journal homepage: www.sabauni.net/ojs

Saba Journal of Information Technology and Networking (SJITN)

S.J.I.T.N Saba Journal of Information Technology and Networking

Article

The formalization information model of scheduling university using language Tutorial D

Dr. Amr Abu Hania

Platov South Russian State Polytechnic University

Article info

Article history:

Accepted Nov, 2016

Keywords:

Relational language ,
Object-Oriented
Database,
Process ,
Timeslot and schedule.

Abstract

The University Class Scheduling Problem is concerned with assigning a number of courses to classrooms, taking into consideration different constraints like classroom capacities and university regulations. It also attempts to optimize the performance criteria and distribute the courses fairly to classrooms depending on the ratio of classroom capacities to course enrollments. The problem is a classical scheduling problem, and is considered to be NP-complete. Several formulations of information models have been proposed to solve scheduling problems, most of which are based on local search techniques. In this paper, we propose a complete approach using two relational database languages that are associated with this model and that are the Structured Query Language (SQL) and Tutorial D. This research describes the composition of information tables, as well as their structure and relationships. The relational model of data has a strong influence on the database field and cannot be discarded for any future direction regarding the development of databases.

* Corresponding author: Dr. Amr Abu Hania
E-mail: am27er@gmail.com

1. Introduction

Since the introduction of the relational model of data [7], relational database management systems came into existence and have proven to be dominant in the database field. These are software products that can be used in creating and managing-relational databases. These software products are all based on relational database languages that give clients the ability to interact with data that is stored in relations. There are many relational database languages in the market but the most popular and standardized one is the Structured Query Language (SQL).

The purpose of this paper is the description of the structure of the model scheduling problems in the language Tutorial D. Database programming language called Tutorial D was defined to replace SQL. Tutorial D is to form a solid foundation in which the Third Manifesto can be built upon. This foundation must be originating from the roots of the relational model of data that was proposed by [7] without any existence of SQL. This is due to the many deficiencies that provide a detail explanation of those SQL deficiencies.

2. Tutorial D

Tutorial D is a relational language with the purpose of forming a solid foundation for future database management systems. It all started when Date and Darwen observed some of the trends which attempted to integrate object and relational technologies but in an ill-defined manner. The first of these attempts was the Object-Oriented Database System Manifesto which was proposed by [8]. It considered object-oriented database systems to be theoretically and experimentally useful, and thus, needed a common data model to formalize those systems.

According to Atkinson et al. (1989), the reason behind this is that object-oriented database systems lacked a common data model before the introduction of the Object-Oriented Database System Manifesto. However, Date and Darwen (2000) criticized this approach to be ignoring the relational model of data,and thus, not suitable for a future direction of database management systems to be moved along. This is due to the importance of the relational model of data, as described earlier. The Second attempt came along after the Object-Oriented Database System Manifesto and was called the Third-Generation Database System Manifesto. Date and Darwen[4,5] proposed a third attempt of directing the next generation of database management systems. A manifesto was formed and was called The Third Manifesto, with the purpose of forming a theoretically correct mechanism of building object/relational database management systems which evolve from the original relational model of data but not replace it, because of its importance and relevance to the database field [4].Subsequently, a new database programming language called Tutorial D was defined to replace SQL[6].

The purpose of Tutorial D is to form a solid foundation in which the Third Manifesto can be built upon. This foundation must be originating from the roots of the relational model of data that was proposed by [7] without any existence of SQL[3,4,5].

2.1 Research Problem

In the base of the mathematical model of the subject area “Schedule”, the concept of “process” is considered the most significant (a set of interrelated resources and activities that are linked to a specific time interval). Let’s perform the formalization of the concepts “resource”, “time interval”, “activity” and “schedule” within the subject area.

