent periodicity. In this area Paolo Terenziani and his followers' research is worth attention (Terenziani 2003, Terenziani and Luca 2006, Luca 2005). This group of researches offer a high-level "symbolic" language for representing user-defined periodicity, which is more human-oriented than mathematical ones. In the above-mentioned works researchers consider temporal relational model which supports user-defined "symbolic" periodicity (e.g., to express "on the second Monday of each month") in the validity time of tuples and cover time ranges (e.g., "from 01.01.2009 to 28.02.2009"). Temporal counterpart of the standard operators of the relational algebra is defined, new temporal operators and functions are introduced. Temporal algebra, which is consistent extension of the classical (atemporal) one, is offered.

Periodicity p can be represented in different ways, ranging from binary or numeric representation and ending with a high-level symbolic language, for example (Terenziani 2003). The first way is not visual and is limited to a specific set of recognizable types of periodicity. That is, when the periodicity is a very complex expression, then you may encounter with a limitation or ambiguity. For example, the periodicity of "the second Monday of every even month, as well as the first working day of every odd month except February" is a complex periodicity based on the basis of more simple types of periodicity connected by the certain rules. This approach allows to take into account such a complex periodicity. To do this, it computes the set of time moments of activity of each simple component of the periodicity and performs the subsequent operations with the computed item sets. Among these operations, there are all operations that can be performed with item sets: *merge*, *intersection*, etc.

However, you can look at the periodicity from another angle. It also represents a logical expression built on the basis of simpler types of periodicity, connected by the logical connectives in a certain sequence. Such a view gives quite different possibilities for computation. However, this approach has never been investigated so far.

4. EMPLOYING THE PERIODICITY PRINCIPLE IN THE RAILWAY SCHEDULE

As a matter of practice the train can have several active schedules simultaneously during the long period of time. Furthermore every train schedule is organized inclusive of days of running with the different *periodicity properties*, for example: on working days, on weekends, on the definite days of week, the first working day of week, the last day off of the week (notably it can be even Monday), on even or odd days etc.

As an example the multi-variant train schedule is presented in

Figure 1. This figure performs that the train has two basic schedules: one for the working days *wd*, and one for the weekends *rd*. Both timetables are valid on the time interval $[t_1, t_4]$, but on different week days. Then on Tuesdays and Thursdays (*Tue,Thu*) the amendment of this train schedule in the time period from $[t_1, t_4]$ is introduced for the systematic repair operations providing. Therefore in the time interval $[t_2, t_3]$ there are three timely schedule versions, activated on the days corresponding to the characteristics of *wd*, *rd* and (*Tue,Thu*), moreover, only one schedule version *wd* or *rd*, or (*Tue,Thu*) can be active in one day.

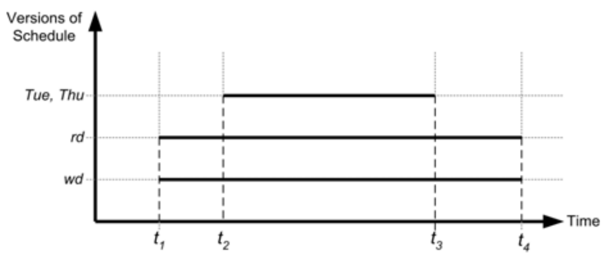

Figure 1: Multitude of the valid versions of the train schedule of different periodicity: *wd*, *rd* and (*Tue,Thu*)

This assertion is illustrated with help of the diagram in Fig.2. The axis "Day of week" shows the days of week, respectively 1 is Monday, 2 is Tuesday and so on. Time in this system is of the discrete nature, the level of the schedule granularity equals one day period. As the mentioned figure presents, one of three versions is the active train timetable in different days of week, and the other two versions are in the "shadow" zone at this moment. Moreover, the train schedule on Tuesdays and Thursdays (characteristic *Tue,Thu*) overlaps the working day schedule with the characteristic *wd*.

However besides the recurring versions of the train schedule there are also one-time changes in the timetable, connected with the special (altered) calendar. The following changes can be listed in this category: the festive days, the additionally appointed days-off, or the cancelled days-off. There is also such special occasion schedule alteration when the calendar working day exchanges the places with the calendar day-off. For example, in Latvia in 2007 the 30th of April, Monday (a working day) was transferred on the 14th of April, Saturday (a day-off). As a result on the 30th of April the trains ran according to the schedule of the weekends, and on the 14th of April the trains ran according to the working days timetable – instead of the 30th of April.