The resources in this case are courses, lecturers, classrooms, and time slots. For example, one obvious constraint is that a lecturer cannot teach more than one course at the same time. In the base of mathematical model of the subject area”Schedule” the concept of “process” is considered the most significant. It is a set of inter-related sources and activities that are linked to a specific time interval. Let’s perform the formalization of the concepts”resource”, “time interval”, “activity” and “schedule”within the subject area [1,2].

There are many deficiencies and flaws surrounding the use of SQL as a relational database language. In an attempt to avoid using SQL, Tutorial D was introduced as anew relational database language that addresses SQL deficiencies and introduces new feature which realizes the full potential of the relational model of data[4]. However, due to the popularity of SQL, adopting Tutorial D can be difficult. Hence, the need to find a suitable mechanism to address these issues is crucial. One suggested solution is to form database in Tutorial D in order to allow legacy SQL clients to inter operate with Tutorial D databases[3,5,6]. As a resource we consider the university classroom assets where it is a multi-index.

2.1.1 The Formalization of the Essence of "Resource"

Let us describe using the basic structures of the language relational relations associated with the formalization of the concept of "resource" [1,5]: (buildings for classrooms - BCR), (types of classrooms for use - TCRU), (types of classrooms on capacity - TCRC) and (classrooms) - CR:

```
VAR BCR BASE RELATION
{NB# INTEGER,
 NAME CHAR}
 PRIMARY KEY {NB#};
VAR TCRU BASE RELATION
{TCRU# INTEGER,
 NAME CHAR}
 PRIMARY KEY {TCRU#};
VAR TCRC BASE RELATION
{TCRC# INTEGER,
 MAX_CAP INTEGER,
 NAME CHAR}
PRIMARY KEY {TCRC#};
VAR CR BASE RELATION
 {NCR# INTEGER,
 NB# INTEGER,
 TCRU# INTEGER,
 TCRC# INTEGER,
NAME CHAR,
CAPACITY INTEGER}
PRIMARY KEY {NCR#},
FOREIGN KEY {NB#}
REFERENCES BCR,
FOREIGN KEY {TCRU#}
REFERENCES TCRU,
FOREIGN KEY {TCRC#}
REFERENCES TCRC;
```

Attributes of these relations have the following meanings: NB #, TCRU #, TCRC #, NCR # - unique keys corresponding tables; NAME - the name of ((buildings for classrooms , types of classrooms for use - TCRU, types of classrooms on capacity and classrooms); MAX_CAP - capacity audience of this kind; CAPACITY - capacity of a specific audience.

2.1.2 The Formalization Entity "Time Interval"

When formalizing the essence timeslot (time interval) we note, first of all, that timetable is compiled for a specified period of training. Let us now describe the process of formalizing the notion of a time interval in the language of Tutorial D[3,4,5]. Basic relational relationship: (training weeks - TW), (days of the week - DW), (timeslots in the day - TD), (timeslots in the period - TP):

```
VAR TW BASE RELATION {
NW# INTEGER,
NAME CHAR}
 PRIMARY KEY {NW#};
VAR DW BASE RELATION
{ND# INTEGER,
 NAME CHAR}
 PRIMARY KEY {ND#};
VAR TD BASE RELATION
{NTD# INTEGER,
TIME INTEGER}
PRIMARY KEY {NTD#};
VAR TP BASE RELATION
{NT# INTEGER,
NW# INTEGER,
ND# INTEGER,
NTD# INTEGER}
PRIMARY KEY {NT#},
FOREIGN KEY {NW#}
REFERENCES TW,
FOREIGN KEY {ND#}
REFERENCES DW,
 FOREIGN KEY {NTD#}
REFERENCES TD;
```

2.1.3 The Formalization Entity of "Activity"

Begin the process of formalizing the notion of activity in the language Tutorial D to determine the following relational relations: (course title - CT), (types of lessons - TL), (courses - CRS), (departments of university - DU), (positions - PST), (teachers - TCH):

```
VAR CT BASE RELATION
{NCT# INTEGER,
 NAME CHAR}
PRIMARY KEY {NCT#};
VAR TL BASE RELATION
{NTL# INTEGER,
 NAME CHAR}
 PRIMARY KEY {NTL#};
VAR CRS BASE RELATION
{NCRS# INTEGER,
 NCT# INTEGER,
 NTL# INTEGER}
PRIMARY KEY {NCRS#},
FOREIGN KEY {NCT#}
REFERENCES CT,
FOREIGN KEY {NTL#}
REFERENCES TL;
VAR DU BASE RELATION
{NDU# INTEGER,
 NDU_P# INTEGER,
 NAME CHAR}
PRIMARY KEY {NDU#},
FOREIGN KEY {NDU_P #}
REFERENCES DU;
VAR PST BASE RELATION
{NPST# INTEGER,
 NAME CHAR}
 PRIMARY KEY {NPST#};
VAR TCH BASE RELATION
{NTCH# INTEGER,
 NPST# INTEGER,
 NDU# INTEGER,
 L_NAME CHAR,
 F_NAME CHAR,
 S_NAME CHAR,
 SHORT CHAR}
PRIMARY KEY {NTCH#},
 FOREIGN KEY {NPST#}
REFERENCES PST,
FOREIGN KEY {NDU#}
REFERENCES DU;
```

2.1.4 The Formalization of Entity "Learning Groups"

Note that between the entities "learn" and "group", there is the relation "many to many".Relational attitude (study groups) can be described as follows[1,2,3]:

```
VAR GROUPS BASE RELATION
{GROUP# INTEGER,
 NAME CHAR,
NDU# INTEGER,
NUMBER# INTEGER}
PRIMARY KEY {GROUP#}
FOREIGN KEY {NDU#}
REFERENCES DU;
```

In this respect, it is assumed that each study group GROUP# belongs to only one department of the university NDU # (department or faculty), although in practice there are cases of "mixed" flows, including academic groups from several faculties and so the attitude GROUPS still require clarification. In addition, each group has a NAME, for example "FITU-1-5, 5B, FMF 1-2, 3" and for each group the number of students in it NUMBER #.

2.1.5 The Formalization of Entity "Learning Unit"

To complete the formalization of the concept of operations, we make a record in the language Tutorial D (teaching unit - U)

```
VAR U BASE RELATION
 {NU# INTEGER,
 NG# INTEGER,
NCRS# INTEGER,
 NTCH# INTEGER}
PRIMARY KEY {NU#}
FOREIGN KEY {NG#}
REFERENCES GROUPS,
FOREIGN KEY {NCRS#}
REFERENCES CRS,
FOREIGN KEY {NTCH#}
REFERENCES TCH;
```

2.1.6 The Formalization of Entity "Schedule"

The above mentioned formalization of such concepts as a resource, is a temporal interval and an activity that allows examining a time-table (schedule) as some unambiguous reflection of from setsU . In a set (work is Cartesian)

$T \times A$:

$$s: U \rightarrow T \times A \in S$$

When choosing a format for the schedule in the computer's memory, it is convenient to introduce the set C={c1,c2,…,cm}={(t1,a1),…,(t1,anA),…,(tnT,a1),…,(tnT,anA)}, which we call the " Schedule Planning Grid "[1,2,5]. This - Table- empty graph, each cell corresponds to a pair "timeslot-audience". Scheduling in these cells is placed ("are assigned") in training units.

	(t_1,a_1)	(t_1,a_2)	…	(t_1,a_{nA})	(t_2,a_1)	…	(t_{nT},a_{nA})
u_1	0	1	…	0	0	…	0
u_2	0	0	…	0	1	…	0
…	…	…	…	…	…	…	…
u_{nU}	0	0	…	1	0	…	0

We shall assume that each grid cell has a unique number (an integer in the range 1 ... m),by which it can uniquely identify and get all necessary supporting information: Code classroom and code timeslot, where you can find the week number, day of the week and the number of training "pair".Example illustrates the process of learning the accommodation unit in the schedule planning grid as shown in Figure 1.