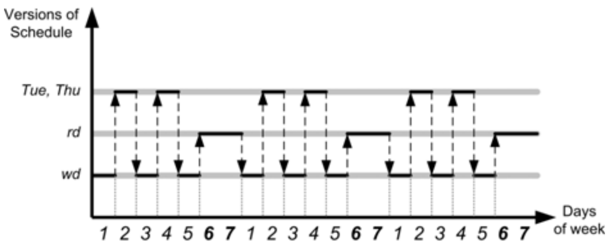

Figure 2: Periodicity in the train schedule

Such types of changes in the timetable or assignations provoke the anomalies (paradoxes). For example, the 19th of November, Monday, in 2007 was announced as an additional day-off, and all the trains moved this day according to the schedule of the weekends and days-off (like on the 18th of November), and this event arouse the malfunction of the periodicity, as it is shown in Fig.3. It is important to add that several trains with periodicity (Sundays) were cancelled on the 18th of November, because the trivial Sunday is not only the day-off, but also the last day of weekend, and the additional Sundays trains are considered for the passengers transportation from the traditional locations of the resting facilities on the days-off.

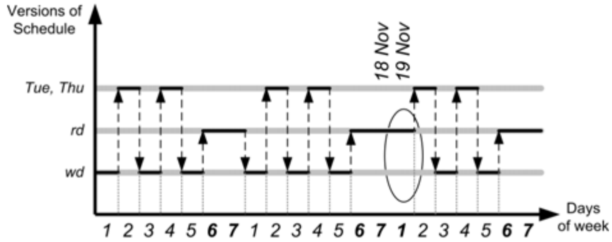

Figure 3: Periodicity malfunction in the train schedule

For correct functioning of the scheduling system, which is to take into account the above considered non-trivial rules, the well-defined formalization of the task assignment and the description of the scheduling temporal model are demanded. For this purpose the group of sets, characterizing the rules of arranging the trains running periodicity, the railway objects, and the entire system of scheduling itself, are introduced.

5. TRAIN SCHEDULE PRESENTATION MODELS

Two basic sets are introduced for the train schedule description:

$S = \{s_1, s_2, ..., s_k\}$ is the set of all stations of the railways, $|S| = k$;

$N = \{n_1, n_2, ..., n_m\}$ is the set of the trains, $|N| = m$.

The schedule for the train with the number $n \in N$ will be determined by the following vector:

$$v = \{n, \langle s_1^{(n)}, T_1 \rangle, \langle s_2^{(n)}, T_2 \rangle, ..., \langle s_\alpha^{(n)}, T_\alpha \rangle\}, \qquad (1)$$

where the pair $\langle s_j^{(n)}, T_j \rangle$ determines the *j*-th train stop, the station $s_j^{(n)} \in S$ exactly, and the train departure time T_j ; α is the number of the stops of the train running on the timetable *v*.

In railway information system the train schedule is stored for the given period of time – the period of observation. In connection with possible changes of the train traffic schedule there appears the task of adjusting the train schedule to the particular dates. Here we can use two kinds of models of presenting the schedule in the point and the interval forms considered below.

Point form. The data model of *the daily schedule presentation* presents data in the point form. This model was very popular in the first generation of train schedule IS. We can find examples of this approach in some up-to-date IS, for example in the International System of the Tickets E-sales and E-reservation "Express-3" (Russia).

The given model is based on making a calendar of each train traffic for every day. Then *i*-th schedule version $v_i^{(n)}$ for the train with number $n \in N$ will be determined by the tuple:

$$v_i^{(n)} = \{n, \langle s_1^{(n)}, T_1\rangle, \langle s_2^{(n)}, T_2\rangle, ..., \langle s_h^{(n)}, T_h\rangle, t_i\},\quad i = 1,2,...,m, \tag{2}$$

where the pair $\langle s_j^{(n)}, T_j\rangle$ determines the j-th stop of the train with number *n*;
h is the quantity of the stops of the train running on the timetable $v_i^{(n)}$; t_i is the day of operational period stored in database, when the train with number *n* is included in daily trains' schedule; *m* is the quantity of days on which the schedule of the train with number *n* is stored in database.