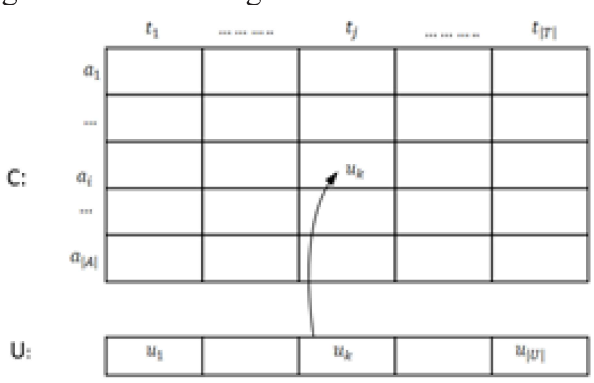

Fig.1. Placing the training unit in the Schedule Planning Grid

Schedule Planning Grid – SPG, can be represented by the following relational relationship:

VAR SPG BASE RELATION

{NCELL# INTEGER,
NCR# INTEGER,
NT# INTEGER}
PRIMARY KEY {NCELL#}
FOREIGN KEY {NCR#}
REFERENCES CR,
FOREIGN KEY {NT#}
REFERENCES TP;

NCELL# - unique key record (code grid cell), NCR# - code audiences, NT# - at timeSlots
At the end of the process of formalization of the basic objects (entities) of the problem, we present a referential attitude to the schedule:

VAR S BASE RELATION

{NS# INTEGER,
NU# INTEGER,
NCELL# INTEGER}
PRIMARY KEY {NS#}
FOREIGN KEY {NU#}
REFERENCES U,
FOREIGN KEY {NCELL#}
REFERENCES SPG;

There NS# - code element schedules, NU# - code teaching unit, NCELL# - code Schedule Planning Grid[1,2].

3.Conclusion and Future Work

In this research a new methodology for solving university course scheduling is proposed and implemented using real data set from the South Russia State Technical University. To Conclude this research, we can say that it was found that Tutorial D addresses the SQL deficiencies. This is achieved through providing true relational features that include rejecting duplicate tuples, NULLs and attribute ordering. Hence, utilizing Tutorial D can be useful in order to conform to the relational model of data and avoid SQL deficiencies. The advantages of this approach are:
-Synergistic effect of the combination of severity of each of notations.
- Problem solving followed by computer implementation models.
-Tasks, with the use of database technology, oriented to the client-server architecture.
-Using data structures in memory, which is typical for desktop applications or "thick" clients.

References

[1] Ivanchenko A.N., Abu hania A.(2013),"information model scheduling problems for the university" News . institutions of higher learning. North -Caucasian region.Tech. Sciences -.№ 3 (172).C. 14-16.

[2] Ivanchenko A.N., Abu hania A. (2012) "On th eformulation scheduling problems for the university" News . institutions of higher learning. North -Caucasian region.Tech. Sciences -.№ 6. - C.9-13.

[3] Date, C. J. (2013). Relational Theory for Computer Professionals. " O'Reilly Media, Inc.".

[4] Date, C. J., & Darwen, H. (2007). Databases, types and the relational model: the third manifesto. Addison-Wesley Longman.

[5] Date, C. J., & Darwen, H. (2010). Database Explorations: Essays on the Third Manifesto and Related Topics. Trafford publishing.

[6] Almulla, A. Y. (2007). Translating between SQL and Tutorial D.

[7] Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387.

[8] Atkinson, M. P., Bancilhon, F., DeWitt, D. J., Dittrich, K. R., Maier, D., & Zdonik, S. B. (1989, December). The Object-Oriented Database System Manifesto. In DOOD (Vol. 89, pp. 40-57).