It should be noted that the number of the stored records, stating on what day, at what time, at what station every train stops, will be approximately equal to the product of multiplication of the number of trains by the average number stops and by the average number of days for which the schedule is stored. So, about two million records are needed to store the schedule of the trains of the inland traffic on the Latvian Railway in the relation database for a year time period, and for the greater railway companies the volume of the corresponding database will increase by times.

Interval form. The model used the interval form of data presentation significantly reduces the amount of stored data and does not limit the time period of data storage (Date, Darwen, and Lorentzos 2002). The usage of the interval form in train schedule IS, as well as its modifications, are considered in the paper (Kopytov, Demidovs, and Petukhova 2008).

For the specific schedule version identification for the train with number $n \in N$ the tuple $\langle n, C, t_f, t_s, t_e\rangle$ is employed. The property C presents the logical expression consisting of one or several elementary characteristics of the periodicity p_j connected by the logical operations signs $\vee, \wedge$ and $\neg$; t_s and t_e are the moments of time of the beginning and ending of the operational period of the considered schedule version, t_f is the time of the fixation of the schedule version in the database.

Then *i*-th schedule version $v_i^{(n)}$ for the train with number $n \in N$ will be

$$v_i^{(n)} = \{\langle n, C, t_f, t_s, t_e\rangle\ \langle s_1^{(n)}, T_1\rangle, \langle s_2^{(n)}, T_2\rangle, ..., \langle s_h^{(n)}, T_h\rangle\},\quad i = 1,2,...,q, \tag{3}$$

where *q* is the quantity of schedule versions of the train with number *n*, it is significant that *m* >> *q*.

For accounting a special calendar, we'll introduce the set $Z = \{z_1, z_2, ..., z_\mu\}$ of the transferred days of the train schedule, where z_i, $i = \overline{1, \mu}$ is the pair of days $\langle d_1^{\{i\}}, d_2^{\{i\}}\rangle$, determining, that the timetable assigned on the date $d_1^{\{i\}}$, is assigned on the date $d_2^{\{i\}}$ as well; $|Z| = \mu$.

Due to the complexity of managing the different periodicity schedule versions set, continuously appearing operative schedule changes, taking into consideration the non-trivial calendar, subjected to the changes, the development of the efficient methods of the active version determination is highly needed.

In the work (Kopytov, Petukhova and Demidovs 2010) the authors have investigated two methods of the temporal object active version determination: the logical rules (LR) method and the temporal elements (TE) method. An approbation of the offered methods for the actual object version determination has been carried out in the train schedule IS (on the example of Latvian Railways) and the efficiency of their application for solving different practical tasks has been determined. Obtained results testify a possibility for practical application of both LR and TE methods. At the same time there is an area of effective application for each of them.

6. INTERVAL DATA TRANSFORMATION INTO THE POINT FORM

The majority of the employed data analysis methods and correspondingly the majority of the analytical systems operate with the point forms of the data representation (for example, with the time sequences), but not with the interval forms. In the point form the fact is associated with the set of the time moments of the previously chosen granularity (for example, with the days, months, or years). The procedures over the described sets are very convenient for the fundamental computations or for the formal characterization. However, the mathematical apparatus of the relation databases management systems does not provide the ready-to-use facilities for the transformation of the interval form into the point one, and this fact required the elaboration of the appropriate procedures to be performed by the investigation under consideration.

Within the framework of the set task, the method of the interval data transformation in the point form was designed (Kopytov and Petukhova 2010). The scheme of data transformation is demonstrated in Fig. 4.

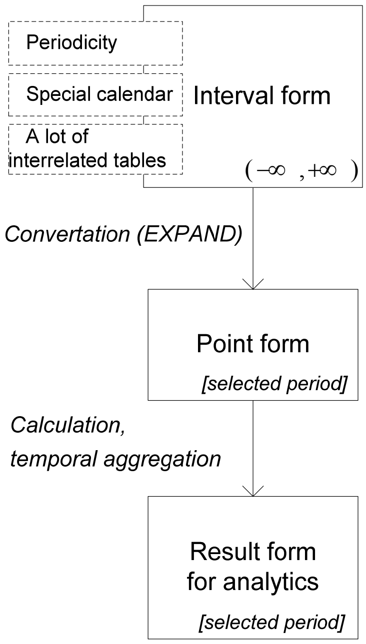

Figure 4: Scheme of the temporal data transformation

There is the explanation of the presented scheme. The classical interval form (*Interval form*) is initially inconvenient for the range of the certain calculations, but their complexity increases multifold when the following aspects exist: the periodicity, the special calendar, the necessity to use the several interconnected tables in the calculations[1].

The application of the procedure EXPAND to the interval form temporal data, suggested by the authors, solves the problem of the inconvenient data presentation for the computations, by the way of turning the interval facts into the time sequences, in which every value gives characteristics of this fact existence in the definite moment of time (*Point form*). Then the data undergo the further processing, often connected with the temporal aggregation procedures. This stage might require even more serious computations connected with the temporal concepts representation. The required data form (*Result form for analytics*) is the final result of the processing procedure.

The initial sets for every stage embrace the different time periods. The set of the interval form temporal data (the first block in Fig. 4) is capable of covering the unlimited time range (-∞,+∞). With the transfer to the point form (the second and the third blocks in Fig. 4) the time area of the temporal data is narrowed to the field of analysis (selected period), and this fact considerably contracts the data bulk in the computations and substantially saves the computing resources, and solves the problem of versions existence with the indefinite values of the functioning periods beginning and ending.

There is the observation of the suggested method.

We consider two basic relations $R1$ and $R2$:

$R1$ is the relation of the interval form, describing the set of the certain object versions and having the scheme $R1(A, t_s, t_e, C)$, where $A = \{a_1, a_2 \dots a_k\}$ is the set of the attributes, describing the object, k is the number of attributes. Every version has the functioning period from the moment t_s to the moment t_e and possesses the feature of periodicity *C*. Besides, the activity moments of this or that versions depend on the specific calendar *Z*, describing the one day properties transfer to the others (the additional days-off or additional working days, etc.);

$R2$ is the point form temporal relation with the scheme $R2(A,d)$, where d is the time moment when fact A is active.

It is required to execute $R1(A, t_s, t_e, C)$ transformation into $R2(A, t_n)$, and the data set of $R2$ has to be restricted by the fixed time period (specified analysis period) $T^{(q)} = [t_s^{(q)}, t_e^{(q)}]$, where $\forall d \in T^{(q)}$.

The transformation formula has the following view:

$$R2 = \sigma_{d \in T^{(q)}} R1 \times \tau(t_s, t_e, C)\,, \qquad (4)$$

where $\tau(t_s, t_e, C)$ is the function distributing the range $[t_s, t_e]$ on the d dates set, responding the requirement of the specified periodicity *C*:

$$\tau(t_s, t_e, C) = \{d \mid d \in [t_s, t_e], p(d, C) = true\}\,, \qquad (5)$$

where $p(d, C)$ is the periodicity predicate, which is true in the case if date d has the property *C*. The predicate $p(d, C)$ takes into consideration the specific calendar *Z* (for example, the transfer of the weekends schedule on first May's Tuesday in the connection with the national holidays in Latvia on 4th of May in year 2011);

$R1 \times \tau(t_s, t_e, C)$ is the construction, implementing the procedure of expansion; this procedure is the central operation of the entire transformation and provides the transformation from the interval form to the point one. The selection $\sigma_{d \in T^{(q)}}(\dots)$ restricts the result by the analysis area $T^{(q)}$. The capacity of the resulting set can reach $|R1| \cdot |\tau(T^{(q)})|$.

[1] The information systems objects in the databases, according to the relation theory are represented in the form of the interconnected tables. That is why receiving the demanded result, as a rule, requires processing procedure for several tables.

After the executing the expansion procedure the following reorganizations (*grouping*, *natural joining* and *projection*) become trivial. The employment the proposed function $\tau(t_s, t_e, C)$ permits to simplify the computations ultimately. For its implementation the SQL recursion mechanism is used. The created function TAU(dStart, dEnd, C) transforms the time interval [dStart, dEnd] into set of dates $D = \{d\}$, which corresponds to the periodicity and takes into account the special calendar.

The instruction of the function TAU creating has the following view:

```
CREATE FUNCTION TAU (
    dStart   DATE,
    dEnd     DATE,
    C        VARCHAR(30) )
  RETURNS TABLE  (D DATE )
  LANGUAGE SQL
  BEGIN ATOMIC
  RETURN
      --start of SQL recursion
     WITH D (d) AS
       (VALUES dStart
         UNION ALL
         SELECT d + 1 day as d
          FROM D
          WHERE d < dEnd
          )
      SELECT d  FROM D
      WHERE P(d,C)=1;
      --end of SQL recursion
   END;
```

Predicate function *P* implementing a predicate *p* determines the date InputDate compliance with the periodicity *C* taking into account the special calendar SPEC_DAYS:

```
CREATE FUNCTION P (
    InputDate DATE,
    C           VARCHAR(30) )
  RETURNS     SMALLINT
  LANGUAGE SQL
BEGIN ATOMIC
      DECLARE r    SMALLINT;
      DECLARE d1   DATE;
      DECLARE c1   VARCHAR(30);
   SET c1=c;
   IF c='wd' THEN
       SET c1='12345';
   IF c='rd' THEN
       SET c1='67';
   --spec. calendar
   SET d1=
     (SELECT InsteadOfDay
              FROM SpecDays s
              WHERE s.changeDay=InputDate);
   SET r=
        CASE
         WHEN LOCATE(LEFT(CHAR(
                DAYOFWEEK_ISO(
                 DATE(d1))), 1), c1) > 0
         THEN 1
         ELSE 0
        END;
RETURN r;
END;
```

7. EXAMPLES OF REALIZATION OF THE TRANSFORMATION METHOD

The first example illustrates the usages of the SQL-functions described in section 5, the second example presents realization of the method of transforming the interval data into the point ones.

Example 1. Let the fragment of data of a special calendar Z from the table SpecDays be given in the form of the following table:

ChangeDay	InsteadOfDay
22.04.2011	23.04.2011
25.04.2011	23.04.2011
04.05.2011	30.04.2011
23.06.2011	18.06.2011
24.06.2011	18.06.2011

where the pair 〈ChangeDay, InsteadOfDay〉 corresponds to the elements $\langle d_1^{\{i\}}, d_2^{\{i\}} \rangle$ of set Z .

For selecting the days-off (*rd*) in the interval [20.09.2011, 25.09.2011], the SQL-query with the use of the function TAU looks as follows:

```
SELECT d, DAYNAME(d) as weekday
 FROM TABLE(
     TAU(CAST('20.06.2011' AS DATE),
     CAST('28.06.2011' AS DATE),'rd'));
```

Having performed the given SQL-query, we'll get the following result:

D	Weekday
23.06.2011	Thursday
24.06.2011	Friday
25.06.2011	Saturday
26.06.2011	Sunday

As we see from the resulting table, the days 23.06.2011 (Thursday) and 24.06.2011 (Friday) are also related to the days-off. These dates have been described in a special calendar as the days-off.

SQL-query for selecting the week days (*wd*) in the interval [20.09.2011, 25.09.2011] looks as follows:

```
SELECT d, DAYNAME(d) as Weekday
FROM TABLE(
      TAU(CAST('20.06.2011' AS DATE),
      CAST('28.06.2011' AS DATE),'wd'));
```

The results of this query are presented in the following table:

D	Weekday
20.06.2011	Monday
21.06.2011	Tuesday
22.06.2011	Wednesday
27.06.2011	Monday
28.06.2011	Tuesday

Note that the re-determined days 23.06.2011 (Thursday) and 24.06.2011 (Friday) are not present in the resulting table.

Example 2. There is examined the task of procurement of the time series, describing the changes of the trains number on the railway line Riga – Daugavpils (in both directions) for May, 2009.

The segment of the database comprising the data, necessary for calculations, is presented in Fig 5. The temporal table Schedule contains a lot of the trains schedule versions, each of which has its own period of operation from the moment SchStart to the moment SchEnd and possesses the periodicity characteristics Periodicity; this fact is reflected in the corresponding temporal attributes of the valid time Valid Time. The moments when this or that version is active depend on the peculiar calendar Special calendar, describing the transition of the properties from one days to the other ones (the additional days off or the working days, and so on) and affecting all the schedule versions. The detailed description of the schedule versions (time of the train stops at the stations) is introduced in the table Schedule Stops. The tables Trains and Lines are not the temporal ones and they contain the information on the trains, as well as their belonging to the railway lines.

The structure of the time series under examination is assumed in the form $(Date, Line, TrainQty)$, its temporal area in the form [01.05.2009; 31.05.2009] accomplishing $T^{(q)}$, the period of the schedule activity [$schStart$, $schEnd$] accomplishing $T^{(S)}$; the periodicity attribute Periodicity accomplishing C.

Then the formula of transformation obtains the following form:

$$R(Date, Line, TrainQty) = \\ = \pi_{d, Line, COUNT(*)}(Lines \bowtie Trains \bowtie \\ \bowtie \sigma_{d \in T^{(q)}}(Schedule \times \tau(T^{(S)}, C))),$$

where construction $\pi_{d, Line, COUNT(*)}(Lines \bowtie Trains \bowtie ...)$ solves the standard task of grouping and the natural joining the connected relation; construction $Schedule \times \tau(T^{(S)}, C)$ executes the expansion operation.

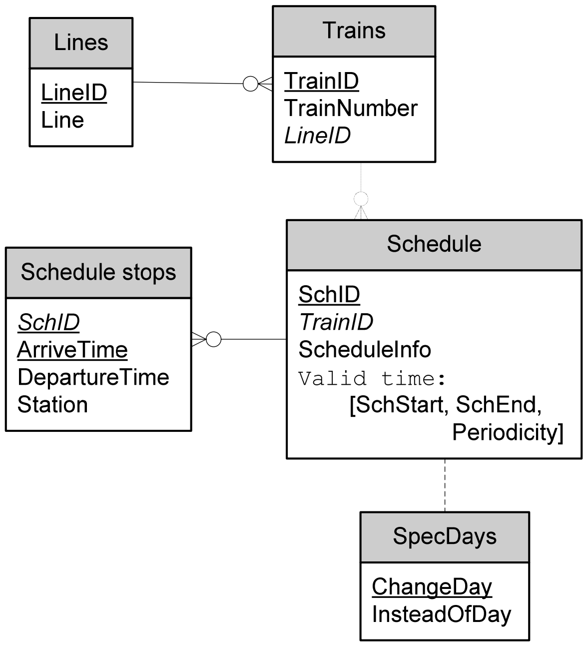

Figure 5: Segment of the database of the passenger trains schedule

As a result of its accomplishing, 80 original tuples of the interval form of the relation *Schedule* (trains schedule on the line Riga – Daugavpils) are transformed into 2170 tuples of the point form. Without taking into account other transformation this construction has the following view in the SQL language:

```
SELECT Schedule.*, d
FROM Schedule,
     TAU(schStart, schEnd, periodicity);
```

The completed SQL-statement for constructing the time series $R(Date, Line, TrainQty)$ has the following view:

```
SELECT d as date, line,
       count(train) as TrainQty
 FROM Schedule,
       TAU(schStart, schEnd, periodicity),
       Lines, Trains
 WHERE Schedule.trainID = Train.trainID
       AND Trains.LinesID = Lines.LineID
       AND d between
       '01.05.2009' AND '31.05.2009'
 GROUP BY line, d;
```

CONCLUSIONS

The approach to presenting the data of the trains' schedule considered in the given paper suggests simultaneous usage of two principally different forms of storing data: point form and interval form. It allows using the advantages of both forms. Thus, the point form provides convenient processing of the schedule data in performing analytical calculations; the interval form substantially reduces the volume of the stored data and the labour consumption on the schedule maintenance in the actual state.

The two forms of presenting data and the need of their simultaneous usage have required development of a mechanism of transforming data from one form to another; for this purpose, the paper suggests development of a method of transforming the schedule data from the interval form to the point one.

The received results are applied in the information system of the inland train traffic schedule on the Latvian Railway. However, they can be implemented in the information systems of other railway companies. Moreover in the obtained results have the universal nature and can be applied in temporal databases regardless their destination.

The paper considers realization of the predicate function P, which allows employment of the simplest logical formula for the periodicity C containing the elementary attributes connected by the operators "OR". At the further stage of research, realization of the predicate function P will be elaborated in such a way that it would be possible to describe more complicated formulae of the periodicity C which can occur in practice. For this, it is supposed to develop a corresponding specialized compiler.

REFERENCES

Date C. J., Darwen H. and Lorentzos N. A., 2003. *Temporal Data and the Relational Model.* Morgan Kaufmann Publisher.

Jensen C. S., Soo M. D. and Snodgrass R. T., 1994. Unifying Temporal Data Models via a Conceptual Model. *Information Systems*, 1, 513-547.

Kopytov E., Demidovs V. and Petoukhova N., 2008. Application of temporal elements in the railway schedule systems. *Transport and Telecommunication*, 9 (2), Riga, 14-23.

Kopytov E. and Petukhova N., 2010. Application of Temporal Databases in Modeling of Railway Processes. *Proceedings of the International Symposium on Stochastic Models in Reliability Engineering, Life Science and Operations Management (SMRLO'10),* pp. 586-594. February 8-11, SCE – Shamoon College of Engeneering (Beer Sheva, Israel).

Kopytov E., Petukhova N., Demidovs V., 2010. Method for Railway Schedule Support in Temporal Databases. *Proceedings of the 9-th Join Conference on Knowledge-Based Software Engineering (JCKBSE'10).* August 25-27, Technologija, pp. 178-191 (Kaunas, Lithuania).

Terenziani P., 2003. Symbolic User-Defined Periodicity in Temporal Relational Databases. *IEEE Transactions on Knowledge and Data Engineering*. February, Volume 15, Issue 2, pp. 489-509.

Terenziani P. and Luca A., 2006. Temporal reasoning about composite and/or periodic events. *Journal of Experimental & Theoretical Artificial Intelligence (Taylor & Francis)*. March, Volume 18, No 1, pp. 87-115.

Luca A., 2005. *Representing and reasoning with classes and instances of possibly periodic events. Theory, algorithms and applications.* Dr. techn. thesis, 118 p.

AUTHORS BIOGRAPHY

EUGENE A. KOPYTOV was born in Lignica, Poland and went to the Riga Civil Aviation Engineering Institute, where he studied Computer Maintenance and obtained his engineer diploma in 1971. Candidate of Technical science degree (1984), Kiev Civil Aviation Engineering Institute. Dr.sc.ing. (1992) and Dr.habil.sc.ing. (1997), Riga Aviation University. Professor (1998). Present position: Chancellor of Transport and Telecommunication Institute, professor of Computer Science Department. Member of International Telecommunication Academy. Fields of research: statistical recognition and classification, modelling and simulation, modern database technologies. Publication: 260 scientific papers and teaching books, 1 certificate of inventions.

His e-mail address is: kopitov@tsi.lv

VASILIJS V. DEMIDOVS was born in Murmansk, Russia and went to the Riga Polytechnic Institute (Riga Technical University), where he obtained his engineer diploma in 1987. He

studied in Transport and Telecommunication Institute under the Doctor's program from 2001 to 2004 and obtained the scientific degree of Doctor of Science in Engineering (Dr.sc.ing.) in 2006 by scientific research direction "Transport and Logistics". He has got a docent since 2007. Present position: a Head of Database Management System Unit of Information Technology Centre of State Joint-Stock Company "Latvian Railway"; docent of Computer Science Department of Transport and Telecommunication Institute. Fields of research: modern database technologies, Business Intelligence, Data warehouse technologies. Publication: 31 scientific works in Belgium, Estonia, Italy, Latvia, Lithuania, Russia, USA, including 18 scientific papers.

His e-mail address is: dem@ldz.lv

NATALIA Y. PETUKHOVA was born in Riga, Latvia. She studied Computer Sciences in Transport and Telecommunication Institute and obtained her MS in 2001, obtained the scientific degree of Doctor of Science in Engineering in 2011 by scientific research direction "Transport and Logistics". Present position: leader specialist and DB administrator of Database Management Systems department of Computer Centre of Latvian Railway; docent in Transport and Telecommunication Institute. Publications: 18 scientific papers. Scientific activities: databases, data warehouses, business intelligences.

Her email address is: natalia@ldz.lv